Enhanced Coherency Technique for XML Keyword Search-A Review

Shabnam, Sumit Kumar Yadav

Abstract*— Keyword search techniques which use advantages of XML structure make it simpler for ordinary users to query XML databases, but latest approaches to processing these queries depend on heuristics that are ultimately ad hoc. These approaches often retrieve not correct answers, overlook appropriate answers, and cannot rank answers properly. To remove these problems for data-centric XML, we propose enhanced coherency ranking, a domain and database design-independent ranking method for XML keyword queries that is based on an extension of the concept of mutual information. Keyword search is widely recognized as a best way to retrieve information from XML data. In order to specifically meet users search requirements, we proof how to effectively return the targets that users intend to search for. We mold XML document as a set of interconnected object-trees, where each object contains a sub tree to represent a concept in the real world. The work focuses on study and performance evaluation of these categories using MATLAB 7.14.

Index Terms*— XML, DATABASE, DATA MINING, Enhanced Coherency.

I. INTRODUCTION

One of the goals of XML is to present document details and we might want to perform a keyword search on particular elements. XML and HTML are parts of SGML (Standard Generalized Markup Language). The goal of this thesis is to combine database-style query language with free-text search. The extreme advantages of web search engines makes keyword search the most important search model for ordinary users. As XML is becoming popular in data representation, it is desirable to support keyword search in XML database. It is a user friendly approach to query XML databases since it allows users to present queries without the knowledge of complicated query languages and the database schema. Many users of XML databases are not familiar with concepts such as schemas and query languages. Keyword search has been proposed as an appropriate interface for them; the challenge is how to find the data most closely related to the user's query, as the query is not framed in terms of the data's actual structure. Ideally, the query answer must include all portions of the data that are associated to the query, and nothing unimportant (high accuracy). The meaning of the XML is Extensible Markup Language. XML and HTML are subsets of SGML (Standard Generalized Markup Language).. Many users of XML databases are not familiar with concepts such as schemas and query languages. Keyword search has been proposed as an appropriate interface for them.. Current XML keyword query answering approaches rely on heuristics that assume certain properties of the DB schema. The current approaches either suffer from low precision and low recall, or both. They either do not rank their query answers or use a very simple ranking heuristic, e.g., smallest answer first. This is undesirable because when queries do not precisely describe what the user wants, a good ranking of answers significantly improves the system's usability. In this paper, we propose a ranking approach for keyword queries that has higher precision and recall and better ranking quality than previous approaches.

Coherency ranking (CR) is a ranking approach which uses the concepts of data dependencies and mutual information. CR avoids pitfalls that lower the precision, recall, and ranking quality of previous approaches.CR finds the most probable intention for queries containing terms with multiple meanings. For instance, in the query Maude References, the term Maude can refer to a programming language or an author's last name.

In Information retrieval field TF*IDF similarity is designed to measure the relevance of the keywords and the documents in keyword search over documents. It resolves the keyword ambiguity problem by designing an XML TF*IDF on tree model, which takes the structural information of XML into account. TF (Term Frequency) is the number of times the word appears in the document and inverse document frequency (IDF) is the number of documents in which the word appears. These ambiguities occur due to the two reasons: 1) a keyword can appear both as an XML tag name and as a text value of some other nodes. 2) A keyword can appear as the text values of different types of XML nodes and carry different meanings.

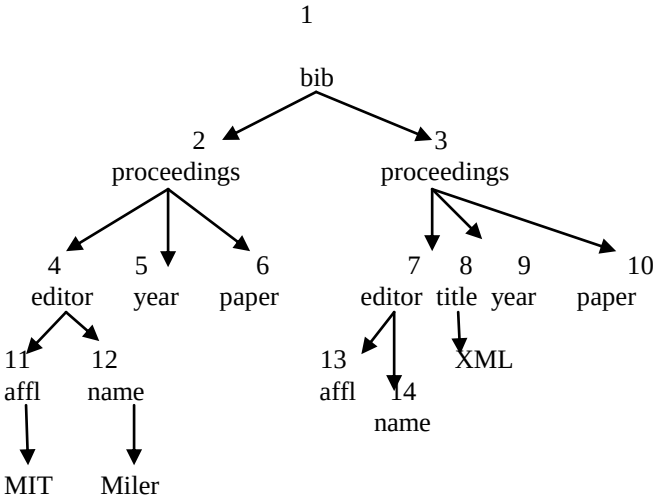

Fig 1. DBLP DATABASE

Manuscript eceived on March, 2013.

Shabnam, Computer Science and Engineering, Lovely Professional University, Jalandhar (Punjab), India.

Asst. Prof. Sumit Kumar Yadav, Information Security, IIIT, Allahabad, India.

II. REVIEW OF LITERATURE AND RELATED WORK DONE PREVIOUSLY

G. Salton and M. J. McGill Introduction to Modern Information Retrieval. [2003]. In this paper, he build a preliminary framework for object-level keyword search over XML data. In particular, he model XML data as the interconnected object-trees, based on which it was proposed two main matching semantics, namely ISO (Interested Single Object) and IRO (Interested Related Object), to capture different user search concerns.

S. Selvaganesan, Su-Cheng Haw and Lay-Ki Soon Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia [2006] found that Research on keyword search in XML database is on the increase, owing to its convenient and extensive use in information retrieval (IR) from XML data. To efficiently use the frequency information, He proposed a new formula based on mutual information between selected tags with respect to XML query keywords, and thereby lessen the uncertainty in finding an exact T typed node. Also, we propose an entropy formula to find the exact data value through the selected T-typed node.

Arash Termehchy Department of Computer Science, University Of Illinois, Urbana [2007] found that keyword search techniques that take advantage of XML structure make it very easy for ordinary users to query XML databases, but existing approaches to processing these queries rely on intuitively appealing heuristics that are ultimately ad hoc. His empirical evaluation with 65 user-supplied queries over two real-world XML data sets shows that CR has better precision and recall and provides better ranking than all previous approaches.

B. Chen and T. Ling. Exploiting id references for effective keyword search in xml documents. In DASFAA, 2008. he model XML document as a set of interconnected object-trees, where each object contains a sub tree to represent a concept in the real world. Based on this model, it was proposed object-level matching semantics called Interested Single Object (ISO) and Interested Related Object (IRO) to capture single object and multiple objects as user's search targets respectively, and design a novel relevance oriented ranking framework for the matching results.

Extensive research efforts have been conducted in XML keyword search to find the smallest sub-structures in XML data that contains all query keywords in either the tree data model or the directed graph model. An object tree t in D is a sub tree of the XML document, where its root node r is a representative node to denote a real world object o, and each attribute of o is represented as a child node of r. The data model for XML is very simple or very abstract. XML provides a baseline on which more complex models can be built.

In tree data model, LCA (lowest common ancestor) semantics is first proposed to find XML nodes, each of which contains all the keywords which are in the query within its sub tree. Consequently, SLCA (smallest LCA) is proposed to find the smallest LCAs that do not contain other LCAs in their sub trees. Each SLCA result of a keyword query contains all query keywords but has no sub tree which also contains all the keywords. SLCA-based approaches only take the tree structure of XML data into consideration, without taking into account the semantics of the query and XML data. SLCA may introduce answers that are either irrelevant to user search intention, or answers that may not be meaningful or informative enough. E.g. when a query “John Smith” that intends to find John Smith’s publications on DBLP is issued, SLCA returns only the author elements that contain both keywords. Besides, SLCA also returns publications written by two authors where “John” is a term in 1st author’s name and “Smith” is a term in 2nd author, and publications with title containing both keywords. 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. Ideally, a practical solution should satisfy two requirements: 1) it can return the meaningful results, meaning that the result sub tree describes the information at object-level; and 2) the result is relevant to the query, meaning that it captures users' search concerns. SLCA cannot remove the ambiguity which occurs due to the reasons: 1) A keyword can appear both as an XML tag name and as a text value of some other nodes. 2) A keyword can appear as the text values of different types of XML nodes and carry different meanings. The solution to handle these ambiguities is proposed in for relevance oriented ranking.

Besides LCA and SLCA, Grouped Distance Minimum Connecting Trees (GDMCT) and Lowest GDMCT as variations of LCA and SLCA for XML keyword search is also proposed. The main difference between GDMCT and LCA is that GDMCT identifies not only the LCA nodes, but also the paths from LCA nodes to their descendants that directly contain query keywords. Similarly, Lowest GDMCT identifies not only the SLCA nodes, but also the paths from SLCA nodes to descendants containing query keywords. GDMCT is useful to show how query keywords are connected to the LCA (or SLCA) nodes in result display, which is classified as path return in contrast to sub tree return in LCA and SLCA.

XRANK presents a ranking method to rank sub trees rooted at LCAs. XRANK extends the well-known Google's Page Rank to assign each node n in the whole XML tree a pre-computed ranking score, which is computed based on the connectivity of node n in the way that n node is given a high ranking score if that node n is connected to more nodes in the XML tree by parent-child edges. The pre-computed ranking scores are independent of queries. For each LCA result with descendant’s n1……n2 to contain query keywords, XRANK computes its rank as an aggregation of the pre-computed ranking scores of each node n decayed by the depth distance between ni and the LCA result. XSearch proposes a variation of LCA to find meaningfully related nodes as search results, called interconnection semantics. According to interconnection semantics, two nodes are considered to be semantically related if and only if there are no two distinct nodes with the same tag name on the paths from the LCA of the two nodes to the two nodes (excluding the two nodes themselves). XSearch 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.

XSeek addresses the search intention of keyword queries to find meaningful return information based on the concept of object classes (which they call entities) and the pattern of query matching.

It proposes heuristics to infer the set of object classes in an XML document and also heuristics to infer the search intentions of keyword queries based on keyword match patterns. Its main idea is if an SLCA result is an object or a part of an object, we should consider the whole object sub tree or some attribute of the object specified in the query that is not the SLCA for result display. XML keyword proximity search techniques based on the tree model are generally efficient. However, they cannot capture important information in ID references which are indications of node relevance in XML and they may return over-whelming information. Also, the ranking method proposed in XRANK only computes ranks among LCAs, thus it is not adequate when a single LCA is overwhelmingly large. GDMCT identifies how query keywords are connected in each LCA or SLCA result, which is useful in result display to enable the searcher to understand the inclusion of each result. XSeek based on the concept of objects is able to identify meaningful result units and to avoid returning overwhelming information. However, it does not consider relationships between objects; as a result of it XSeek may miss meaningful results of query relevant object relationships that contain all keywords.

XML graph model the major matching semantics is to find a set of reduced sub tree G0 of database graph G, such that each G0 is the smallest sub graph containing all keywords. However, the cost of finding all such G0 ranked by size is intrinsically expensive due to its NP-hard nature. Bidirectional expansion is proposed to find ranked reduced sub trees, but it requires the entire visited graph in memory, and suffers inefficiency. BLINKS improves it by designing a bi-level index for result pruning, with the tradeoffs in index size and maintenance cost. BANKS uses bidirectional expansion heuristic algorithms to search as small portion of graph as possible. X Keyword uses schema information to reduce search space, but its query evaluation is based on the method of DISCOVER built on RDBMS, which cannot distinguish the containment and reference edges to further reduce search space.

III. ISSUES AND CHALLENGES WHILE PROVIDING KEYWORD SEARCH TECHNIQUE

Coherency Technique Algorithm has been proposed for DBLP (Bibliography database) and IMDB (Internet Movie database) but there have been always need for better coherency technique algorithm.

*The existing Coherency Technique doesn't remove all the problems related to DBLP (Bibliography database) and IMDB (Internet Movie Database) .

* The existing Coherency Technique algorithm for DBLP and IMDB are less accurate and more time consuming.

*The existing Coherency Technique algorithm is costlier.

IV. PROPOSED WORK

*We propose an enhanced coherency technique algorithm which removes all the problems associated with DBLP (Bibliography database) and IMDB (Internet Movie Database).

*Our enhanced coherency technique algorithm is low cost and more accurate.
*Our enhanced coherency technique algorithm assures quality of result.
*Our enhanced coherency technique algorithm is fast and thus saves time.
*Our enhanced coherency technique algorithm is distributed and is range independent for DBLP (Bibliography database) and IMDB (Internet Movie Database).

A. Equations

The total correlation of the random variables A1,…,An is:

$$I(t) = \sum_{1 \le i \le n} H(Ai) - H(A1, \ldots, An)$$

For XML DBs:

$$I(t) = \sum_{1 \le i \le n} H(pi) - H(p1, \ldots . pn)$$

B. Algorithm to compute NTCs

Input: XML data file data
Input: Maximum size EV of patterns to compute exact NTC for
Input: Maximum size MQL of patterns to compute exact or estimated NTC for
Input: Minimum frequency thresholds MIN FREQ
Output: Table CT of NTCs for data
/* Create path indices for all frequent
root-path patterns */
1 indxs = Depth_FirstScan(data, MINFREQ);
// Compute the entropy for root-path patterns
2 for all p=2 indxs do
3 p.entropy();
/* Initialize the set of prefix classes */
4 pfxSet
/* Add all root-path patterns as one prefix
class */
5 pfxSet.add(indxs);
6 for k = 2 to MQL do
7 nextPfxSet
8 last = ();
9 forall pfx =pfxSet do
10 forall p = pfx do
/* Compute all prefix classes whose prefixes are p */
11 nextPfx
12 forall q = pfx do
13 Jnt joinPattern(p,q);
14 forall r =Jnt do
15 if subTrees(r) = pfxSet then
16 continue;
17 if k < EV then
/* Join indices and get
frequency as well as NTC
*/
18 w=join Indices(p,q);
19 if w.freq < MIN_FREQ then
20 continue;
21 CT[r] =w.I
22 else
23 CT[r]=approximateI(r);
24 if k not=MQL then
25 nextPfx.add(r);
26 if k not = MQL then

```
27 nextPfxSet.add(nextPfx);
28 pfxSet =nextPfxSet
29 return CT;
```

V. CONCLUSION

We have proposed enhanced coherency ranking, an improved ranking method for XML keyword search that ranks candidate answers based on statistical measures of their cohesiveness for IMDB (Internet Movie Database) and DBLP (Bibliography Database). Coherency ranks are computed based on a single preprocessing phase that exploits the structure of the data set. For each type of subtree in the data set, the preprocessing phase computes its normalized total correlation (NTC), an enhanced statistical measure.

REFERENCES

1. A. Schmidt, M. L. Kersten, and M. Windhouwer, "Querying xml documents made easy: Nearest concept queries." in ICDE, 2001, pp. 321–329.
2. H. He, H. Wang, J. Yang, and P. S. Yu, "Blinks: ranked keyword searches on graphs," in SIGMOD Conference , 2007, pp. 305–316.
3. L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram, "XRANK:ranked keyword search over XML documents," in SIGMOD , 2003.
4. S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv, "XSEarch: A semantic search engine for XML," in Proc. of VLDB Conference , 2003, pp. 45–56.
5. S.Agrawal, S.Chaudhuri, and G.Das. DBXplorer: A system for keyword-based search over relational databases. In *ICDE*, page 5, Washington, DC, USA, 2002. *IEEE* Computer Society.
6. Y. Xu and Y. Papakonstantinou, "Efficient keyword search for smallest LCAs in XML databases," inSIGMOD , 2005, pp. 537–538.
7. A. Baid, I. Rae, J. Li, A. Doan, and J. F. Naughton. Toward scalable keyword search over relational data. *PVLDB*, 3(1):140–149, 2010.
8. A. Balmin, V. Hristidis, and Y. Papakonstantinou. ObjectRank: authority-based keyword search in databases. In *VLDB*, pages 564–575. *VLDB* Endowment, 2004.
9. G. Bhalotia, A. Hulgeri, C. Nakhe, and Chakrabarti. Keyword searching and browsing in databases using BANKS. In *ICDE*, page 431, Washington, DC, USA, 2002. *IEEE* Computer Society.
10. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.
11. B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin. Finding top-k min-cost connected trees in databases. In *ICDE*, pages 836–845, 2007.
12. V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient IR-style keyword search over relational databases. In *VLDB*, pages 850–861. VLDB Endowment, 2003.
13. V. Hristidis, H. Hwang, and Y. Papakonstantinou. Authority-based keyword search in databases. *ACM Trans. Database Syst.*, 33(1), 2008.
14. V. Hristidis and Y. Papakonstantinou. DISCOVER: keyword search in relational databases. In *VLDB*, pages 670–681. VLDB Endowment, 2002.
15. V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar. Bidirectional expansion for keyword search on graph databases. In *VLDB*, pages 505–516. VLDB Endowment, 2005.

AUTHORS PROFILE

Shabnam, has done her graduation (B.tech) from HPU in 2011 and pursuing M.Tech in Computer Science and Engineering from Lovely Professional University, Jalandhar, India.

Sumit Kumar Yadav, has done his graduation (B.Tech) from UPTU in 2008 and achieved M.S degree in Information Security from IIIT, Allahabad, India. Currently, He is working as Assistant Professor of Computer Science and Engineering in Lovely Professional University, Jalandhar. His research area includes Database, Data mining and Data warehouse. He has published many papers in International Journals and Conferences..