Implementation of a novel technique for computing top-k routing plans based on their potentials to contain results for a given keyword query.

B.Jhansiratna
M.Tech Student
Department of CSE,
St.Peter's Engineering College,
Hyderabad, TS, INDIA.

M SharadhaVaralakshmi
Assistant Professor
Department of CSE,
St.Peter's Engineering College,
Hyderabad, TS, INDIA.

S Sudeshna
Assistant Professor
Department of CSE,
St.Peter's Engineering College,
Hyderabad, TS, INDIA.

Abstract: ***Keyword search is type of search that looks for matching documents that contain one or more words specified by the user.A popular form of keywords on the web are tags which are directly visible and can be assigned by non-experts also. Index terms can consist of a word, phrase, or alphanumerical term. They are created by analyzing the document either manually with subject indexing or automatically with automatic indexing or more sophisticated methods of keyword extraction. Index terms can either come from a controlled vocabulary or be freely assigned.Keyword search is an intuitive paradigm for searching linked data sources on the web. We propose to route keywords onlyto relevant sources to reduce the high cost of processing keyword search queries over all sources. In this paper we implement 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-elementrelationship 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.***

*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, andpublished as Linked Data. Collectively, Linked Data comprise hundreds of sources containing billions of RDF triples, which are connected by millions of links. 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.

Keywords are stored in a search index. Common words like articles (a, an, the) and conjunctions (and, or, but) are not treated as keywords because it is inefficient to do so. Almost every English-language site on the Internet has the article "the", and so it makes no sense to search for it. The most popular search engine, Google removed stop words such as "the" and "a" from its indexes for several years, but then re-introduced them, making certain types of precise search possible again.

It is difficult for the typical web users to exploit this web data by means of structured queries using languages likeSQL 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 underlyingdata 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 tocompute the most relevant combinations of sources. The goal is to produce routing plans, which can be used to compute results from multiple sources.

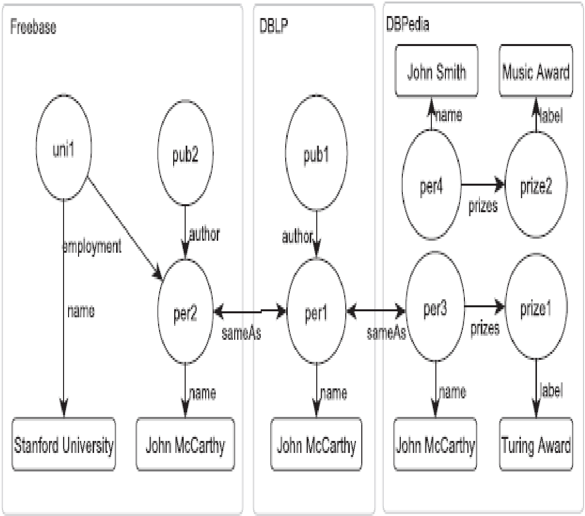


Fig. 1. Extract of the web data graph.

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.

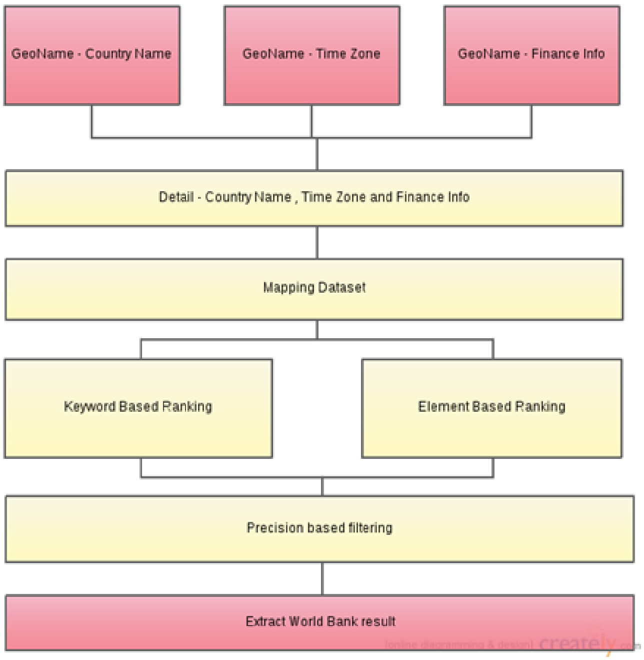
Proposed System:

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 subgraphs 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.

System Architecture:


GeoName - Country Name
GeoName - Time Zone
GeoName - Finance Info
Detail - Country Name , Time Zone and Finance Info
Mapping Dataset
Keyword Based Ranking
Element Based Ranking
Precision based filtering
Extract World Bank result
(online diagramming & design) creately.com

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 ingeneral), 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; kji indicates that there exists at least two connected tuples ti $ tj that match ki andkj. Moreover, the distance between ti and tj 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.

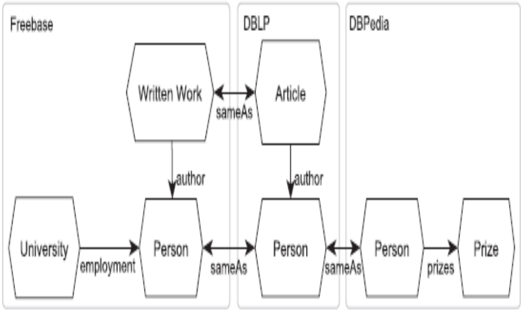


Fig. 2. Set-level web data graph.

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 RDFliterals, and relations and attributes correspond to RDF triples. While it is primarily used to model RDF LinkedData 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 interentity 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 : NE 7!N0 exists and use n 2 n0 to denote that n belongs to the set n0. 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 dmax can be used to constrain the connections to be considered. The type of Steiner graphs that is of particular interest is dmax-Steiner graphs WSðN S; ESÞ, where for all ni; nj 2 NS, paths between

ni and nj is of length dmax or less. This work also relies on this assumption to constrain the size of the search space.

Conclusion:

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 ¼ 0:92) that are highly relevant (mean reciprocal rank ¼ 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, substantialperformance gain can be achieved.

References

[1] Thanh Tran and Lei Zhang, “Keyword Query Routing”.IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 2, FEBRUARY 2014

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

[3] Y. Luo, X. Lin, W. Wang, and X. Zhou, “Spark: Top-K KeywordQuery in Relational Databases,” Proc. ACM SIGMOD Conf.,pp. 115-126, 2007.

[4] M. Sayyadian, H. LeKhac, A. Doan, and L. Gravano, “EfficientKeyword 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, “FindingTop-K Min-Cost Connected Trees in Databases,” Proc. IEEE 23rdInt’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 SIGMODConf., pp. 139-150, 2007.

[7] Q.H. Vu, B.C. Ooi, D. Papadias, and A.K.H. Tung, “A GraphMethod 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 Searchin 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: ThePower of RDBMS,” Proc. ACM SIGMOD Conf., pp. 681-694, 2009.

[10] G. Li, S. Ji, C. Li, and J. Feng, “Efficient Type-Ahead Search onRelational Data: A Tastier Approach,” Proc. ACM SIGMOD Conf.,pp. 695-706, 2009.

[11] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, andH. Karambelkar, “Bidirectional Expansion for Keyword Search onGraph 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 KeywordSearches 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 Effective3-in-1 Keyword Search Method for Unstructured, Semi-Structuredand Structured Data,” Proc. ACM SIGMOD Conf., pp. 903-914,2008.

[14] T. Tran, H. Wang, and P. Haase, “Hermes: Data Web Search on aPay-as-You-Go Integration Infrastructure,” J. Web Semantics, vol. 7,no. 3, pp. 189-203, 2009.

[15] R. Goldman and J. Widom, “DataGuides: Enabling QueryFormulation 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 JoinAlgorithms for Native Keyword Search Databases,” Proc. 20thACM Int’l Conf. Information and Knowledge Management (CIKM),pp. 1505-1514, 2011.