ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(29), DOI: 10.17485/ijst/2015/v8i29/61056, November 2015* ISSN (Online) : 0974-5645

Comparison of XReal and XDMA for Keyword Search in XML Databases: A Case Study

S. Selvaganesan* **and G. V. Shrichandran**

Department of Information Technology, J. J. College of Engineering and Technology, Tiruchirappalli - 620009, Tamil Nadu, India; sselvaganesan@gmail.com, gvshrichandran@gmail.com

Abstract

More research works have been carried out in the area of keyword search on XML databases. This is mainly because of the fact that users do not require knowledge of database schema and query language to search XML databases based on keyword queries. In the recent past, XReal and XDMA are the most prominent keyword search approaches for XML databases. Both XReal and XDMA make use of the numerical facts and hierarchical structure of XML databases. Never the less, these approaches are not reliant on schema information of XML data. In XReal, term frequency and document frequency have been mainly utilized. However, XDMA employs dual indices and mutual summation with the use of the frequency of keyword matching tags and data. In this paper, we discuss briefly the very important aspects of these approaches. Also, we compare these two keyword search approaches based on their methodologies and experimental results. It is found that both XReal and XDMA are generally effective XML keyword search approaches. Moreover, XDMA is comparatively slower than XReal for larger XML datasets. Notably, XDMA has identified a new keyword ambiguity and also addressed all ambiguities.

Keywords: Keyword Search Approach, XDMA, XML Databases, XReal

1. Introduction

The most important objective of XML keyword search is to determine precisely the users' intention while searching in the existence of keyword ambiguities[1,2]. Moreover, querying XML databases is entirely different from querying text databases. As a result, a number of challenges exist in keyword search on XML databases. The primary challenges to be resolved are determination of intention of users while searching, addressing ambiguity problems and ranking (grading) the search results effectively. Of course, a lot of research works on XML keyword search have been carried out. However, some aspects of these challenges are yet to be resolved. In the recent past, XReal[1,2] and XDMA[3–6] are the notable Information Retrieval style keyword search approaches for XML database, to resolve these challenges. These approaches exploit the numerical facts (statistics) of XML database. Moreover, hierarchical structure of XML database is taken into consideration in these approaches. However, they do not rely on schema information of XML data. In this paper, we study and compare these two keyword search approaches XReal and XDMA for XML databases.

This paper is arranged as follows. We briefly discuss the keyword search approach XReal in Section 2. Section 3 discusses the keyword search approach XDMA. In Section 4, a comparative study of XReal and XDMA based on the experimental results has been made. Finally, in Section 5, we conclude the paper.

2. XReal

The keyword search approach XReal addresses the challenges mentioned in Section 1. In this section, the important features of XReal are briefly discussed.

In XReal, Bao *et al.*[2] have identified and highlighted three keyword ambiguities in XML keyword queries. Figure 1 illustrates the portion of dblp[5] XML data tree. The three keyword ambiguities stated in [2] can be described

* *Author for correspondence*

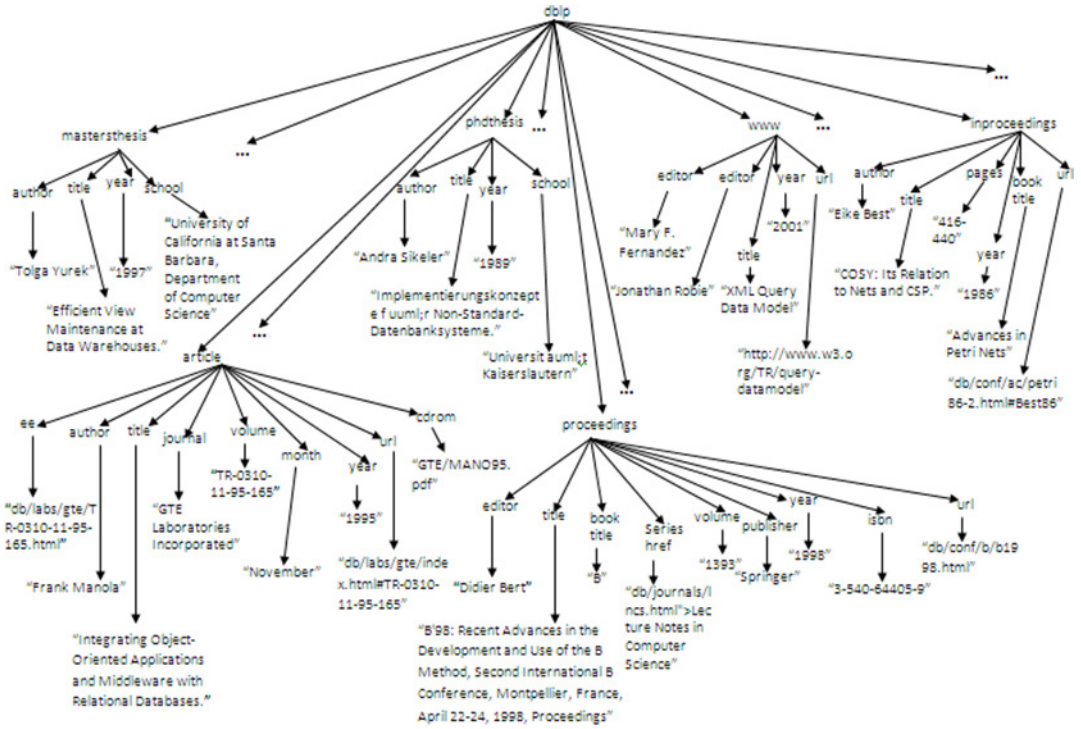


Figure 1. Part of dblp XML Data Tree.

using the following example keywords issued on dblp XML dataset in Figure 1.

A keyword "*journal*" exists as a tag in *dblp, proceedings* and *dblp, article* and, also as a data (text) value in *dblp, inproceedings*.

A keyword "*September*" exists as a data (text) value of the tag month in *dblp, phdthesis* as well as a data (text) value of the tag title in *dblp, proceedings*, and it does not have same meaning.

A keyword "*year*" exists, in different circumstance, as the year (tag) of phdthesis and proceedings, and it does not have same meaning.

This keyword search approach is designed and developed for data centric XML datasets. In this approach, Bao *et al.*[1,2] used the numerical facts (statistics) of XML database to address the problems stated in Section 2 when keyword ambiguities are present.

In this approach, two indices, i.e., keyword Inverted List (IL) and Frequency Table, are constructed. Using these two indices, the processing of query is accelerated. The keyword inverted list takes out a list of data nodes having values which hold the query keyword, in document order. There exists an index built on the top of each inverted list. Each IL holds the input query keyword in a tuple form <*DeweyID, prefixPath*, $f_{a,k}$, *Wa*>.

The frequency f_k^T only is stored in the frequency table. f_k^T is the number of nodes of type T whose subtrees contain the query keyword k in XML data, for each combination of keyword k and node type T in XML document.

In XReal, Bao *et al*[2]. make use of statistics to deduce search intention of user and grade the query results. In their work, they defined the following two frequencies.

- XML Term Frequency $f_{a,k}$ ("XML TF"): The number of occurrences of a keyword k in a given data node a in XML data.
- XML Document Frequency f_k^T ("XML DF"): The number of T-typed nodes that contain keyword k in their subtrees in XML data.

Based on these two frequencies, formulae were designed to determine "search for" and "search via" nodes of a given query. Moreover, Bao *et al.*[2] presented a ranking scheme using XML TF*IDF similarity.

3. XDMA

An effective keyword search approach for XML databases, XDMA[3–6], has been designed and developed using the concepts of dual indexing and mutual summation, distinctively for data centric XML datasets. The primary objective of XDMA is to resolve the problems mentioned

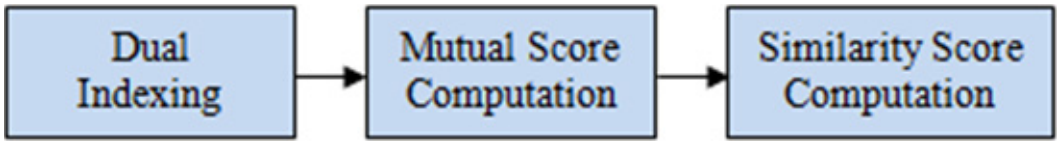


Figure 2. Overview of XDMA.

in Section 1 and also to address the limitations of the keyword search approach, XReal.

An overview of XDMA is illustrated in Figure 2 and the main features of XDMA are dual indexing, mutual score computation and similarity score computation.

3.1 Dual Indexing

XDMA builds dual indices, i.e., tag information table and data node information table for structural nodes and data nodes of XML databases respectively[3–7]. The information about each structural node and data node in XML database are stored in the tag information table and data node information table respectively. The data node information table depends on the tag information table with regard to the tag name. Figure 3 shows the portion of tag information table index and data node information table index of dblp XML dataset and also illustrates the dependence between these two tables. Obviously, the dual indexing handles each structural node and data node separately in XML database so that the query processing is simplified.

3.2 Mutual Score

In case of occurrence of larger keyword matching in XML database for a query, filtering out the desired *T*-typed node is very much complicated. Mainly for this reason, an arithmetic formula has been devised in XDMA. The mutual dependence between the two indices has been utilized in such a way that mutual score integrating the mutual summation among tag and data keyword is applied to XML keyword search. Using the mutual score among tag and data indices, the desired *T*-typed node is selected. In XDMA, mutual score between selected tags and query keywords has been defined using the logarithmic and probability functions in order to determine the precise *T*-typed node.

3.3 Similarity Score

XDMA determines similarity between the XML leaf nodes and the given query to find out the exact data, through the selected *T*-typed node. For the purpose, the approach defines a formula for similarity measure among the leaf nodes and query. XDMA provides the ranking (grading) based on the sum of mutual score and similarity score.

4. Comparison of XReal and XDMA

4.1 Based on Methodology

In XReal, it is found that the size of Dewey ID[2] is obviously bigger. Also the length of Dewey ID of an XML element increases as the depth of the XML element increases in the

Tag information table index (Portion)

Sl.No	Tag	Frequency	Path
11	article	90	dblp,article
12	editor	3	dblp,article
13	journal	90	dblp,article
14	volume	90	dblp,article
15	year	90	dblp,article
16	cdrom	87	dblp,article

Data Node Information table index (Portion)

Sl.No	Data	Tagname	Frequency
100	TM-0149-06-89-165	volume	2
101	1989	year	8
102	db/labs/gte/index.html# TM-0149-06-89-165	url	2
103	GTE/MANO89a.pdf	cdrom	2
104	db/labs/gte/ TR-0310-11-95-165.html	ee	2

Figure 3. Dual indices and dependence between them.

XML data tree, in that way making the process complex. Because of each IL having index on its top, storage requirements will be more. Information about leaf tag containing data values in XML databases are not provided by keyword IL. XReal does not take into consideration the frequency of each data (text) value contained in a leaf node in XML databases.

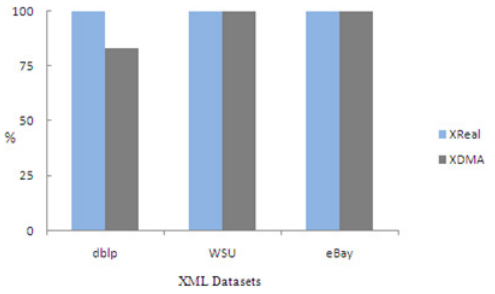


Figure 4. Percentage of search effectiveness for XReal and XDMA.

The frequency Table 2 stores only f_k^T. Only *T-typed* nodes in XML databases are considered. Also, the frequency table does not handle each tag and data node separately.

In addition to the three ambiguities stated in [2], XReal does not consider the following ambiguity. "A keyword can exist as the name of a tag for node types having different data (text) values and vice versa".

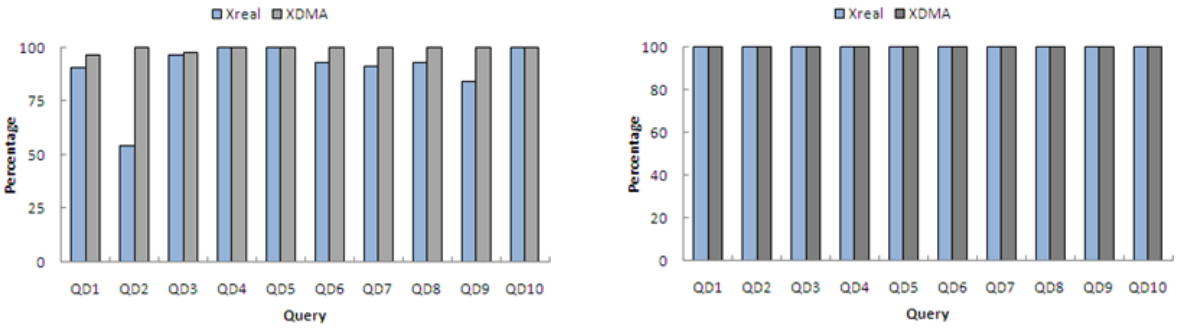
The existing keyword search approaches for XML databases do not identify whether the query keyword is a tag or data.

It is noted that the keyword search approach XDMA[3–6], addresses the limitations of XReal.

4.2 Based on Experimental Results

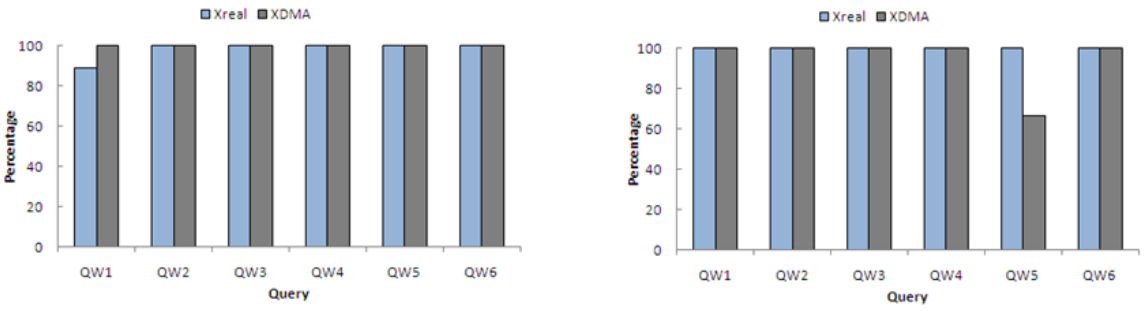
Experiments have been conducted on three XML datasets[8], i.e., dblp, WSU and eBay using SLCA[9], XSeek[10] and XReal[1,2]. In these experiments, the keyword queries issued on dblp are "QD1: Java, book, QD2: author, Chen, Lei, QD3: Jim, Gray, article, QD4: xml, twig, QD5: Ling, tok, wang, twig, QD6: vldb, 2000". Also, queries on WSU are "QW1; 230, QW2: CAC, 101, QW3: ECON, QW4: Biology, QW5: place, TODD, QW6: days, TU, TH" and queries on eBay are "QE1: 2, days, QE2: cpu, 933, QE3: Hard, drive, CA".

XDMA deduces the desired search for node in most of



(a) Precision (b) Recall

Figure 5. Comparison of XReal and XDMA using dblp dataset.



(a) Precision (b) Recall

Figure 6. Comparison of XReal and XDMA using WSU dataset.

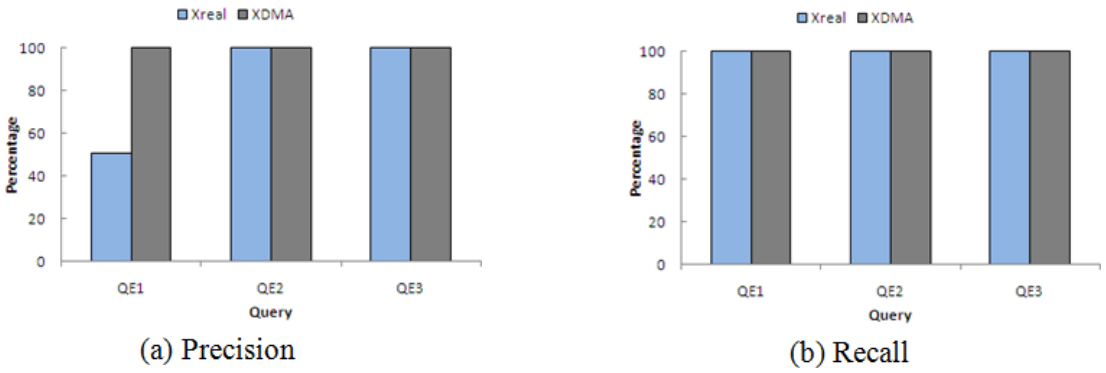


Figure 7. Comparison of XReal and XDMA using eBay dataset.

the test queries on dblp, WSU and eBay datasets. Figure 4 illustrates the percentage of search effectiveness for both XReal and XDMA.

As illustrated in Figure 5, 6, 7 and 8, the experimental evaluation using IR metrics, namely, precision, recall and F-measure shows the effectiveness of XDMA as comparable to XReal. Like XReal, XDMA has achieved better effectiveness.

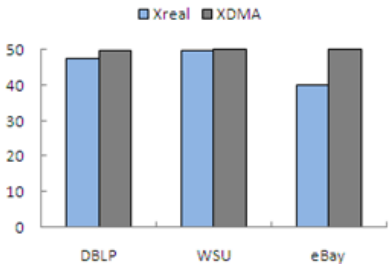


Figure 8. Comparison of XReal and XDMA based on F-measure.

Both XReal and XDMA are scalable as the response time of XReal. As shown in Figure 9, XDMA increases with increase in number of keywords in query and size of result. XDMA addresses all the keyword ambiguities including the newly identified ambiguity.

In XDMA, in case of tag keywords in query, keyword matching and information retrieval occur only in the tag information table. On the other hand, in case of data keyword in query, initial search occurs in the tag information table and then keyword matching data value is searched in the data node information table. For the corresponding leaf tag of each keyword matching data value, the relevant information is taken from the tag information table. In case of query having both tag and data keywords or data keyword(s) only, the combined frequency for mutual score is determined by selecting the frequency of tag of leaf node containing keyword matching data value from the tag information table with regard to leaf tag having keyword matching data from the data node information table. Therefore, comparing with XReal, query processing in XDMA has a considerable overhead. Because of the fact that keyword matching and information retrieval takes place in larger indices of larger

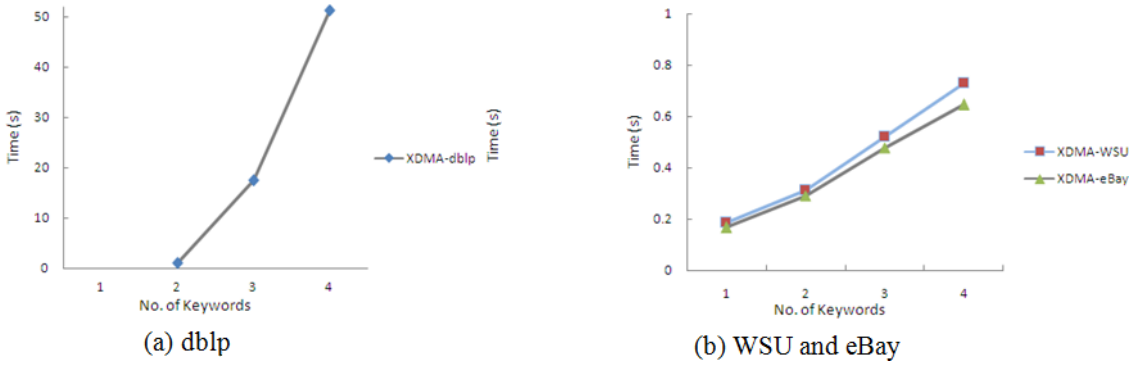


Figure 9. Response time of XDMA with respect to number of keywords.

dblp XML dataset, XDMA is notably slower on dblp than XReal. For larger XML datasets, XDMA is significantly less efficient than XReal.

5. Conclusion

In this paper, we have discussed various aspects of XReal and XDMA for keyword search on XML databases. These approaches are well suited for data centric XML datasets. In XReal, term frequency and document frequency have been primarily used whereas dual indices and mutual summation have been employed using the frequency of keyword matching tags and data values in XDMA. Comparison of XReal and XDMA has been made based on their methodologies and experimental results. Like XReal, XDMA is an effective keyword search. In case of larger datasets, XDMA is slower than XReal. Moreover, XDMA identifies a new keyword ambiguity and addresses all ambiguities.

6. References

1. Bao Z, Ling TW, Chen B, Lu J. Effective XML keyword search with relevance oriented ranking. Proceedings of the IEEE International Conference on Data Engineering; 2009. p. 517–28.
2. Bao Z, Lu J, Ling TW, Chen B. Towards an effective XML keyword search. IEEE Transactions on Knowledge and Data Engineering. 2010; 22(8):1077–92.
3. Selvaganesan S, Haw SC, Soon LK. Effective XML keyword search using dual indexing technique. Inform Tech J. 2014; 13(4):643–51.
4. Selvaganesan S, Haw SC, Soon LK. XDMA: A dual indexing and mutual summation based keyword search algorithm for XML databases. Int J Software Eng Knowl Eng. 2014; 24(4):591–616.
5. Selvaganesan S. Dual indexing and mutual summation based keyword search method for XML databases [PhD thesis]. Cyberjaya, Malaysia: Multimedia University: 2014.
6. Selvaganesan S, Haw SC, Soon LK. Effective keyword search approach XDMA for XML databases. Mitteilungen K losterneuburg Journal. 2014; 64(10):43–53.
7. Selvaganesan S, Haw SC, Soon LK. Towards developing an efficient approach to keyword search for XML documents. Proceedings of International Conference on System Engineering and Modeling; 2012. p. 78–83.
8. Miklau G. XML data repository. 2002. Available from: http://www.cs.washington.edu/research/xmldatasets/www/repository.html
9. Xu Y, Papakonstantinou Y. Efficient keyword search for smallest LC As in XML databases. Proc ACM SIGMOD International Conference on Management of Data. 2005. p. 537–8.
10. Liu Z, Chen Y. Identifying meaningful return information for XML keyword search. Proc ACM SIGMOD International Conference on Management of Data. 2007. p. 329–40.