Global Journal of Computer Science and Technology
Volume 11 Issue 16 Version 1.0 September 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

An Effective XML Keyword Search with User Search Intention over XML Documents

By Pradeep Kumar Reddy Gade, N Prasanna Balaji, U Sreenivasulu
Ibrahimpatnam, Andhra Pradesh, India

Abstract - The extreme success of web search engines makes keyword search the most popular search model for ordinary users. Keyword search on XML 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. The three main challenges faces in XML keyword search: 1) Identify the user search intention, i.e., identify the XML node types that users want 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 sub trees of the XML documents, 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 rankingfor search results over XML documents. Then, based on theseguidelines, we design novel formulae to identify the search fornodes and search via nodes of a query, and present a novelXML TF*IDF ranking strategy to rank the individual matches ofall possible search intentions over XML documents.

Keywords : *XML, keyword search, ranking.*

GJCST Classification : *H.2.8, D.2.9*

Strictly as per the compliance and regulations of:

An Effective XML Keyword Search with User Search Intention over XML Documents

Pradeep Kumar Reddy Gade[α], N Prasanna Balaji[Ω], U Sreenivasulu[β]

Abstract - The extreme success of web search engines makes keyword search the most popular search model for ordinary users. Keyword search on XML 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. The three main challenges faces in XML keyword search: 1) Identify the user search intention, i.e., identify the XML node types that users want 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 sub trees of the XML documents, 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 over XML documents. 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 over XML documents.

Keywords : XML, keyword search, ranking.

I. Introduction

The extreme success of web search engines makes keyword search the most popular search model for ordinary users. In the real world, computer systems and databases contain data in incompatible formats. XML data is stored in plain text format. This provides a software- and hardware-independent way of storing data. AsXML 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.

Author [α] : 2nd year M-tech, GuruNanak Engineering College, Ibrahimpatnam, Andhra Pradesh, India. Email : reddys_gp@yahoo.com
Author [Ω]: Professor & HOD(IT), GuruNanak Engineering College, Ibrahimpatnam, Andhra Pradesh, India.
Email : gneccsebalaji@gmail.com
Author [β] : Asst. Professor, GuruNanak Engineering College, Ibrahimpatnam, Andhra Pradesh, India.
Email : ulsa535@gmail.com

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&2 : Capture user's search intention.
i) Identify the target that user intends to search for.
ii) Infer the predicate constraint that user intends to search via.

Issue 3 : Result ranking.
i) Ranking the query results according to their objective relevance to user search intention.

Issues 1&2 addresses the search intention problem, while the third one addresses the relevance-based ranking problem w.r.t. the search intention. The search intention for a keyword query is not easy to determine and can be ambiguous, because the search via condition is not unique. While performing keyword search on XML database, three Ambiguities arises. They are:

- 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.

Although many research efforts have been conducted in XML keyword search [8] [10] [29] [22][23], none of them has been addressed and resolved the above three issues in the presence of ambiguities. So far some efforts have been conducted to satisfy the user search intention but none of them addressed relevance oriented result ranking in depth.

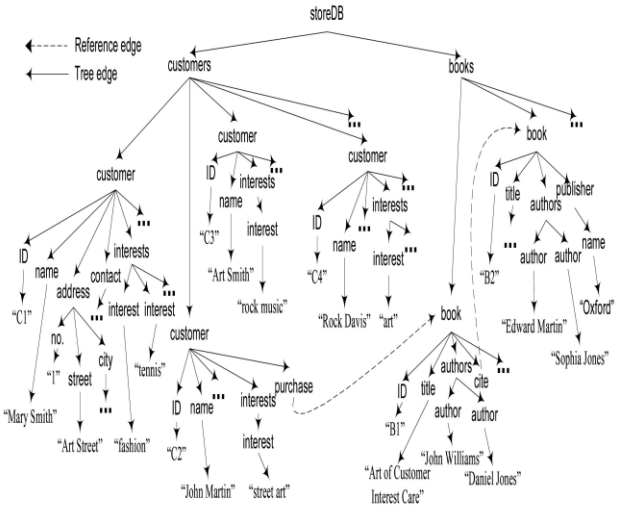

Fig. 1 : Portion of data tree for an online bookstore XML database.

Consider a keyword query "Customer name martin". The user search intention is to find the customers whose name is martin. By XML keyword search we will get two results C2 and B2 who has the keyword martin.

Even though B2 contains the name martin the XML search engine XReal give only C2 because we are searching for customer whose name is martin not the author name. So, C2 is relevant data and B2 is irrelevant data. Finally the main objective of this paper is to catch the user search intention and ranking the results in the presence of keyword ambiguities over multiple XML databases.

II. Related Work

Although many efforts have been conducted to find smallest substructures in XML data that each contains all query keywords in tree data or digraph data model. In tree data model, at first lowest common ancestor [17] (LCA) semantics is proposed to find XML nodes, each of which contains all query keywords within their subtree. Subsequently, Smallest LCA (SLCA [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 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 inferred by keyword match pattern and the concept of entities in XML data which neither addresses the ranking problem nor keyword ambiguity problem. 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. 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] propose 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[4] also extends the notion of PageRank to XML data, 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 heterogeneous 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.

III. Preliminaries

a) *Your 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

The data model for XML is very simple - or very abstract, depending on one's point of view. XML provides no more than a baseline on which more complex models can be built.

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.

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.

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.

Definition 3.4 (Single-Valued Type): A structural node t is of single-valued type if each node of type t has at most one occurrence within its parent node.

Definition 3.5 (Multivalued Type) : A structural node t is of multivalued type if some node of type t has more than one occurrence within its parent node.

Definition 3.6 (Grouping Type) : An internal node t is defined as a grouping type if each node of type t contains child nodes of only one multivalued type.

Single-valued type and multivalued type of XML nodes can be easily identified when parsing the data. Every multivalued node has a grouping node as its parent and a grouping node is also a single-valued node. Thus, the children of an internal node are either of same multivalued type or of different single-valued types. An internal node n contains both data nodes and structural nodes.

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.

IV. Inferring Keyword Search Intention

In this section, we discuss how to interpret the search intentions of keyword query according to the statistics in XML data and the pattern of keyword co-occurrence in a query.

a) Inferring the Node Type to Search for

The desired node type to search for is the first issue that a search engine needs to address in order to retrieve the relevant answers, as the search target in a keyword query may not be specified explicitly like in structured query language. Given a keyword query q, a node type T is considered as the desired node to search for only if the following three guidelines hold:

Guideline 1 : T is intuitively related to every query keyword in q, i.e., for each keyword k, there should be some (if not many) T-typed nodes containing k in their subtrees.

Guideline 2 : XML nodes of type T should be informative enough to contain enough relevant information.

Guideline 3 : XML nodes of type T should not be overwhelming to contain too much irrelevant information.

b) Inferring the Node Types to Search via

Similar to inferring the desired search for node, Intuition 1 is also useful to infer the node types to search via. However, unlike the search for case which requires a node type to be related to all keywords, it is enough for a node type to have high confidence as the desired search via node if it is closely related to some (not necessarily all) keywords, because a query may intend to search via more than one node type. For example, we can search for customer(s) named "Smith" and interested in "fashion" with query "name smith interest fashion." In this case, the system should be able to infer with high confidence that name and interest are the node types to search via, even if keyword "interest" is probably not related to name nodes.

V. 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 semi-structured and unstructured data, because unstructured data is a simpler kind of semi-structured 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.

c) XML keyword search over xml documents

The main objective of XReal search engine is to capture users search intention and relevance ranking the results in the presence of keyword ambiguity problems mentioned above. In these paper, an algorithms is used for searching a keyword in folder (having recursive folders containing xml databases) containing different xml databases.

For example, an xml database maintaining particular database for each academic year, then XReal search engine is used.

The important steps followed are:

Step 1 : Searching for keywords in every database and collecting list of databases containing the keywords.

Step 2 : keyword search by applying search for and search via node for an individual database.

Step 3 : Appling XML TF*IDF similarity on the results obtained for an individual database.

Algorithm. RecurrsivePath()

1. *Let FolderSearch = True, Result[] = Null, RecursiveSearch= True*
2. *If (FolderSearch)*
3. *ScanDir(FolderPath, RecursiveSearch)*

Function ScanDir(FolderPath, RecursiveSearch)

1. *Files = GetFiles(StartingPath)*
2. *foreach f ∈ Files*
3. *If (KWSearch(Q[m], IL[m], F[m]))*
4. *Result = XMLFileListItem(filename)*
5. *If (RecursiveSearch)*
6. *Folders = GetDirectories(StartingPath)*
7. *foreach f ∈ Folders*
8. *ScanDir (f, RecurresiveSearch)*

Algorithm. KWSearch(Q[m], IL[m], F[m]) [21] is used for keyword search in individual xml keywords.

VI. 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.

Refrences Refrences Refrencias

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, 034 International Journal of Current Research, Vol. 33, Issue, 4, pp.030-035, April, 2011 "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.unitrier.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. 035 International Journal of Current Research, Vol. 33, Issue, 4, pp.030-035, April, 2011.
21. Zhifeng Bao, Jiaheng Lu, Tok Wang Ling, Senior Member, IEEE, and Bo Chen, "Towards an Effective XML Keyword Search," Proc. IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No. 8, August-2010.

This page is intentionally left blank