IJDCST @Aug-Sept-2015, Issue- V-3, I-6, SW-08
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Keyword Query Routing using top-k Routing Plans

Naraharisetty Naga Anitha[1], U Ganesh Naidu[2]

[1] M.Tech (CSE), Sri Vasavi Institute of Engineering and Technology, A.P., India.

[2]Assistant Professor, Dept. of Computer Science & Engineering, Sri Vasavi Institute of Engineering and Technology, A.P., India.

Abstract — Keyword search looks for words anywhere in the record. Keyword searches are a good substitute for a subject search when you do not know the standard subject heading. Keyword may also be used as a substitute for a title or author search when you have incomplete title or author information. You may also use the Guided Keyword search option to combine search elements, group terms, or select indexes or fields to be searched. 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 sub graphs 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.

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

I. INTRODUCTION

To search by keyword, select Keyword from the search options and type the word(s) you wish to search. Keyword searches can retrieve a large number of results. Several options are available to help refine your search and results. Quick Limits can be used when doing a keyword search. Pre-set Limits can be selected before doing a keyword search. 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 same As links, which denote that two RDF resources represent the same real-world object. A sample of Linked Data on the web is illustrated in Fig. 1. 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. To this end, we provide the following contributions: i propose to 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. To the best of our knowledge, the work presented in this paper represents the first attempt to address this problem. Existing work uses keyword relationships (KR) collected individually for single databases [6], [7]. We represent relationships between keywords as well as those between data elements. They are constructed for the entire collection of linked sources, and then grouped as elements of a compact summary called the set-level keyword-element relationship graph (KERG). Summarizing relationships is essential for addressing the scalability requirement of the Linked Data web scenario. IR-style ranking has been proposed to incorporate relevance at the level of keywords [7]. To cope with the increased keyword ambiguity in the web setting, we employ a multilevel relevance model, where elements to be considered are keywords, entities mentioning these keywords, corresponding sets of entities, relationships between elements of the same level, and inter-relationships between elements of different levels. I implemented the approach and evaluated it in a real-world setting using more than 150 publicly available data sets. The results show the applicability of this approach: valid plans (precision@1 ¼ 0.92) that are highly relevant to the user information need (mean reciprocal rank (RR) ¼ 0.86) can be computed in 1 second on average using a commodity PC. Further, we show that when routing is applied to an existing keyword search system to prune sources, substantial performance gain can be achieved.

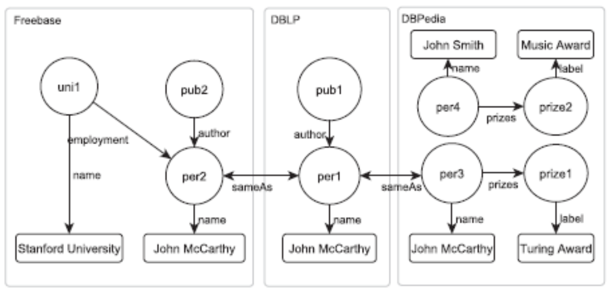

Fig. 1. Extract of the web data graph.

II PROBLEM STATEMENT

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. 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.

III 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.

Keyword Search

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 graphic 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 multisource setting [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 precompiled 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]. In our scenario, the search space drastically increases, and also, 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. A solution to keyword query routing can address these problems by pruning unpromising sources and enabling users to select combinations that more likely contain relevant results. For the routing problem, we do not need to compute results capturing specific elements at the data level, but can focus on the more coarse-grained level of sources.

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, *h* Stanford; Award *i* 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 and kj. Moreover, the distance between ti and tj are marked on the edges. Compared to M-KS, G-KS computes more relevant sources due to these differences: G-KS adopts IR-style ranking to compute TF-IDF for keywords and for keyword relationships. Further, it helps to reduce the number of false positives. It provides an additional level of filtering, validating connections between keywords based on complex relationships and distance information in the KRG. Both M-KS and G-KS assume that sources are independent and answers reside within one single source. The KRG, its keywords, relationships between keywords as well as scores are derived from and built for one single database. The solution we propose for modeling and scoring relationships is geared toward the entire Linked Data collection. Moreover, we make use of a summary, which instead of capturing relationships at the level of keywords (and data elements), it operates at the level of element sets. I am use a graph-based data model to characterize individual data sources. In that model, we distinguish between an element-level data graph representing relationships between

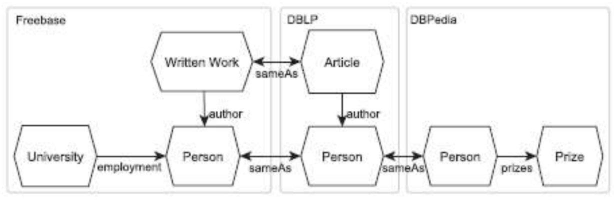

Fig. 2. Set-level web data graph.

individual data elements, and a *set-level data graph*, which captures information about group of elements.

Definition 1 (Element-level Data Graph). *An element-level data graph $g(\mathcal{N}, \mathcal{E})$ consists of*

- *the set of nodes $\mathcal{N}$, which is the disjoint union of $\mathcal{N}_\mathcal{E} \uplus \mathcal{N}_\mathcal{V}$, where the nodes $\mathcal{N}_\mathcal{E}$ represent entities and the nodes $\mathcal{N}_\mathcal{V}$ capture entities' attribute values, and*
- *the set of edges $\mathcal{E}$, subdivided by $\mathcal{E} = \mathcal{E}_\mathcal{R} \uplus \mathcal{E}_\mathcal{A}$, where $\mathcal{E}_\mathcal{R}$ represents interentity relations, $\mathcal{E}_\mathcal{A}$ stands for entity-attribute assignments. We have $e(n_1, n_2) \in \mathcal{E}_\mathcal{R}$ iff $n_1, n_2 \in \mathcal{N}_\mathcal{E}$ and $e(n_1, n_2) \in \mathcal{E}_\mathcal{A}$ iff $n_1 \in \mathcal{N}_\mathcal{E}$ and $n_2 \in \mathcal{N}_\mathcal{V}$.*

The set of attribute edges $\mathcal{E}_\mathcal{A}(n) = \{e(n,m) \in \mathcal{E}_\mathcal{A}\}$ is referred to as the description *of the entity n.*

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 tupelo 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). *A set-level data graph of an element-level graph $g(\mathcal{N}_\mathcal{E} \uplus \mathcal{N}_\mathcal{V}, \mathcal{E}_\mathcal{R} \uplus \mathcal{E}_\mathcal{A})$ is a tuple $g' = (\mathcal{N}', \mathcal{E}')$. Every node $n' \in \mathcal{N}'$ stands for a set of element-level entities $\mathcal{N}_{n'} \subseteq \mathcal{N}_\mathcal{E}$, i.e., there is mapping* $\text{type} : \mathcal{N}_\mathcal{E} \mapsto \mathcal{N}'$ *that associates every element-level entity $n \in \mathcal{N}_\mathcal{E}$ with a set-level element $n' \in \mathcal{N}'$. Every edge $e'(n'_i, n'_j) \in \mathcal{E}'$ represents a relation between the two sets of element-level entities n'_i and n'_j. We have $\mathcal{E}' = \{e'(n'_i, n'_j) | e(n_i, n_j) \in \mathcal{E}_\mathcal{R}, \text{type}(n_i) = n'_i, \text{type}(n_j) = n'_j\}$.*

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 pseudo schema can be obtained by computing a structural summary such as a data guide [15]. A set-level data graph can be derived from a given schema or a generated pseudo schema. An example of the set level graph is given in Fig. 2.We consider the search space as a set of Linked Data sources, forming a web of data.

TABLE 1
Notation

Symbols	Description
$k, \mathcal{K}$	keyword (term), keyword query
$\mathcal{N}, \mathcal{N}_{\mathcal{E}}, \mathcal{N}_{\mathcal{V}}$	graph, entity and attribute value nodes
$\mathcal{E}, \mathcal{E}_{\mathcal{R}}, \mathcal{E}_{\mathcal{A}}$	graph, relation and attribute edges
$\mathcal{E}_{\mathcal{A}}(n)$	description of entity n
$g, \mathcal{G}$	graph, set of graphs
$\mathcal{W}^*; \mathcal{W}, \mathcal{W}'$	Web graph; element-level and set-level Web graph
$\mathcal{W}'_{\mathcal{K}}, n'_{\mathcal{K}}, e'_{\mathcal{K}}$	set-level keyword-element relationship graph, node and edge
$\mathcal{W}^S$	Steiner graph (keyword query result)
$\mathcal{W}'^S_{\mathcal{K}}$	routing graph
$\mathcal{RP}$	routing plan
$\mathcal{M}^{\mathcal{E}}_g(d)$	matrix capturing paths in g of length d
$\mathcal{M}^{\mathcal{E}_e}_{(g_i,g_j)}(d)$	matrix capturing paths between g_i and g_j of length d
$tf(k_i, n'_{\mathcal{K}})$	frequency (count) of k_i in $n'_{\mathcal{K}}$

source, and $\mathcal{E}^*_e$ is the set of all "external" edges, which establish links between elements of two different sources, i.e., $\mathcal{G}^* = \{g_1(\mathcal{N}^*_1, \mathcal{E}^*_1), g_2(\mathcal{N}^*_2, \mathcal{E}^*_2), \ldots, g_n(\mathcal{N}^*_n, \mathcal{E}^*_n)\}, \mathcal{N}^* = \bigcup^n_{l=1} \mathcal{N}^*_l$, $\mathcal{E}^*_e = \{e(n_i, n_j) | n_i \in \mathcal{N}^*_i, n_j \in \mathcal{N}^*_j, \mathcal{N}^*_i \neq \mathcal{N}^*_j\}$, *and* $\mathcal{E}^* = \bigcup^n_{l=1} \mathcal{E}^*_{l_l} \cup \mathcal{E}^*_{e_l}$. *When considering the nodes and edges only, we simply use* $\mathcal{W}^*(\mathcal{N}^*, \mathcal{E}^*)$. *We use* $\mathcal{W}(\mathcal{G}, \mathcal{N}, \mathcal{E})$ *to distinguish the* element-level web graph *from the* set-level web graph $\mathcal{W}'(\mathcal{G}', \mathcal{N}', \mathcal{E}')$.

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 4 (Keyword Query Result). *A web graph $\mathcal{W}(\mathcal{N}, \mathcal{E})$ contains a result for a query $\mathcal{K} = \{k_1, k_2, \ldots, k_{|\mathcal{K}|}\}$ if there is a subgraph also called* Steiner graph *$\mathcal{W}^S(\mathcal{N}^S, \mathcal{E}^S)$, which for all $k_i \in \mathcal{K}$, contains a* keyword element *node $n_i \in \mathcal{N}^S$ whose description $\mathcal{E}_{\mathcal{A}}(n_i)$ matches k_i, and there is a path between n_i and n_j ($n_i \leftrightarrow n_j$) for all $n_i, n_j \in \mathcal{N}^S$.*

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* $\mathcal{W}^S(\mathcal{N}^S, \mathcal{E}^S)$, where for all $n_i, n_j \in \mathcal{N}^S$, 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.

Definition 5 (Keyword Routing Plan). *Given the web graph $\mathcal{W} = (\mathcal{G}, \mathcal{N}, \mathcal{E})$ and a keyword query $\mathcal{K}$, the mapping $\mu : \mathcal{K} \rightarrow 2^{\mathcal{G}}$ that associates a query with a set of data graphs is called a* keyword routing plan *$\mathcal{RP}$. A plan $\mathcal{RP}$ is considered* valid *w.r.t. $\mathcal{K}$ when the union set of its data graphs contains a result for $\mathcal{K}$.*

The problem of keyword query routing is to find the top-k keyword routing plans based on their relevance to a query. A relevant plan should correspond to the information need as intended by the user. Table 1 provides an overview of all symbols.

IV Conclusion

Keyword search tools are supposed to help you reach potential customers by telling you how they search for what you're offering. I 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 = 0.92) that are highly relevant (mean reciprocal rank @= 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 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 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.