Available online at http://www.journalcra.com

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

International Journal of Current Research
Vol. 3, Issue, 4, pp.030-035, April, 2011

ISSN: 0975-833X

REVIEW ARTICLE

DATA SCRUBBING ON SEARCH ENGINE USING XML WITH DUPLICATE ELIMINATION

Dhenakaran, S.S. and *Mathi, R

Department of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu, India

ARTICLE INFO

Article History:

Received 12th January, 2011
Received in revised form
20th February, 2011
Accepted 5th March, 2011
Published online 17th April 2011

Key words:

Information retrieval
XML
Text database
Intention
Ambiguity problems

ABSTRACT

Inspired by the great success of information retrieval (IR) style keyword search on the web, keyword search on XML has emerged recently. The difference between text database and XML database results in three new challenges: 1) Identify the user search intention, i.e., identify the XML node types that user wants to search for and search via. 2) Resolve keyword ambiguity problems: a keyword can appear as both a tag name and a text value of some node; a keyword can appear as the text values of different XML node types and carry different meanings; a keyword can appear as the tag name of different XML node types with different meanings. 3) As the search results are subtrees of the XML document, new scoring function is needed to estimate its relevance to a given query. However, existing methods cannot resolve these challenges, thus return low result quality in term of query relevance. In this paper, we propose an IR-style approach which basically utilizes the statistics of underlying XML data to address these challenges. We first propose specific guidelines that a search engine should meet in both search intention identification and relevance oriented ranking for search results. Then, based on these guidelines, we design novel formulae to identify the search for nodes and search via nodes of a query, and present a novel XML TF*IDF ranking strategy to rank the individual matches of all possible search intentions. To complement our result ranking framework, we also take the popularity into consideration for the results that have comparable relevance scores. Lastly, extensive experiments have been conducted to show the effectiveness of our approach.

Corresponding author:
ssdarvind@yahoo.com; mathi.mphil@gmail.com

INTRODUCTION

The extreme success of web search engines makes keyword search the most popular search model for ordinary users. As XML is becoming a standard in data representation, it is desirable to support keyword search in XML database. It is a user friendly way to query XML databases since it allows users to pose queries without the knowledge of complex query languages and the database schema. Effectiveness in terms of result relevance is the most crucial part in keyword search, which

can be summarized as the following three issues in XML field:

Issue 1: It should be able to effectively identify the type of target node(s) that a keyword query intends to search for. We call such target node as a search for node.

Issue 2: It should be able to effectively infer the types of condition nodes that a keyword query intends to search via. We call such condition nodes as search via nodes.

Issue 3: It should be able to rank each query result in consideration of the above two issues.

The first two issues address the search intention problem, while the third one addresses the relevance-based ranking problem w.r.t. the search intention. Regarding to Issue 1 and Issue 2, XML keyword queries usually have ambiguities in interpreting the search for node(s) and search via node(s), due to three reasons.

- Ambiguity 1: A keyword can appear both as an XML tag name and as a text value of some other nodes.
- Ambiguity 2: A keyword can appear as the text values of different types of XML nodes and carry different meanings.
- Ambiguity 3: A keyword can appear as an XML tag name in different contexts and carry different meanings.

For example, see the XML document in Fig. 1, keywords customer and interest appear as both an XML tag name and a text value (e.g., value of the title for book B1); art appears as a text value of interest, address, and name node; name appears as the tag name of the name of both customer and publisher. Regarding to Issue 3, the search intention for a keyword query is not easy to determine and can be ambiguous, because the search via condition is not unique; so, how to measure the confidence of each search intention candidate, and rank the individual matches of all these candidates are challenging. In particular, regarding to Issues 1 and 2, SLCA may introduce answers that are either irrelevant to user search intention, or answers that may not be meaningful or informative enough. For example, when a query "Jim Gray" that intends to find Jim Gray's publications on DBLP [10] is issued, SLCA returns only the author elements containing both keywords. Besides, SLCA also returns publications written by two authors where "Jim" is a term in first author's name and "Gray" is a term in second author, and publications with title containing both keywords.

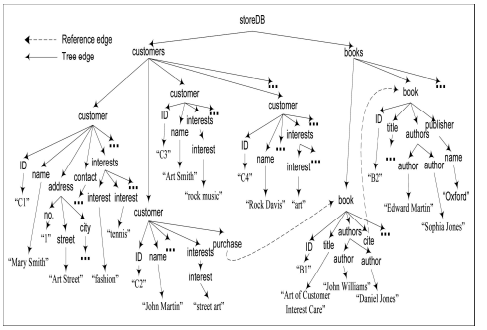

Fig. 1. XML Document

It is reasonable to return such results because search intention may not be unique; however, they should be given a lower rank, as they are not matches of the major search intention. Regarding to Issue 3, no existing approach has studied the problem of relevance oriented result ranking in depth yet. Moreover, they don't perform well on pure keyword query when the schema information of XML data is not available [14]. The rest of the paper is organized as follows: We present the related work in Section 2 and data model in Section 3. Section 4 discusses the ranking scheme. Experiment is discussed in Section 5, and we conclude in Section 6.

RELATED WORK

Extensive research efforts have been conducted in XML keyword search to find the smallest substructures in XML data that each contains all query keywords in either the tree data model or the directed graph (i.e., digraph) data model. In tree data model, lowest common ancestor (LCA) semantics is first proposed and studied in [17], to find XML nodes, each of which contains all query keywords within its subtree. Subsequently, SLCA (smallest LCA [13], [20]) is proposed to find the smallest LCAs that do not contain other LCAs in their subtrees. GDMCT (minimum connecting trees) [7] excludes the subtrees rooted at the LCAs that do not contain the query keywords. Sun et al.

[18] generalize SLCA to support keyword search involving combinations of AND and OR boolean operators. XSeek [14] generates the return nodes which can be explicitly inferred by keyword match pattern and the concept of entities in XML data. However, it addresses neither the ranking problem nor the keyword ambiguity problem. Besides, it relies on the concept of entity (i.e., object class) and considers a node type t in DTD as an entity if t is "*"-annotated in DTD. As a result, customer, phone, interest, and book in Fig. 1 are identified as object classes by XSeek. However, it causes the multivalued attribute to be mistakenly identified as an entity, causing the inferred return node not as intuitive as possible. For example, phone and interest are not intuitive as entities. In fact, the identification of entity is highly dependent on the semantics of underlying database rather than its DTD, so it usually requires the verification and decision from database administrator. Liu and Chen [15] propose an axiomatic way to judge the completeness and correctness of a certain keyword search semantics.

In digraph data model, previous approaches are heuristics based, as the reduced tree problem on graph is as hard as NP-complete. BANKS [6] uses bidirectional expansion heuristic algorithms to search as small portion of graph as possible. BLINKS [9] proposes a bilevel index to prune and accelerate searching for top-k results in digraphs. Cohen et al. [3] study the computation complexity of interconnection semantics. XKeyword [8] provides keyword proximity search that conforms to an XML schema; however, it needs to compute candidate networks and, thus, is constrained by schemas.

On the issue of result ranking, XRANK extends Google's PageRank to XML element level, to rank among the LCA results; but no empirical study is done to show the effectiveness of its ranking function. XSEarch adopts a variant of LCA, and combines a simple tf*idf IR ranking with size of the tree and the node relationship to rank results; but it requires users to know the XML schema information, causing limited query flexibility. EASE [12] combines IR ranking and structural compactness based DB ranking to fulfill keyword search on heterogenous data. Regarding to ranking methods, TF*IDF similarity [16] which is originally designed for flat document retrieval is insufficient for XML keyword search due to XML's hierarchical structure and the presence of Ambiguity 1-3.Several proposals for XML information retrieval suggest to extend the existing XML query languages [4], [1], [19] or use XML fragments [2] to explicitly specify the search intention for result retrieval and ranking.

PRELIMINARIES

*A.TF*IDF Cosine Similarity*

TF*IDF (Term Frequency * Inverse Document Frequency) similarity is one of the most widely used approaches to measure the relevance of keywords and document in keyword search over flat documents. We first review its basic idea, then address its limitations for keyword search in XML. The main idea of TF*IDF is summarized in the following three rules:

Rule 1: A keyword appearing in many documents should not be regarded as being more important than a keyword appearing in a few.
Rule 2: A document with more occurrences of a query keyword should not be regarded as being less important for that keyword than a document that has less.
Rule 3: A normalization factor is needed to balance between long and short documents, as Rule 2 discriminates against short documents which may have less chance to contain more occurrences of keywords.

B. Data Model:

We model XML document as a rooted, labeled tree plus a set of directed IDRef edges between XML nodes, such as the one in Fig. 1. In contrast to general directed graph model, the containment edge and IDRef edge are distinguished in our model. Our approach exploits the prefix path of a node rather than its tag name for result retrieval and ranking. Note that the existing works [14], [11] rely on DTD while our approach works without any XML schema information.

Definition 3.1 (Node Type). The type of a node n in an XML document is the prefix path from root to n. Two nodes are of the same node type if they share the same prefix path. In Definition 3.1, the reason that two nodes need to share the same prefix path instead of their tag name is, there may be two or more nodes of the same tag name but of different semantics (i.e., in different contexts) in one document. For example, in Fig. 1, the name of publisher and the name of customer are of different node types, which are storeDB/ books/book /publisher/name and store DB/ customers / customer /name, respectively. Besides, when XML database contains multiple XML documents, the node type should also include the file name. To facilitate our discussion later, we use the tag name instead of the prefix path of a node to denote the node type in all examples throughout this paper. Besides, in order to separate the content part from leaf node, we distinguish an XML node into either a data node or a structural node.

Definition 3.2 (Data Node). The text values that are contained in the leaf node of XML data and have no tag name are defined as data node.

Definition 3.3 (Structural Node). An XML node labeled with a tag name is called a structural node. A structural node that contains other structural nodes as its children is called an internal node; otherwise, it is called a leaf node.

C. Capturing Keyword Co-Occurrence

In this section, we discuss the search via confidence for a data node. Although statistics provide a macro way to compute the confidence of a structural node type to search via, it alone is not adequate to infer the likelihood of an individual data node to search via for a given keyword in the query. Example 6. Consider a query "customer name Rock interest Art" searching for customers whose name includes "Rock" and interest includes "Art." Based on statistics, we can infer that name-typed and interest-typed nodes have high confidence to search via by (7), as the frequency of keywords "name" and "interest" are high in node types name and interest, respectively. However, statistics is not adequate to help the system infer that the user wants "Rock" to be a value of name and "Art" to be a value of interest, which is intuitive with the help of keyword co-occurrence captured. Thus, if purely based on statistics, it is difficult for a search engine to differ customer C4 (with name "Art" and interest "Rock") from C3 (with name "Rock" and interest "Art") in Fig. 1.

RELEVANCE ORIENTED RANKING

A. Principles of Keyword Search in XML

Compared with flat documents, keyword search in XML has its own features. In order for an IR-style ranking approach to smoothly apply to it, we present three principles that the search engine should adopt.

Principle 1: When searching for XML nodes of desired type D via a single-valued node type V , ideally, only the values and structures nested in V -typed nodes can affect the relevance of D-typed nodes as answers, whereas the existence of other typed nodes nested in D-typed nodes should not. In other words, the size of the subtree rooted at a D-typed node d (except the subtree rooted at the search via node) shouldn't affect d's relevance to the query.

Principle 2: When searching for the desired node type D via a multivalued node type V 0, if there are many V 0-typed nodes nested in one node d of type D, then the existence of one query-relevant node of type V 0 is usually enough to indicate, d is more relevant to the query than another node d0 also of type D but with no nested V 0-typed nodes containing the keyword(s). In other words, the relevance of a D-typed node which contains a query-relevant V 0-typed node should not be affected (or normalized) too much by other query-irrelevant V 0-typed nodes.

Principle 3: The proximity of keywords in a query is usually important to indicate the search intention.

*B. Advantages of XML TF*IDF*

Compatibility: The XML TF*IDF similarity can work on both semistructured and unstructured data, because unstructured data is a simpler kind of

semistructured data with no structure, and XML TF*IDF ranking (9a) for data node can be easily simplified to the original TF*IDF (1) by ignoring the node type.

Robustness. Unlike existing methods which require a query result to cover all keywords [14], [20], [7], we adopt a heuristic-based approach that does not enforce the occurrence of all keywords in a query result; instead, we rank the results according to their relevance to the query. In this way, more relevant results can be found, because a user query may often be an imperfect description of his real information need [5]. Users never expect an empty result to be returned even though no result can cover all keywords; fortunately, our approach is still able to return the most relevant results to users.

RESULT AND DISCUSSION

To evaluate the effectiveness of XML TF*IDF alone, we use three measures widely adopted in IR field. 1) Number of top-1 answers that are relevant. 2) Reciprocal rank (R-rank). For a given query, the reciprocal rank is 1 divided by the rank at which the first correct answer is returned, or 0 if no correct answer is returned. 3) Mean Average Precision (MAP). A precision is computed after each relevant answer is retrieved, and MAP is the average value of such precisions. The first two measure how good the system returns one relevant answer, while the third one measures the overall effectiveness for top-k

Table 1. Ranking Performance of XReal

Dataset	*Top-1 Number/Total Number*	*R-Rank*	*MAP*
DBLP	27/30	0.946	0.925
WSU	8/10	0.85	0.803
eBay	9/10	0.9	0.867
XMark	7/10	0.791	0.713

answers returned, k ¼ 40 for DBLP (as DBLP data has very large size) and k ¼ 20 for others (if they do exist).

We evaluate a set of 30 randomly generated queries on DBLP, and 10 queries on WSU, eBay, and XMark, with an average of three keywords. The average values of these metrics are recorded in Table 3. We find XReal has an average R-rank greater than 0.8 and even over 0.9 on DBLP. Besides, XReal returns the relevant result in its top-1 answer in most queries, which shows high effectiveness of our ranking strategy.

CONCLUSION

In this paper, we study the problem of effective XML keyword search which includes the identification of user search intention and result ranking in the presence of keyword ambiguities. We utilize statistics to infer user search intention and rank the query results. In particular, we define XML TF and XML DF, based on which we design formulae to compute the confidence level of each candidate node type to be a search for/search via node, and further propose a novel XML TF*IDF similarity ranking scheme to capture the hierarchical structure of XML data. Lastly, the popularity of a query result (captured by IDRef relationships) is considered to handle the case that multiple results have comparable relevance scores. In future, we would like to extend our approach to handle the XML document conforming to a highly recursive schema as well.

REFERENCES

[1] S. Amer-Yahia, L.V.S. Lakshmanan, and S. Pandit,"Flexpath: Flexible Structure and Full-Text Querying for XML," Proc. ACM SIGMOD Conf., 2004.

[2] D. Carmel, Y.S. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer, "Search XML Documents via XML Fragments," Proc. ACM SIGIR, pp. 151-158, 2003.

[3] S. Cohen, Y. Kanza, B. Kimelfeld, and Y. Sagiv, "Interconnection Semantics for Keyword Search in XML," Proc. ACM Int'l Conf. Information and Knowledge Management (CIKM), pp. 389-396, 2005.

[4] N. Fuhr and K. Großjohann, "XIRQL: A Query Language for Information Retrieval in XML Documents," Proc. ACM SIGIR, pp. 172-180, 2001.

[5] R. Jones, B. Rey, O. Madani, and W. Greiner, "Generating Query Substitutions," Proc. Int'l Conf. World Wide Web (WWW), 2006.

[6] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar,

"Bidirectional Expansion for Keyword Search on Graph Databases," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 505-516, 2005.

[7] V. Hristidis, N. Koudas, Y. Papakonstantinou, and D. Srivastava, "Keyword Proximity Search in XML Trees," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 4, pp. 525-539, Apr. 2006.

[8] V. Hristidis, Y. Papakonstantinou, and A. Balmin, "Keyword Proximity Search on XML Graphs," Proc. IEEE Int'l Conf. Data Eng.(ICDE), pp. 367-378, 2003.

[9] H. He, H. Wang, J. Yang, and P.S. Yu, "Blinks: Ranked Keyword Searches on Graphs," Proc. ACM SIGMOD Conf., pp. 305-316, 2007.

[10] M. Ley DBLP, http://www.informatik.uni-trier.de/ley/db/, 2009.

[11] G. Li, J. Feng, J. Wang, and L. Zhou, "Effective Keyword Search for Valuable LCAs over XML Documents," Proc. ACM Int'l Conf. Information and Knowledge Management (CIKM), pp. 31-40, 2007.

[12] G. Li, B.C. Ooi, J. Feng, J. Wang, and L. Zhou, "Ease: Efficient and Adaptive Keyword Search on Unstructured, Semi-Structured and Structured Data," Proc. ACM SIGMOD Conf., 2008.

[13] Y. Li, C. Yu, and H.V. Jagadish, "Schema-Free XQuery," Proc. Int'l Conf. Very Large Data Bases (VLDB), 2004.

[14] Z. Liu and Y. Chen, "Identifying Meaningful Return Information for XML Keyword Search," Proc. ACM SIGMOD Conf., 2007.

[15] Z. Liu and Y. Chen, "Reasoning and Identifying Relevant Matches for XML Keyword Search," Proc. Int'l Conf. Very Large Data Bases (VLDB) vol. 1, no. 1, pp. 921-932, 2008.

[16] G. Salton and M.J. McGill, Introduction to Modern Information Retrieval. McGraw-Hill, Inc., 1986.

[17] A. Schmidt, M.L. Kersten, and M. Windhouwer, "Querying XML Documents Made Easy: Nearest Concept Queries," Proc. IEEE Int'l Conf. Data Eng. (ICDE), pp. 321-329, 2001.

[18] C. Sun, C.Y. Chan, and A.K. Goenka, "Multiway SLCA-Based Keyword Search in XML Data," Proc. Int'l Conf. World Wide Web (WWW), pp. 1043-1052, 2007.

[19] A. Theobald and G. Weikum, "The Index-Based XXL Search Engine for Querying XML Data with Relevance Ranking," Proc. Int'l Conf. Extending Database Technology (EDBT), 2002.

[20] Y. Xu and Y. Papakonstantinou, "Efficient Keyword Search for Smallest LCAs in XML Databases," Proc. ACM SIGMOD, pp. 537-538, 2005.

****