

Routing of queries By Keywords



B.Dharma Raju
M.Tech Student,
Department of CSE,

Sree Rama institute of Technology and Science,
Kuppenakuntla, Penuballi, Khammam, TS India.



P.Karunakar Reddy
Assistant Professor,
Department of CSE,

Sree Rama institute of Technology and Science,
Kuppenakuntla, Penuballi, Khammam, TS India.

ABSTRACT:

Keyword search is an intuitive paradigm for searching linked data sources on the web. We propose to route keywords only to relevant sources to reduce the high cost of processing keyword search queries over all sources. We propose a novel method for computing top-k routing plans based on their potentials to contain results for a given keyword query. We employ a keyword-element relationship summary that compactly represents relationships between keywords and the data elements mentioning 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. Experiments carried out using 150 publicly available sources on the web showed that valid plans (precision@1 of 0.92) that are highly relevant (mean reciprocal rank of 0.89) can be computed in 1 second on average on a single PC. Further, we show routing greatly helps to improve the performance of keyword search, without compromising its result quality.

Index Terms:

Keyword search, keyword query, keyword query routing, graph-structured data, RDF.

INTRODUCTION:

THE web is no longer only a collection of textual documents but also a web of interlinked data sources (e.g., Linked Data). One prominent project that largely contributes to this development is Linking Open Data. Through this project, a large amount of legacy data have been transformed to RDF, linked with other sources, and published as Linked Data.

Collectively, Linked Data comprise hundreds of sources containing billions of RDF triples, which are connected by millions of links (see LOD Cloud illustration at <http://linkeddata.org/>). While different kinds of links can be established, the ones frequently published are sameAs links, which denote that two RDF resources represent the same real-world object. It is difficult for the typical web users to exploit this web data by means of structured queries using languages like SQL or SPARQL. To this end, keyword search has proven to be intuitive. As opposed to structured queries, no knowledge of the query language, the schema or the underlying data are needed.

In database research, solutions have been proposed, which given a keyword query, retrieve the most relevant structured results [1], [2], [3], [4], [5], or simply, select the single most relevant databases [6], [7]. However, these approaches are single-source solutions. They are not directly applicable to the web of Linked Data, where results are not bounded by a single source but might encompass several Linked Data sources. As opposed to the source selection problem [6], [7], which is focusing on computing the most relevant sources, the problem here is to compute the most relevant combinations of sources. The goal is to produce routing plans, which can be used to compute results from multiple sources.

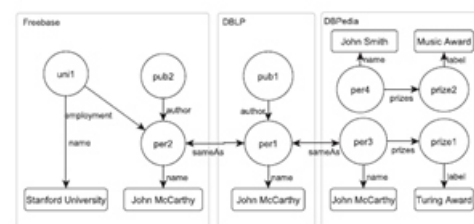


Fig. 1. Extract of the web data graph.

Existing System:

Existing work can be categorized into two main categories:

- » schema-based approaches
- » Schema-agnostic approaches

There are schema-based approaches implemented on top of off-the-shelf databases. 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. Schema-agnostic approaches operate directly on the data. 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 elements. 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 and dynamic programming. Existing work on keyword search relies on an element-level model (i.e., data graphs) to compute keyword query results.

DISADVANTAGES OF EXISTING SYSTEM:

1. The number of potential results may increase exponentially with the number of sources and links between them. Yet, most of the results may be not necessary especially when they are not relevant to the user.
2. The routing problem, we need to compute results capturing specific elements at the data level.
3. Routing keywords return all the source which may or may not be the relevant sources.

Proposed System:

We propose to route keywords only to relevant sources to reduce the high cost of processing keyword search queries over all sources. We propose a novel method for computing top-k routing plans based on their potentials to contain results for a given keyword query. We employ a keyword-element relationship summary that compactly represents relationships between keywords and the data elements mentioning 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 sub-graphs that connect these elements. We propose to investigate the problem of keyword query routing for keyword search over a large number of structured and Linked Data sources.

ADVANTAGES OF PROPOSED SYSTEM:

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.

RELATED WORK

There are two directions of work: 1) keyword search approaches compute the most relevant structured results and 2) solutions for source selection compute the most relevant sources.

Existing work can be categorized into two main categories: There are schema-based approaches implemented on top of off-the-shelf databases [8], [1], [2], [3], [9], [10]. 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. Schema-agnostic approaches [11], [12], [13], [5] operate directly on the data. 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 elements [13]. For the query “Stanford John Award” for instance, a Steiner graph is the path between uni1 and prize1 in Fig. 1. 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 [11] and dynamic programming [5]. Recently, a system called Kite extends schema-based techniques to find candidate networks in the multisourcesetting [4]. 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 precomputed links, Hermes [14] translates keywords to structured queries. However, experiments have been performed only for a small number of sources so far. Kite explicitly considered only the setting where “the number of databases that can be dealt with is up to the tens” [4].

Database Selection:

More closely related to this work are existing solutions to database selection, where the goal is to identify the most relevant databases. 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, hStanford; Awardi is a keyword relationship as there is a path between uni1 and prize1 in Fig. 1. A database is relevant if its keyword relationship model covers all pairs of query keywords. MKS [6] captures relationships using a matrix. Since 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 pairwise 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 hki; kj indicates that there exists at least two connected tuples t_i and t_j that match k_i and k_j . Moreover, the distance between t_i and t_j are marked on the edges.

OVERVIEW:

In this section, we discuss the data, define the problem, and then briefly sketch the proposed solution.

Web of Data:

We use a graph-based data model to characterize individual data sources.

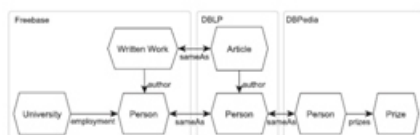


Fig. 2. Set-level web data graph.

In that model, we distinguish between an element-level data graph representing relationships between individual data elements, and a set-level data graph, which captures information about group of elements.

Definition 1 (Element-level Data Graph):

Note that this model resembles RDF data where entities stand for some RDF resources, data values stand for RDF literals, and relations and attributes correspond to RDF triples. While it is primarily used to model RDF Linked Data on the web, such a graph model is sufficiently general to capture XML and relational data. For instance, a tuple in a relational database can be modeled as an entity, and foreign key relationships can be represented as inter-entity relations.

Definition 2 (Set-level Data Graph):

This set-level graph essentially 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. In such a case, a pseudoschema can be obtained by computing a structural summary such as a dataguide [15]. A set-level data graph can be derived from a given schema or a generated pseudoschema. Thus, we assume a membership mapping type $n \in n_0$ exists and use $n \in n_0$ to denote that n belongs to the set n_0 . An example of the setlevel graph is given in Fig. 2. We consider the search space as a set of Linked Data sources, forming a web of data.

Keyword Query Routing:

We aim to identify data sources that contain results to a keyword query. In the Linked Data scenario, results might combine data from several sources:

Definition 3 (Keyword Query Result):

Typical for all keyword search approaches is the pragmatic assumption that users are only interested in compact results such that a threshold d_{max} can be used to constrain the connections to be considered. The type of Steiner graphs that is of particular interest is d_{max} -Steiner graphs $WS_{\delta} S; ESP$, where for all $n_i; n_j \in NS$, paths between n_i and n_j is of length d_{max} or less. This work also relies on this assumption to constrain the size of the search space.

CONCLUSIONS:

We have presented a solution to the novel problem of keyword query routing. Based on modeling the search-space as a multilevel inter-relationship graph, we proposed a summary model that groups keyword and element relationships at the level of sets, and developed a multilevel ranking scheme to incorporate relevance at different dimensions. The experiments showed that the summary model compactly preserves relevant information. In combination with the proposed ranking, valid plans (precision@1 $\frac{1}{4}$ 0:92) that are highly relevant (mean reciprocal rank $\frac{1}{4}$ 0:86) could be computed in 1 s on average. Further, we show that when routing is applied to an existing keyword search system to prune sources, substantial performance gain can be achieved.

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] F. Liu, C.T. Yu, W. Meng, and A. Chowdhury, "Effective Keyword Search in Relational Databases," Proc. ACM SIGMOD Conf., pp. 563-574, 2006.
- [3] 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.
- [4] 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.
- [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 Int'l Conf. Data Eng. (ICDE), pp. 836-845, 2007.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] L. Qin, J.X. Yu, and L. Chang, "Keyword Search in Databases: The Power of RDBMS," Proc. ACM SIGMOD Conf., pp. 681-694, 2009.
- [10] 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.
- [11] 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.
- [12] H. He, H. Wang, J. Yang, and P.S. Yu, "Blinks: Ranked Keyword Searches on Graphs," Proc. ACM SIGMOD Conf., pp. 305-316, 2007.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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's:

B.Dharmaraju is a student of Sree Rama Institute of Technology & Science, Kuppenakuntla, Penuballi, Khammam, TS, India. Presently he is Pursuing his M. Tech (CSE) from this college. His area of interests includes Information Security, Cloud Computing, Data Communication & Networks.

P. Karunakar Reddy is an efficient teacher, received M.Tech from JNTU Kakinada is working as an Assistant Professor in Department of C.S.E, Sree Rama Institute of Technology & Science, Kuppenakuntla, Penuballi, Khammam, TS, India. He has published many papers in both National & International Journals. His area of Interest includes Data Communications & Networks, Information Security, Database Management Systems, Computer Organization, C Programming and other advances in Computer Applications.