IPHV8I1020X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

EFFICENT KEYWORD SEARCH IN RELATIONAL DATA USING INDEX STRUCTURES

[#1]KOTTE THRINATH REDDY, M.Tech Student,
[#2]RAMESH PONNALA, Assistant Professor,
Dept of CSE,
SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES, KARIMNAGAR, T.S., INDIA.

Abstract: A type of search that looks for matching documents that contain one or more words specified by the user is called keyword search. Find the info we need. It is for searching linked data sources on the web. There is a method for computing top-k routing plans based on their potentials to contain results for a given keyword query. Keywords and the data elements mentioning them are related by using a keyword-element relationship. A multilevel scoring mechanism is 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. Hence, we use Keyword Query searching for linked data. In this paper, we propose different approaches for keyword query routing through which the efficiency of keyword search can be improved greatly. By routing the keywords to the relevant data sources the processing cost of keyword search queries can be greatly reduced. In this paper, we contrast and compare four models – Keyword level, Element level, Set level and query expansion using semantic and linguistic analysis. These models are used for keyword query routing in keyword search.

Key words: Keyword searching, Uncertain graph, algorithm, Keyword routing, graph data, Keyword query.

I.INTRODUCITON

Keyword search has been deduced to retrieve useful data from database, graph data. Keyword search has major advantage i.e. it is easy to operate. Users do not have to understand the query language and the database schema, and can gain the knowledge very quickly how to use information retrieval. Now a days, the study of keyword search technology based on Graph data has become a hot spot, and it is generally applied to the field of information retrieval. In the field of traditional graph database, the research on keyword search has already gained some achievement, but in the field of uncertain graph data, the study on keyword search has barely started. Especially recently, quite a lot efforts have been put for keyword search over graphs, However, all graphs in the database are assumed to be certain or precise, and this assumption is often not valid in real-life applications. As RDF data and XML data can be highly unreliable due to errors in the web data or data expiration.

In the application of the data integration, it is needed to incorporate such RDF data from various data sources into an incorporated database. Uncertainties or inconsistencies often exist in this case. Like In social networks, each link between any two persons is often associated with a probability that represents the uncertainty of the link or the strength of influence a person has over another person in viral marketing. XML data having graph or tree form, uncertainties are integrated in XML documents known as probabilistic XML document (p-document).Keyword searching in RDF data, social networks and XML data have many important applications. For data with relational and XML schema, specific query languages, such as SQL and XQuery, have been developed for information retrieval. In order to query such data, the user must master a complex query language and understand the underlying data schema. In relational databases, information about an object is often scattered in multiple tables due to normalization considerations, and in XML datasets, the schema are often complicated and embedded XML structures often create a lot of difficulty to express queries that are forced to traverse tree structures. Furthermore, many applications work on graph-structured data with no obvious, well-structured schema, so the option of information retrieval based on query languages is not applicable. Both relational databases and XML databases can be viewed as graphs. Specifically, XML datasets can be regarded as graphs when IDREF/ID links are taken into consideration, and a relational database can be regarded as a data graph that has tuples and keywords as nodes.

In the application of the data integration, it is needed to incorporate such RDF data from various data sources into an integrated database. In this case, uncertainties/inconsistencies

often exist. Like In social networks, each link between any two persons is often associated with a probability that represents the uncertainty of the link or the strength of influence a person has over another person in viral marketing. In XML data (a tree or graph structure), uncertainties are incorporated in XML documents known as probabilistic XML document (p-document). Keyword searching in RDF data, social networks and XML data has many important applications.

Therefore, it is necessary to relax the strict assumption of Deterministic or well certain graphs and study keyword search over uncertain graphs. Keyword Query Analysis and Mining sub-graph pattern is the ultimate goal of research on uncertain graph data management to retrieve the useful data from uncertain graph data.

II.RELATED WORK

Keyword Query Search can be divided into two directions of work. They are:

1) Keyword search approaches compute the most relevant structured results.

2) Solutions for source selection compute the most relevant sources.

In the keyword searching, we mainly follow two approaches. They are schema-based approaches and schema-agnostic approaches.

Schema-based approaches are implemented on top of off-the-shelf databases. A keyword is processed by mapping keywords to the elements of the databases, called keyword elements. Then, using the schema, valid join sequences are derived and are employed to join the computed keyword elements to form the candidate networks that represent the possible results to the keyword query.

Schema-agnostic approaches operate directly on the data. By exploring the underlying graphs the structured results are computed in these approaches. Keywords and elements which are connected are represented using Steiner trees/graphs. The goal of this approach is to find structures in the Steiner trees. Various kinds of algorithms have been proposed for the efficient exploration of keyword search results over data graphs, which might be very large. Examples are bidirectional search [4] and dynamic programming [5]. Recently, a system called Kite extends schema based techniques to find candidate networks in the multi source setting [6]. It employs schema matching techniques to discover links between sources and uses structure discovery techniques to find foreign key joins across sources. Also based on pre computed links, Hermes [7] translates keywords to structured queries. In order to get the efficient results for keyword search, the selection of the relevant data sources plays a major role. The main idea is based on modeling databases using keyword relationships. A keyword relationship is a pair of keywords that can be connected via a sequence of join operations. A database is considered relevant if its keyword relationship model covers all pairs of query keywords.

M-KS [3] considers only binary relationships between keywords. It incurs a large number of false positives for queries with more than two keywords. This is the case when all query keywords are pair wise related but there is no combined join sequence which connects all of them. GKS [8] addresses this problem by considering more complex relationships between keywords using a Keyword Relationship Graph (KRG). Each node in the graph corresponds to a keyword. Each edge between two nodes corresponding to the keywords. For routing the keywords to the relevant data sources and searching the given keyword query, we propose four different approaches.

They are:

1) Keyword level model

2) Element level model,

3) Set level model, and

4) Query expansion using linguistic and semantic features.

We compute the keyword query result and keyword routing plan [1] which is the two important factors of keyword routing. In keyword level, we mainly consider the relationship between the keywords in the keyword query. This relationship can be represented using Keyword Relationship Graph (KRG) [8]. It captures relationships at the keyword level. As opposed to keyword search solutions, relationships captured by a KRG are not direct edges between tuples but stand for paths between keywords.

For database selection, KRG relationships are retrieved for all pairs of query keywords to construct a sub graph.

Based on these keyword relationships alone, it is not possible to guarantee that such a sub graph is also a Steiner graph (i.e., to guarantee that the database is relevant). To address this, sub graphs are validated by finding those that contain Steiner graphs. This is a filtering step, which makes use of information in the KRG as well as additional information about which keywords are contained in which tuples in the database. It is similar to the exploration of Steiner graph in keyword search, where the goal is to ensure that not only keywords but also tuples mentioning them are connected. However, since KRG focuses on database selection, it only needs to know whether two keywords are connected by some

IPHV8I1020X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

join sequences or not. This information is stored as relationships in the KRG and can be retrieved directly. For keyword search, paths between data elements have to be retrieved and explored. Retrieving and exploring paths that might be composed of several edges are clearly more expensive than retrieving relationships between keywords.

Keyword search over relational databases finds the answers of tuples in the databases which are connected through primary/foreign keys and contain query keywords. As there are usually large numbers of tuples in the databases, these methods are rather expensive to find answers by on-the-fly enumerating the connections. To address this problem, proposed tuple units [9] to efficiently answer keyword queries. A tuple unit is a set of highly relevant tuples which contain query keywords.

III. PROPOSED SYSTEM

To route keywords only to relevant sources to reduce the high cost of processing keyword search queries over all sources. A novel method was used for computing top-k routing plans based on their potentials to contain results for a given keyword query. It employs a keyword element relationship summary that compactly represents relationships between keywords and the data elements mentioning them. A multilevel scoring mechanism was proposed for computing the relevance of routing plans based on scores at the level of keywords, data elements, element sets, and sub graphs that connect these elements. Also to investigate the problem of keyword query routing for keyword search over a large number of structured and Linked Data sources. This system was having more advantages: 1) Routing keywords only to relevant sources can reduce the high cost of searching for structured results that span multiple sources. 2) The routing plans, produced can be used to compute results from multiple sources.

In order to get the efficient results for keyword search, the selection of the relevant data sources plays a major role. The main idea is based on modeling databases using keyword relationships. A keyword relationship is a pair of keywords that can be connected via a sequence of join operations. For instance, (Stanley, Award) is a keyword relationship as there is a path between uni1 and prize1 in Fig. 1. A database is considered relevant if its keyword relationship model covers all pairs of query keywords. M-KS considers only binary relationships between keywords. It incurs a large number of false positives for queries with more than two keywords. This is the case when all query keywords are pair wise related but there is no combined join sequence which connects all of them.

G-KS [7] addresses this problem by considering more complex relationships between keywords using a Keyword Relationship Graph (KRG). Each node in the graph corresponds to a keyword. Each edge between two nodes corresponding to the keywords (ki, kj) indicates that there exists at least two connected tuples ti ↔ tj that match ki and kj. Moreover, the distance between ti and tj are marked on the edges.

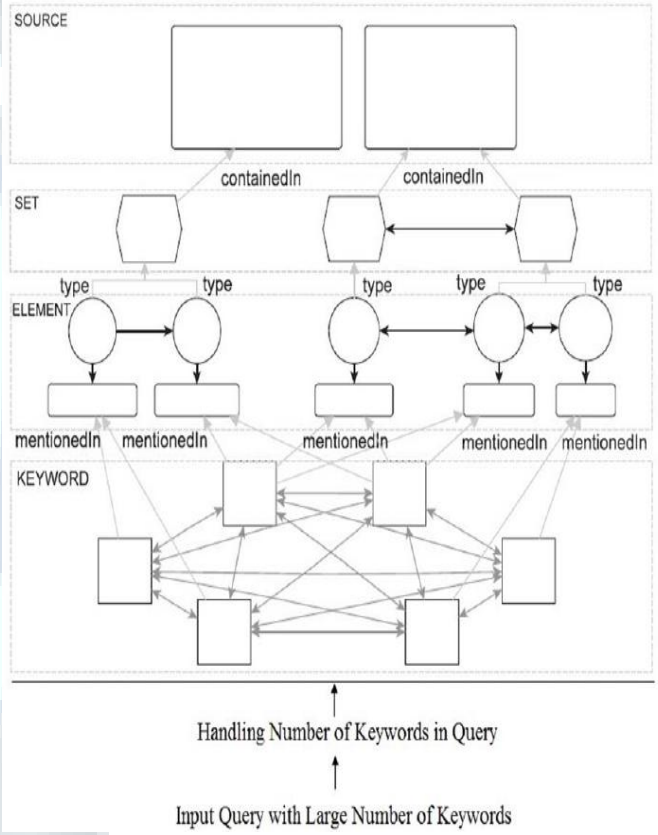

Figure 1: Inter Relationship between Elements.

However, queries with more keywords could not be handled efficiently. For instance, queries with more than two keywords needed several seconds up to one minute. Thus, while this setting produced results of highest quality, it is not really affordable in a typical web scenario demanding high responsiveness. To produce results in minimum time, while not compromising too much on quality. The results suggest that keyword search without routing is especially problematic when the number of keywords is large. Thus the proposed system uses routing keyword search for the queries having large number of keywords.

The search space of keyword query routing using a multilevel inter-relationship graph. At the lowest level, it models relationships between keywords. In the upper most levels,

there are W (N, ε) and the source-level web graph, which contains sources as nodes. The inter-relationships between elements at different levels are illustrated in Figure 1. A keyword is mentioned in some entity descriptions at the element level. Entities at the element level are associated with a set-level element via type. A set-level element is contained in a source. There is an edge between two keywords if two elements at the element level mentioning these keywords are connected via a path. Fig.1 represents a holistic view of the search space. Based on this view, we propose a ranking scheme that deals with relevance at many levels. Further, Fig. provides different perspectives on the search space. Based on this representation of the search space, existing work on keyword search and database selection can be extended to solve the problem of keyword query routing. For selecting the correct routing plan, we use graphs that are developed based on the relationships between the keywords present in the keyword query. This relationship is considered at the various levels such as keyword level, element level, set level e.t.c. The goal is to produce routing plans, which can be used to compute results from multiple sources. However, queries with more keywords could not be handled efficiently. For instance, queries with more than two keywords needed several seconds up to one minute. Thus proposed system tries to handle such queries with number of keywords and tries to minimize the computing time.

Scoring Mechanism-In addition we are showing the relevant data in result to input keyword query. For example we consider the twitter dataset which includes tweets. These keywords are searched in twitter dataset. Those tweets having these keywords, only that tweets will be shown in results. But for effective results they are ranked by scoring them for each tweet. By calculating the score of keywords for every tweet, that score is again manipulated by comparing it with the graph levels. If we do not set the score with respect to the graph, then we will get normal re-ranking without proper score. So with proper score, relevant tweets are re-ranked and efficient results are generated.

Routing Method-

Routing method routes the keyword to highly relevant data sources within some instant of time. While in keyword searching on all sources, it reduces the high cost required for query processing. Firstly in this method, the selected sources are preprocessed (pruned) then the keyword graph is generated for more relevant sources. According to the routing plan, the query including keywords is processed and delivers only the most relevant and matching information needed. As the keyword searching using other approaches is problematic when the number of keywords is large in a query. But routing method can be used for large keywords in a query because if the information need is well described then only more relevant data can be retrieved.

In our approach as per the input query keywords, the algorithm scan the entire graph from root node to leaf nodes till reaching to the all keyword. It maintains an index to store all the routes reaching to the keywords and finally shows the subtree in output result.

IV.CONCLUSION

This paper helps to improve the performance of keyword search, without compromising its result quality. Investigate the problem of keyword query routing for keyword search over a large number of structured and Linked Data sources. Routing keywords only to relevant sources can reduce the high cost of searching for structured results that span multiple sources. We use a graph-based data model to characterize individual data sources. For selecting the correct routing plan, we use graphs that are developed based on the relationships between the keywords present in the keyword query. This relationship is considered at the various levels such as keyword level, element level, set level e.t.c. In the existing system, Routing keywords return all the source which may or may not be the relevant sources.

However, queries with more keywords could not be handled efficiently. For instance, queries with more than two keywords needed several seconds up to one minute. Thus, while this setting produced results of highest quality, it is not really affordable in a typical web scenario demanding high responsiveness. To produce results in minimum time, while not compromising too much on quality. The results suggest that keyword search without routing is especially problematic when the number of keywords is large. Thus the proposed system uses routing keyword search for the queries having large number of keywords.

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. Karam-belkar, "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, Semi Structured 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 con-strained optimization". In CIKM. ACM, 2009.
[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 classiffier weights: interaction with classiffication models". In Pro-ceedings of the 27th Annual International ACM SIGIRConference SIGIR2004. ACM, 2004.
[15] Saeedeh Shekarpour, Jens Lehmann, and Sren Auer, "Keyword Query Expan-sion on Linked Data Using Linguistic and Semantic Features", IEEE Seventh International Conference on Semantic Computing, 2013.