ISSN: 2321-8169
1922 – 1928

Ranking clustered Keyword Search On Semi- structured data

Dayananda P
Assistant Professor, Department of
Information Science and Engg, MSRIT, Bangalore-54
e-mail: dayanandap@msrit.edu

Dr. Rajashree Shettar
Professor, Department of Computer Science and Engg,
RVCE, Bangalore-59

Abstract**—** Searching based on keyword in semi structure data is an important task. In different areas most of the common document type used is XML, support of keyword search techniques can also be used on XML data. In this paper, Identification of problem is done for clustering the result of keyword performed on XML data by the user. Results of matched queries are grouped to form the clusters. When high-level overview is provided, user can attain knowledge about the search results and able to find the relevant answers which can be easily identified. Each cluster of resulted clustering will be assigned with a meaningful label so that, users will know the results. Suggestions are given to the user and a set of related keywords are generated which the user may be interested in. The System performance depends on different sizes of XML documents. Finally ranking is performed based on clustered results and generated KMP's, ranking is performed for getting more relevant answers among clustered results.

Keywords*- Search results clustering, Keyword matching patterns, XML keyword search, keyword suggestions.*

*****

I. Introduction

Keyword search is used to discover the information from flat documents. Since XML documents are used in increasingly interested areas, it is important to extend keyword searching in XML data Search engine uses the practice of giving keyword for searching and search engine searches the keyword entered by user. Searching based on keyword in semi structure data is an important task.

XML keyword search has attracted is prevalent in the research community [7], [16], [17], [18], [24], [30]. The problem of Returning clustered results in XML documents is addressed in this paper. Cluster-based interfaces are used to provide a high-level overview for search results. The cluster hierarchy is generated and organized based on the search result. So that user can look precisely in what the user interested in. Each cluster is given a label to easily identify the contents of the cluster. Cluster analysis is the process of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other clusters. clustering is the main task in data mining and statistical data analysis is used in common and used in many fields. Studies on clustering XML documents have also been conducted widely [14], [22]. Cluster analysis is done using many algorithms as it is not a specific algorithm alone, that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Some of the various clustering approaches are centroid based clustering, hierarchical clustering, density based clustering and distribution based clustering. Suggestions are given to the user and a set of related keywords are generated which the user may be interested in. the clustered results are ranked inside the clusters so that user gets more relevant answers among clustered results. The time taken is measured for various sizes of XML documents, with pre-ranking, with post-ranking, and without ranking to analyze the performance of the system.

The main source of motivation for this paper is the fact that keyword search is used frequently in web pages and web search engines. In web development, XML used in many aspects and it is used to simplify the sharing and storage .of data. Many companies are moving to XML to write their internal documentation. XML has also been employed as the base language for communication protocols. Many applications also use XML files for configuration purposes. XML has also been used as a data type in some languages like ECMAScript. The advantage of XML for documentation is that it can be used to define the common traits in magazines, advertisement stories, book, and so forth. The XML document is presented to the user as a tree like structure as it is much easier to view and understand the contents of the document when it is presented in a tree interface rather than a text interface. There are several ways to query XML documents as presented in [9]. In different areas, most of the common document type used is XML. for storage and transferring data, keyword search can be extended to them.

II. LITERATURE REVIEW

In user perspective, XML keyword searching is not fully efficient as it faces of many problems such as Keyword searching results are devastating to explore by the users. Users get many answers for XML search as the search engine provides XML fragments instead of documents Users may spend a lot of time in examining irrelevant results and some relevant results may be overlooked.

Many studies have undergone on semantics of XML keyword search to discover how to connect the resultant matches produced by the keyword search to get the useful meaning..

Many studies are based on LCA [16] [31]. XSeek [23] represents entities and it also differs from nodes representing attributes, and produces answers for a key word match based on the entities. In XReal [7], differentiation of search via node from search from node in done for XML documents, but there is no definite semantics for search for nodes. This method cannot guarantee the semantic meaningfulness of answers for a query result.

Generation of result snippets for XML is discussed in [18].
The ranked list of output will be displayed for users and the query results are presented in interface. The interfaces which are poorly designed will cause burden on the system by allowing users repeatedly refine and submit the queries which results in decreased user experience. These problems are addressed using top-k query processing technique[2], snippet generation and ranking methods[3]. In web search field, clustering the search results has become one of the effective technique[5], Web searching techniques are not suitable for clustering XML keyword search results because structure information will not be considered which is intrinsic in XML documents

As the existing XML search engines are not satisfactory due to the above mentioned reasons, this paper aims at overcoming the drawbacks of existing XML search engines. The search results are present in a clustered interface instead of a plain interface. The clustered interface have ranked results. The statistics have been calculated with ranking (pre-ranking and post ranking), and without ranking. In web search fields[5], clustering is proved to be an effective retrieval task. The cluster-based interface presents a high-level overview on the search results. Then clusters are formed based on the way the keyword match the query. Results with similar KMP (Keyword Matching Pattern) will be grouped in one cluster. Each cluster is given a label to easily identify the contents of the cluster. The cluster based ranked interface helps the users directly concentrate on the group of results (clusters) they are interested in and ignore the rest.

III. KEYWORD SEARCH

A sample of XML document is shown in Fig 1.1 , the part of bibliography database is represented . For a keyword query "Database", multiple intension of query can be thought looking at keyword based on its intension. User may need to find articles titled XML or articles related to XML documents etc.

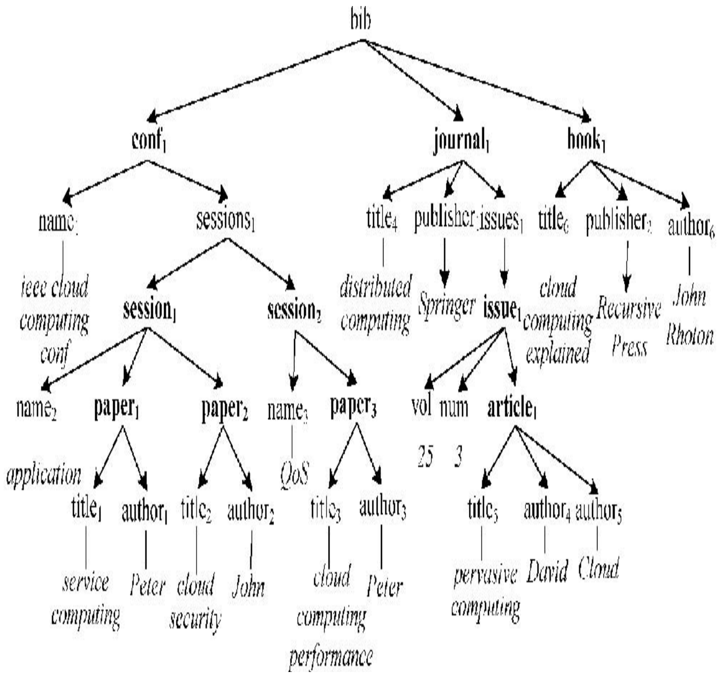

Fig 1: Sample XML document

Definition is given for keyword matching pattern (KMP) for this purpose. Keyword matching pattern is nothing but the path from the root of the XML document to the keyword match found in child node. The KMP for different occurrences of the keyword "XML" are as shown in Fig. 2. The figure shows two keyword matching patterns which indicate two occurrences of the keyword "XML".

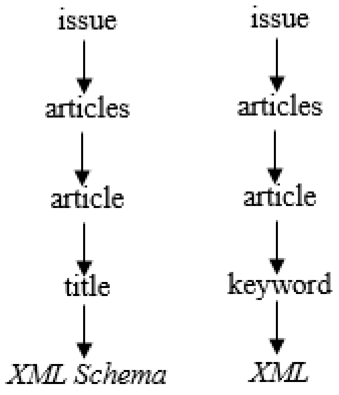

Fig 2: KMP's for the keyword "XML"

The architecture diagram is as shown Fig 3 gives the overall architecture of a system. The user enters the path to the XML document and the keyword as the input. These are taken as the input parameters to the XML parser. The XML parser then fetches that document from the user's hard drive based on the path entered by the user. The XML parser then parses the document and loads it into the memory. All the occurrence of the keyword in that document is found by keyword search module. The KMP generator then generates keyword matching patterns (KMP's) for each occurrence of the keyword in the document. The generated kmp's are then given to the ranking module, to obtain ranked kmp's which is passed into clustering module. The ranked kmp's are given as a input for clustering algorithm to generate the clusters. The suggestion module finds related items to the keyword which the user may also be interested in. Finally the summary of the transaction is grouped in a record and inserted in a log file. In the user's hard drive the log file is then stored.

For finding the occurrences of the keyword and generating the keyword matching patterns, the procedure is as follows: the XML document is read into the memory and represented as a DOM tree structure. The tree is traversed to find the occurrence of the keyword. Once a match is found the nodes are backtracked to reach the root node in order to determine the keyword matching patterns. For each occurrence of the keyword, a keyword matching pattern is generated. This KMP will be used later on for the clustering purpose.

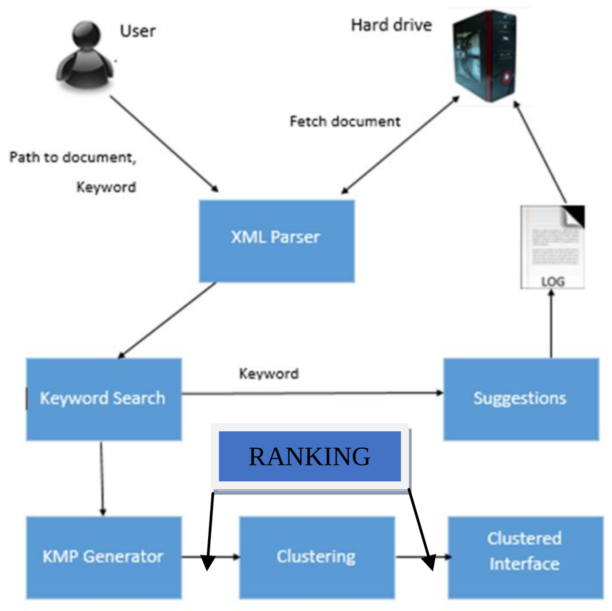

Fig 3: Architecture of the system

IV. CLUSTERING OF SEARCH RESULTS

Grouping of similar items into the same cluster forms a cluster. Here clusters can be generated by grouping similar KMP's in the same cluster. The clustering methodology should help in disambiguating keyword query and make the users to concentrate only on the needed clusters directly, hence relieving them from the overload of results. The main objective of clustering is to provide the user the item he is interested in faster and not spend time looking at irrelevant items.

Analysis of Cluster it is not one complete algorithm, The algorithms that differs in notations will forms the clusters accordingly. Finding the efficient clusters among them is important. Some of the various clustering approaches are hierarchical clustering, distribution based clustering density based clustering and centroid based clustering. Below in Fig 4 discussion of clustering methodology that can be used to cluster the searched results.

Fig 4 shows the clustering algorithm implemented in this paper. The passive approach is used for clustering algorithm for clustering XML node. Following three steps are used:

1. Answers are retrieved for input Query
2. Retrieve the KMPs for the answers
3. Clustering the result based on KMPs formed.

Below, proposal of algorithm is done that follows this approach.

```
Algorithm: Clustering
Input: A set of KMP's
Output: A set of clusters

Let C be the set of clusters
Initialize C to empty
int node = 0;
int no_of_clusters = 0;
while(true) loop
        node++;
        for each (KMP) loop
        extract 'node' from KMP
        if (cluster already present)
                insert KMP into that cluster
        else
                create new cluster
                insert KMP into that cluster
                no_of_clusters++
        end for
        if (no_of_clusters != 1) break
end while
return C
```

Fig 4: Clustering – Proposed Algorithm

V. RANKING

The ranking is performed in two ways: first pre-ranking and the second is post ranking. In the pre-ranking, the generated kmp's are passed to the ranking module, and then the ranked kmp's are passed to the cluster module. While in the post-ranking technique, the generated kmp's are passed to the cluster module and for the each cluster formation ranking module is called and ranking is performed.

Although the total processing time includes ranking time, but the ranking time is constant in both the ranking applied, i.e. any ranking algorithm can be used in the implementation. In this work, simply ranking technique is used to perform ranking among clustered results. The simple ranking technique works based on the exact match of the keyword query and the retrieved result, If it matched exactly as the searched query among the retrieved result. The retrieved results will be raked first and later results are ranked later on depending on the KMP's generated.

VI. SUGGESTIONS AND LOGGING

Suggestions are related keywords to the input keyword that are given to the user which he may also be interested in. These related keywords are logically related to the input keyword. For an example, if the input keyword is "data mining", the related keywords may be "multimedia mining" ,"web mining" and "text mining", "multimedia mining" etc.

For giving suggestions to the user, Google suggest tool is used. It takes the keyword given by the user as the input and generates an XML document and returns it. This XML document contains the related keywords, the document is parsed and the related keywords are extracted and given to the user.

Finally the summary of the whole transaction is compressed into a record and added to a log file. This log file contains details about all the actions carried out by users on the system. The contents of each record are – the path to the XML document, the keyword, time taken for keyword search, time taken for clustering and a timestamp of when the system was executed.

The log contains the timestamp, XML document, keyword, search time and clustering time in the same order. Similar to this each record will be added on to the log file every time the system is executed. This log file can be later used by system administrators or data analysts to analyze what users are searching and the time taken for different XML documents.

VII. EXPERIMENTAL RESULTS AND DISCUSSION

Java had lot of built in function and application so whole system is implemented using java concepts and fact that it is platform independent allows us to execute the system on any platform. The system was developed in a Windows operating system, however it can be executed in any platform. The Integrated Development Environment (IDE) used for development is Eclipse or STS. DOM (Document Object Model) and SAX (Simple API for XML) are the XML parsing libraries that are used.

The following datasets have been used to test and evaluate the system.

1. Bibliography: This contains information about books conferences and journals.
2. Resume: This contains information from the resume of different people like the skills and technology known etc.
3. Clinical study: This contains clinical records of patients in a hospital and the diseases they have.
4. Medicine: This contains information about the medicines and their properties.

The system asks the user to enter the path of the XML document and the keyword as the input. It then calculates the total number of occurrences of that keyword and generates keyword matching patterns and displays it to the user as shown.

A tree like structure is generated for the user which contains the xml tree like structure. This window is given to the user because it is easier to view the XML document in a graphical way instead of reading lines of text. The keyword matches are marked in blue so that the user can easily identify his search result.

In this example, three clusters are generated by the labels – books, journals and conferences. Each cluster has items relevant to the label. So the user can directly look at a cluster label and if he is not interested in that cluster, he can close that window thereby eliminating many results.

The performance of the system is compared with the active and passive clustering approaches presented in [2]. The datasets used are Mondial [31] and Sigmod Record [31]. A set of keyword queries is tested for each dataset. The list of queries is given in Table 1 and Table 2.

No.	Query
QM1	France territory
QM2	New York
QM3	Lake located
QM4	River Colorado
QM5	Singapore country
QM6	Religious Christian muslim
QM7	Province Houston dallas

Table 1: Mondial

No.	Query

QD1	PODS
QD2	DIVESH SRIVASTAVA DAN SUCIU
QD3	CONCEPT MODEL 2003
QD4	DATABASE WORKLOAD
QD5	AUTHOR PETER CHEN
QD6	XML DATABASE AUTHOR

Table 2: Dblp Record

The queries presented in Table 1 and 2 were tested on a system with the following configurations: RAM – 6 GB; processor – Intel core i5 and the results obtained are given in Table 3, 4 and 5. A comparison between the passive, pre-ranking and post-ranking approaches are shown in Table 5. The table 5 shows the comparison in term of throughput time for various approach for some queries the throughput time for our approach is good and ranking implementation gives the relevant results among the clustered results, so that user finds the appropriate answer listed at the top of clustered results. Throughput time is calculated by keyword search time and Clustering time.

Query	**Keyword Search Time (milliseconds)**	**Clustering + Ranking Time (milliseconds)**	**Total Processing Time(ms)**
QD1	496	824	1320
QD2	911	978	1889
QD3	1079	1187	2266
QD4	533	657	1190
QD5	867	1018	1885
QD6	511	682	1193
QM1	75	158	233
QM2	71	103	174
QM3	84	116	200
QM4	73	77	150
QM5	76	70	146

Table 3: Time taken for pre-ranking

Query	**Keyword Search Time (milliseconds)**	**Clustering + Ranking Time (milliseconds)**	**Total Processing Time(ms)**
QD1	511	1499	2010
QD2	1247	1746	2993
QD3	956	2178	3134
QD4	401	1399	1800
QD5	1343	1802	3145
QD6	523	1556	2079
QM1	70	626	696
QM2	77	784	861
QM3	69	673	742
QM4	76	674	750
QM5	57	679	736

Table 4: Time taken for post-ranking

Query	**Xmean-Passive approach (without ranking)**	**Proposed Pre-ranking**	**Proposed Post- ranking**
QD1	1315	1320	2010
QD2	1889	1889	2993
QD3	2266	2266	3134
QD4	1190	1190	1800
QD5	1885	1885	3145
QD6	1193	1193	2079
QM1	233	233	696
QM2	174	174	861
QM3	200	200	742
QM4	150	150	750
QM5	146	146	736

Table 5: Comparison between different approaches

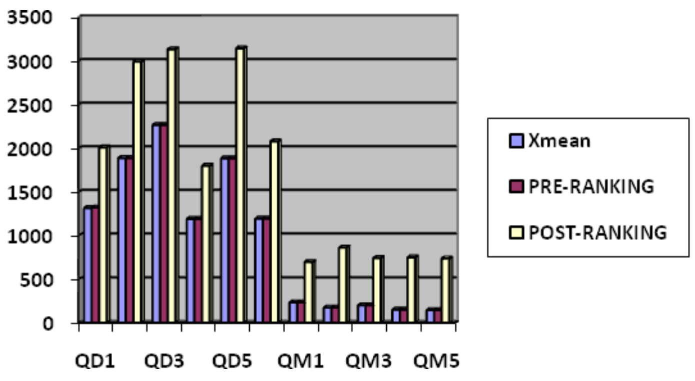

Fig 5 : Comparison of different approaches in term of throughput time

The Fig 5 shows the comparison of passive approach, Pre-raking and post-Ranking. The experimental results, shows that the throughput time is good for pre- ranking then XMean[2] and post ranking approach. Raking among clustered results gives the most relevant answers within the clusters.

Fig 6 and Fig 7 shows the comparisons of precision and recalls of XMean and proposed approach. perfect recalls on DBLP is given by both our proposed approach and XMean For certain queries like QM1 the semantic fits good, but queries like QD2 the semantics does not fit at all. In this work , it infers the nodes according to the relationship and conceptual role, but not respect to paths or tags . The duplication tag existence can be easily predicted to exists or not accordingly.

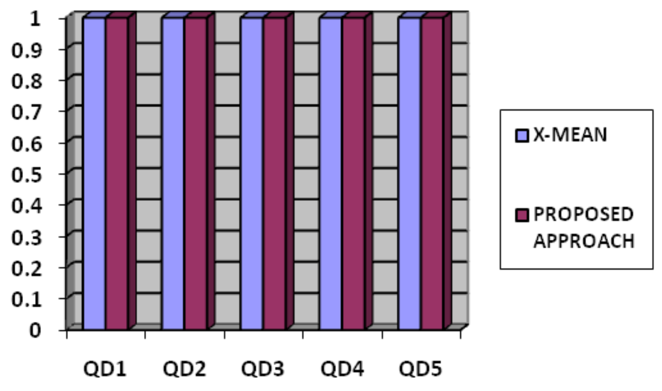

Fig 6: Recall on DBLP

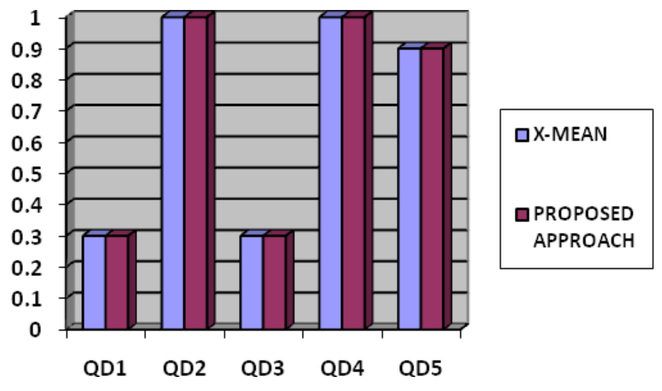

Fig 7: precision on DBLP

VIII. CONCLUSION

In our work, Possibility of all problem occurrence is done for clustering performed on the xml document for the keyword search. Keyword search is performed on the XML document and generation of keyword matching patterns (KMP's) is done first. Then, the proposal of a new algorithm is done to cluster search results based on the similarity between KMP's. The main objective of clustering is to provide the user the item user is interested in faster and not to spend time looking at the irrelevant items. Each cluster is given a unique label to easily identify the contents of that cluster. Suggestions are given to the user on related keywords to the input keyword which he may also be interested in. A log file is generated with the summary of the transactions. Finally the performance of the system is measured with respect to throughput time taken.

A comparison of time taken for different sizes of XML documents are discussed. The search time is completely depends on the size of the document. The size of the document is increases, the time taken to search the occurrences of the keyword will also get increases. The clustering time however is not dependent on the size of the document. The clustering time depends on the number of KMPs generated and finally pre-ranking and post-ranking implementation gives the relevancy among the clustered results. The Experimental results show the throughput time is good for mondial and dblp database with respect to proposed pre-Ranking with clustering than XMean approach.

IX. REFERENCES

[1] G. Costa, R.Ortale, "Structure-oriented clustering of XML documents: A transactional approach", 6th IEEE International Conference on Intelligent systems, September 2012.

[2] Xiping Liu, Changxuan Wan, and Lei Chen, "Returning Clustered Results for Keyword Search on XML Documents", IEEE Transactions on Knowledge and data engineering, VOL. 23, NO. 12, December 2011.

[3] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, "On Finding Lowest Common Ancestors in Trees," Proc. Fifth Ann. ACM Symp. Theory of Computing, 1973.

[4] S. Amer-Yahia, L.V.S. Lakshmanan, and S. Pandit, "FleXPath: Flexible Structure and Full-Text Querying for XML," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2004.

[5] Z. Bao, T.W. Ling, B. Chen, and J. Lu, "Effective XML Keyword Search with Relevance Oriented Ranking," Proc. 25th Int'l Conf. Data Eng., 2009.

[6] C. Carpineto, S. Osinski, G. Romano, and D. Weiss, "A Survey of Web Clustering Engines," ACM Computing Surveys, vol. 41, no. 3 , pp. 1-38, 2009.

[7] L. Chen and Y. Papakonstantinou, "Supporting Top-K Keyword Search in XML Databases," Proc. 26th Int'l Conf. Data Eng., 2010.

[8] T. Chen, J. Lu, and T.W. Ling, "On Boosting Holism in XML Twig Pattern Matching Using Structural Indexing Techniques," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2005.

[9] J. Clark and S. DeRose, "XML Path Language (XPath) Version 1.0," W3C Recommendation, 1999.

[10] 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), 2005.

[11] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv, "XSEarch: A Semantic Search Engine for XML," Proc. 29th Int'l Conf. Very Large Data Bases, 2003.

[12] S. Flesca, G. Manco, E. Masciari, L. Pontieri, and A. Pugliese, "Fast Detection of XML Structural Similarity," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 2, pp. 160-175, Feb. 2005.

[13] R. Goldman and J. Widom, "DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases," Proc. 23rd Int'l Conf. Very Large Data Bases, 1997.

[14] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram, "XRANK: Ranked Keyword Search over XML

Documents," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2003.

[15] 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.

[16] Y. Huang, Z. Liu, and Y. Chen, "Query Biased Snippet Generation in XML Search," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2008.

[17] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram, "A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results," Proc. 13th Int'l Conf. World Wide Web, 2004.

[18] G. Li, J. Feng, J. Wang, and L. Zhou, "Effective Keyword Search for Valuable LCAs over XML Documents," Proc. 16th ACM Conf. Information and Knowledge Management, 2007.

[19] Y. Li, C. Yu, and H.V. Jagadish, "Schema-Free XQuery," Proc. 30 th Int'l Conf. Very Large Data Bases, 2004.

[20] W. Lian, D.W.-L. Cheung, N. Mamoulis, and S.-M. Yiu, "An Efficient and Scalable Algorithm for Clustering XML Documents by Structure," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 1, pp. 82-96, Jan. 2004.

[21] Z. Liu and Y. Chen, "Identifying Meaningful Return Information for XML Keyword Search," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2007.

[22] Z. Liu and Y. Chen, "Reasoning and Identifying Relevant Matches for XML Keyword Search," Proc. VLDB Endowment, vol. 1, no. 1 , pp. 921-932, 2008.

[23] Z. Liu and Y. Chen, "Return Specification Inference and Result Clustering for Keyword Search on XML," ACM Trans. Database Systems, vol. 35, no. 2, pp. 1-47, 2010.

[24] Z. Liu, P. Sun, and Y. Chen, "Structured Search Result Differentiation," Proc. VLDB Endowment, vol. 2, no. 1, pp. 313324, 2009.

[25] M. Necasky, "Conceptual Modeling for XML: A Survey," Technical Report No. 2006-3, Dept. of Software Eng., Faculty of Math. and Physics, Charles Univ., 2006, http://www.necasky. net/papers/tr2006.pdf.

[26] A. Schmidt, M. Kersten, and M. Windhouwer, "Querying XML Documents Made Easy: Nearest Concept Queries," Proc. 17th Int'l Conf. Data Eng., 2001.

[27] C. Sun, C.-Y. Chan, and A.K. Goenka, "Multiway SLCA-Based Keyword Search in XML Data," Proc. 16th Int'l Conf. World Wide Web, 2007.

[28] Y. Xu and Y. Papakonstantinou, "Efficient Keyword Search for Smallest LCAs in XML Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2005.

[29] Y. Xu and Y. Papakonstantinou, "Efficient LCA Based Keyword Search in XML Data," Proc. 11th Int'l Conf. Extending Database Technology: Advances in Database Technology (EDBT), 2008.

[30] "DBLP Bibliography," www.informatik.uni-trier.de/~ley/db/, 2011.

[31] http://www.cs.washington.edu/research/xmldatasets/.