ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 3, Special Issue, March-April 2013, An ISO 9001: 2008 Certified Journal.
E-NSPIRE, A National Level Conference held at Pravara Rural Engineering College, Loni, Maharastra, INDIA.

Retrieval of Structure and Content Information from XML Document

P. B. Vikhe, B. L. Gunjal

Abstract— **Day by day the documents over internet are increasing rapidly. Each day huge amount of data is reply as answer for simple query. Therefore to get expected information from documents became very hard task. In this paper, we describe branch organization rule which will provide approximate answer for queries. For mining all of this information we have used XML (Extensible Markup Language). XML is a portable language used by most of web technology.**

Index Terms— **Data Mining, Query Answering, XML.**

I. INTRODUCTION

Most of the time data over internet is not found in structured and it’s also not flexible to store and parse it using databases. XML is used to represent huge amount of data without any absolute schema and structure. To retrieve information from XML document two techniques are used keyword search and query retrieval. Keyword search is used when we have to match exact word and it does not support for exact answer. Query retrieval is used whenever document is following certain schema but its availability of documents with schema is partially fulfilled. So when we search query over document without schema it fails [1]. Unstructured document causes excess of information to be included in answer which is not required and formulation of query becomes hard task. If at all your formulation of query goes wrong the resultant will also fail you to give exact expected answer. Mining of XML documents differs from structured data mining and text mining. The structure of an XML document is indicated by element tags and their nesting. It allows the representation of semi-structured and hierarchal data containing the values of individual items and the relationships between data items. Mining of contents along with structure provides new means into the process of knowledge discovery [2].

II. RELATED WORK

The problem of XML context was proposed by D. Braga and G. Dobbie. Later on G. Dobbie proposed XQuery to extract association rules from simple XML documents. By using Apriori algorithm they propose a set of functions written only in XQuery. G. Dobbie and J.W.W. Wan shown that their approach performs well on simple XML documents but it is very difficult to apply on complex XML documents with an irregular structure. This drawback are overcome by D. Braga and A. Campi, where they proposed to enrich XQuery with data mining and knowledge discovery capabilities, by introducing XMINE RULE.

It is a specific operator for mining association rules for native XML documents. The main goal is to take a more general approach for the problem of extracting association rules from XML documents, i.e. to mine all frequent rules, without having any a-priori knowledge of the XML dataset. The same idea was presented by J. Paik and H.Y. Youn introducing HoPS, for extracting association rules in a set of XML documents called as XML association rules. The idea of using association rules as summarized representations of XML documents was also introduced by E. Baralis, P.Garza, E. Quintarelli and L. Tanca where the XML summary is based on the extraction of association rules both on the instance pattern and on schema patterns from given datasets. Our idea is to mine starting from frequent subtrees of the tree-based representation of a document.

III. OBJECTIVES

Our main objective in this paper is to create process which allows us to generate BOR over unstructured XML documents which is used further for intentional query retrieval. Our secondary objective is that the process should allow directly work with XML document without considering changes in XML file in to other formats. BOR also stores information in XML format so that it can be easily used [3]. In this paper we address the problem of query over unstructured documents by mining patterns and knowledge in document i.e. frequent data. By using branch organization rules we can show an intentional answer. XML’s XQuery can be used to fire a query on XML document which is having certain specific schema. In this paper we propose branch organization rules for representation of frequent answer in XML document [5]. The intentional information generated by branch organization rules can provide answer support in following cases: i) To understand frequent patterns from data set, ii) Most of XML don’t follow any DTD or schema. So user can’t specify XQuery over XML document. Therefore BOR can be used for formulation of query. iii) It constructs indexes and patterns with related constraints which can be used for query optimization. iv) Sensitive details can be hidden from user as concerned with privacy. v) BOR is useable not only when user requires quick answer but also when original document is lost by using extracted information. So BOR provides useful information for getting abstract level of information from unstructured XML document [10].

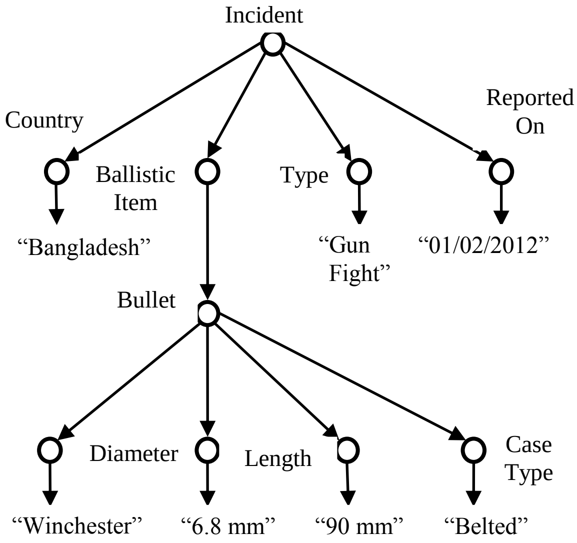
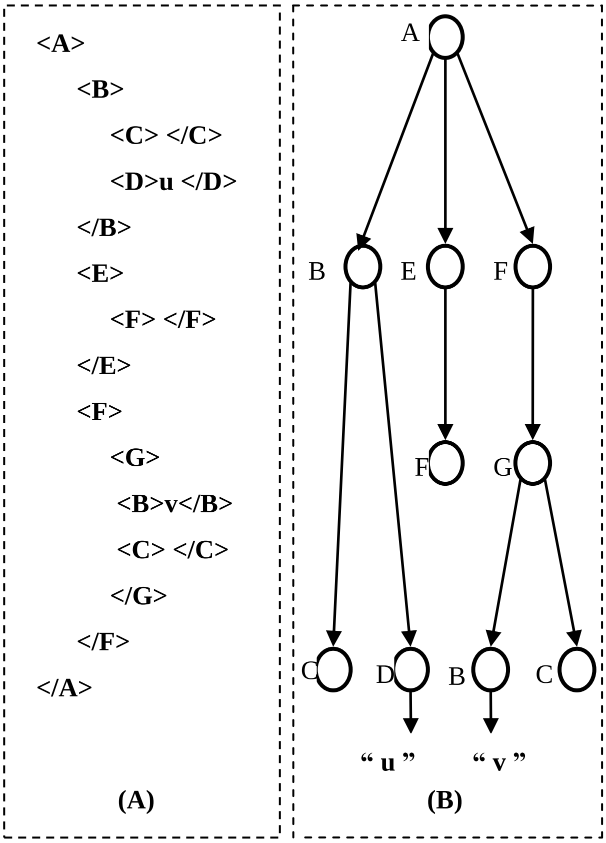


Fig 1. Sample XML File: "Incidents.XML"

For testing our schema we can use Odyssey EU as our training data set. Odyssey EU project contains information of crimes in Europe. By querying in such large information we are able to test our system effectiveness. And if at all original data is lost we are able to work with BOR to retrieve information.

Branch Organization Rule:-

Let's take a sample code for a XML file.



```
<A>
    <B>
        <C> </C>
        <D>u </D>
    </B>
    <E>
        <F> </F>
    </E>
    <F>
        <G>
         <B>v</B>
         <C> </C>
        </G>
    </F>
</A>
```

Fig 2. (A) An example of XML file. (B) Its tree based representation

The Quality of association rules are checked by using two factors those are support and confidence. Support refers to frequency of set in the data set whereas Confidence refers to probability of next subset at a given node.

To understand support and confidence let's take one example. Consider two sets of data items A and B in the form A ➔B such that A ∩ B = Ø. Support can be calculated as number of occurrence of the set A ∪ B in the data set, while confidence can be calculated as finding B when found A [13].

IV. BOR SUNDER OUT

BOR mining is divided into two parts. In first part mining of subtree is done i.e. finding support. And in second part interesting rules are computed i.e. finding confidence. Algorithm 1 can be used for general frequent subtree mining algorithm in order to compute interesting rules.

Algorithm 1:-

```
Deriving_Interested_Rules (X, Minsupport, Minconfidence)
// Frequent Subtrees
Fsearch = SearchFrequentSubtrees (X, Minsupport)
Set_Rule = Ø
For all s Є Fsearch do
// Computing Rules from s
Temporary_Set = Computed_Rules (s, Minconfidence)
// For All Rules
Set_Rule = Set_Rule ∪ Temporary_Set
End For
Return Set_Rule
```

Input for algorithm 1 is a XML document X with threshold for support i.e. minsupport and threshold for confidence i.e. minconfidence.

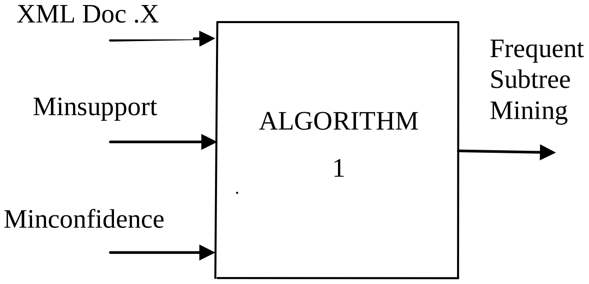


Fig 3. Frequent Sub tree Mining

Function 1:-

```
(
List_Black = Ø
New_Rule_Set = Ø
For All St, Subtree of s do
If St is not a subtree in List_Black Then
Confidence = Support (s) / Support (St)
If Confidence >= Minconfidence Then
   New_Rule_Set = ( St, s, Confidence, Support (s))
   Set_Rule = Set_Rule ∪ (New_Rule_Set)
Else
   List_Black = List_Black ∪ St
End if
End if
End for
Return Set_Rule
```

The given algorithm will find all frequent sub tree and then each sub tree will be forward to a function 1 that compute rules. Amount of rules generated will be directly proportional to number of nodes in the sub tree. If there are n nodes in given sub tree it could generate $2^n - 2$ rules [16]. It allows optimization of XML document. Hence for a document that is frequently updated, we need to apply this algorithm regularly. But those document which uploaded rarely for them we don't need to use this algorithm again and again till it gets updated. So from algorithm 1 we will get number of rule those will be stored in XML file with 3 attributes ID, support and confidence. And to store blank elements <blank> is used. So rules in XML file are sorted on the number of nodes of their antecedent used to optimistic query containing count operation. The use of this abstracted document is faster than firing query on main file.

V. CONCLUSION

The main objective are: 1) use of extracted knowledge to gain information by using query language; 2) to mine frequent association rules without considering any a-priori restriction on the structure and the content of the rules; 3) and can store mined information in XML format. As ongoing work, we can study for further optimization of mining algorithm.

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Proc. 20th Int'l Conf. Very Large Data Bases, pp. 478-499, 1994.

[2] Mirjana Mazuran, Elisa Quintarelli, Letizia Tanca, "Data Mining for XML Query-Answering Support," IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 8, August 2012.

[3] Mirjana Mazuran, Elisa Quintarelli, and Letiza Tanca, "Mining Tree-Based Association Rules from XML Documents," technical report, Politecnico di Milano, http://home.dei.polimi.it/quintare /Papers/MQT09-RR.pdf, 2009.

[4] Mirjana Mazuran, Elisa Quintarelli, and Letiza Tanca, "Mining Tree-Based Frequent Patterns from XML," Proc. Eighth Int'l Conf. Flexible Query Answering Systems, pp. 287-299, 2009.

[5] E. Baralis, P. Garza, Elisa Quintarelli, and Letiza Tanca, "Answering XML Queries by Means of Data Summaries," ACM Trans. Information Systems, vol. 25, no. 3, p. 10, 2007.

[6] World Wide Web Consortium. XQuery 1.0: An XML query language, 2007

[7] D. Barbosa, L. Mignet, and P. Veltri, "Studying the XML Web: Gathering Statistics from an XML Sample," World Wide Web, vol. 8, no. 4, pp. 413-438, 2005.

[8] C. Combi, B. Oliboni, and R. Rossato, "Querying XML Documents by Using Association Rules," Proc. 16th Int'l Conf. Database and Expert Systems Applications, pp. 1020-1024, 2005

[9] S. Gasparini and E. Quintarelli, "Intensional Query Answering to XQuery Expressions," Proc. 16th Int'l Conf. Database and Expert Systems Applications, pp. 544-553, 2005.

[10] J. Paik, H.Y. Youn, and U.M. Kim, "A New Method for Mining Association Rules from a Collection of XML Documents," Proc. Int'l Conf. Computational Science and Its Applications, pp. 936-945, 2005

[11] Y. Chi, Y. Yang, Y. Xia, and R.R. Muntz, "CMTreeMiner: Mining both Closed and Maximal Frequent Subtrees," Proc. Eighth Pacific-Asia Conf. Knowledge Discovery and Data Mining, pp. 63-73, 2004.

[12] B. Goethals and M.J. Zaki, "Advances in Frequent Itemset Mining Implementations: Report on FIMI 03," SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 109-117, 2004

[13] D. Braga, A. Campi, S. Ceri, M. Klemettinen, and P. Lanzi, "Discovering Interesting Information in XML Data with Association Rules," Proc. ACM Symp. Applied Computing, pp. 450-454, 2003

[14] T. Asai, H. Arimura, T. Uno, and S. Nakano, "Discovering Frequent Substructures in Large Unordered Trees," Technical Report DOI-TR 216, Dept. of Informatics, Kyushu Univ., http:// www.i.kyushu-u.ac.jp/doitr/trcs216.pdf, 2003.

[15] L. Feng, T.S. Dillon, H. Weigand, and E. Chang, "An XMLEnabled Association Rule Framework," Proc. 14th Int'l Conf. Database and Expert Systems Applications, pp. 88-97, 2003.

[16] J. W. W. Wan and G. Dobbie."Extracting association rules from xml documents using xquery," In WIDM '03: Proceedings of the 5th ACM international workshop on Web information and data management, pages 94{97, New York, NY, USA, 2003. ACM Press.

[17] T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto, and S. Arikawa, "Efficient Substructure Discovery from Large Semi-Structured Data," Proc. SIAM Int'l Conf. Data Mining, 2002.

[18] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy Preserving Mining of Association Rules," Proc. Eighth ACM Int'l Conf. Knowledge Discovery and Data Mining, pp. 217-228, 2002.

AUTHOR'S PROFILE

Mr. Prashant B. Vikhe has completed B.E Computer Engineering from University of Pune and pursuing M.E Computer Engineering from Amrutvahini College of Engineering, Sangamner, India.

Prof. Baisa L. Gunjal has completed her B.E. Computer from University of Pune and M.Tech in I.T. from Bharati Vidyapeeth, Pune, and Maharashtra, India. She is having 13 Years teaching experience in Computer Engineering. She is having 18 international and national journals and conferences publications. Presently she is working on "Watermarking research project" funded by BCUD, University of Pune. Her areas of interest include Image Processing, Advanced databases and Computer Networking.