ADVANCES in NATURAL and APPLIED SCIENCES

ISSN: 1995-0772
EISSN: 1998-1090
2016November10(16):pages 66-74

Published BY**AENSI Publication**
http://www.aensiweb.com/ANAS
Open Access Journal

A New and Innovative Method to Managing Semi Structured Data in Distributed Environment

[1]E. Seshatheri and [2]Dr.T.Bhuvaneswari

[1]Research Scholar, Manonmaniam Sundaranar University, Tirunelveli.
[2]Department of Computer Applications Queen Mary's College (Autonomous) Chennai

Received 23 August 2016; Accepted 1 November 2016; Published 20 November 2016

Address For Correspondence:
E. Seshatheri, Research Scholar, Manonmaniam Sundaranar University, Tirunelveli.

Copyright © 2016 by authors and American-Eurasian Network for Scientific Information (AENSI Publication).
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0/

Open Access

ABSTRACT

Background/Objectives: The large amount of internet causes enormous amount of data to be stored and processed which contains structured and semi structured data formats. Due to extreme size of semi structured document, results of a specific query may be massive which leads to retrieving interpretable knowledge a deadly process. The query-answering system makes it attainable to make queries and recuperate results for XML documents and reduce processing of tremendous sized files of data. In this research, it is proposed a method for retrieving more efficient more accurate results for the queries made by the users on the XML document. The original XML document is interpreted to Tree based Association Rules(TAR) files which were shaped by frequent patterns on the original document. **Methods/Statistical Analysis:** In this study, an approach to combine the concept of batch processing and stream processing to an end where we can query the data set which also supports Adhoc Querying with less latency, that can be run on any Large scale Machine Learning Algorithms for recognizing any interest pattern in the streaming data set was employed. **Conclusion:** The Tree based Association Rules files provide considered information on the structure and content of XML document. Also here it's using CMTree Miner algorithm to mine most frequent set of XML sub-trees, and also comparing Bit cube, Range Cube and Proposed method with several existing approaches.

KEYWORDS: XML document, Query-answering, Tree-based Association Rule, Data Mining

INTRODUCTION

The goal of data mining is to extract or mine knowledge from large amounts of data. Consequently, data mining consists of more than gathering and managing data, it also includes analysis and prediction. The extensible Markup Language (XML)[1] has become a standard language for data depiction and exchange XML is a Standard, flexible syntax for data exchanging Regular, Structured and Semi Structured data. Mining of XML documents significantly differs from structured data mining and text mining. The XML allows the representation of semi-structured and hierarchical data containing not only the values of individual items, but also the relationships between data items. Due to the inherent flexibility of XML, in both structure and semantics, discovering knowledge from XML data is faced with new challenges as well as benefits. Mining of structure along with content provides new insights and means into the process of knowledge discovery.

As for query-answering, since query languages for semi structured data depend on the document structure to carry its semantics, in order for query formulation to be actual users need to know this structure in advance, which is often not the case. This limitation is a crucial problem which did not arise in the context of relational database management systems(RDBMS). As a consequence, when accessing for the first time a large dataset, gaining some general information about its main structural and semantic characteristics helps study on more specific details. This research work addresses the need of getting the essence of the document earlier querying it,

To Cite This Article: E. Seshatheri and Dr.T.Bhuvaneswari., A New and Innovative Method to Managing Semi Structured Data in Distributed Environment. **Advances in Natural and Applied Sciences**. 10(16);Pages: 66-74

both in terms of content and structure. Determining recurrent patterns inside XML documents provides high-quality knowledge about the document content: frequent patterns are in fact intentional information about the data contained in the document itself, that is, they specify the document in terms of a set of properties rather than by means of data. As opposed to detailed and precise information conveyed by the data, this information is partial and often approximate, but synthetic, and concerns both the document structure and its content.

1.1 Tree-Based Association Rules from XML Document:

Association rules [2] describe the co-occurrence of data items in a large amount of collected data and are usually represented as implications in the form $X \Rightarrow Y$, where X and Yare two arbitrary sets of data items, such that $X \cup Y = \Phi$;.The quality of an association rule is usually measured by means of support and confidence. Support corresponds to the frequency of the set $X \cup Y$ in the dataset, while self-assurance corresponds to the conditional probability of finding Y, having found X and is given by $sup(X \cup Y) = sup(X)$.In this work we extend the notion of association rule originally introduced in the context of relational databases, in order to adapt it to the hierarchical nature of XML documents. In particular, it is consider the element-only Info set content model [27], which allows an XML no terminal tag to include only other elements and/or attributes, while the text is confined to terminal elements. Furthermore, without loss of generalization, here it is do not consider some features of the Info set that are not relevant to the present work, such as names ID-IDREF attributes, URIs, and Links.

Following the Infoset conventions, we represent an XML document by a labeled tree (N,E,γ) where N is the set of nodes, γ€ N is the root of the tree (i.e. the root of the XML document), E is the set of edges. Moreover, the following gproperties on nodes and edges hold:

1) Each node$_i$ has a tuple of labels NL_i = {N *tag*$_i$; N *type*$_i$; N *content*$_i$);the type label N*type*$_i$ indicates whether the node is the root, an element, text, or attribute, whereas the label N*content*$_i$ can assume as value a PCDATA or ⊥(undefined, for non terminals).

2) Each edge $e_j = h(n_h,n_k),EL_j)$,with n_h and n_k in N, has a label ELj = (E*type*$_j$), E*type*$_j$€{attribute of, sub-element of}.

Note that edges represent the “containment" relationship between different items of an XML document, thus edges do not have names. In this work interested in finding relationships among sub trees of XML documents. Thus, here do not distinguish between textual content of leaf elements and value of attributes. As a consequence, in order to draw graphical concepts in a more readable way, here do not report the edge label and the node type label. Attributes and elements are characterized by empty circles, whereas the textual content of elements, or the value of attributes, is reported under the outgoing edge of the element or attribute it refers to.

2. Literature Survey:

The problem of association rule mining was initially proposed and many implementations of the algorithms were developed in the database literature. Some methods use X Query to extract association rules from simple XML documents. They propose a set of functions, written in X Query, which implement the Apriori algorithm.

The problem related to XML context was produced in the year 2003 by J.W. Won &G. dobbie which uses X Query to extract association rules. This approach performs well on simple XML documents but not on complex XML document with irregular structure also the tool called X Query is a language which is used for finding & extracting element, attributes from XML documents[2].

Braga et al, [16], introduced a proposal to enrich XQuery with data mining and knowledge discovery capabilities, by introducing XMINE RULE, an operator for mining association rules for native XML documents. They formalize the syntax and semantics for the operator and propose some examples of complex association rules. However, XMINE is based on the MINE RULE operator, which works on relational data only. This means that, after a step of pruning of unnecessary information, the XML document is translated into the relational format.

Wan and Dobbie [2] show that their approach performs well on simple XML documents but it is very difficult to apply to complex XML documents with an irregular structure.

This limitation is overcome by where Braga et al, [1], introduced a proposal to enrich XQuery with data mining and knowledge discovery capabilities, by introducing XMINERULE, an operator for mining association rules for XML documents. They formalize the syntax and semantics for the operator and propose some examples of complex association rules. However, XMINE is based on the MINERULE operator, which works on relational data only. This resources that, after a step of pruning of unnecessary information, the XML document is translated into the relational format. Moreover, in many techniques, the designer is forced to specify the structure of the rule to be extracted and then to mine it, if possible. This means that the designer has to specify what should be contained in the body and head of the rule, i.e., the designer has to know the structure

of the XML document in advance, and this is an irrational requirement when the document does not have a Document Type Definition. A document type definition (DTD) is a set of markup declarations that define a document type for an SGML-family markup language (SGML, XML and HTML). A DTD uses a terse formal syntax that declares precisely which elements and references may appear wherein the document of the particular type, and what the elements' contents and attributes are.

Disadvantages of Existing System

Search intention for a keyword query is not easy to determine. It returns low result quality in term of query relevance. Rank the individual matches of all these queries are challenging. In our research, the above all disadvantages is also incorporated and discover the new technique.

3. Problem Statement:

- The goal of data mining is to extract or mine knowledge from huge amounts of data. Thus, data mining consists of more than collecting and managing data. Sometimes, it also includes analysis and prediction of the data. The XML allows the representation of hierarchal and semi-structured data containing not only the values of individual items but also the relationships between data items from the different type of database. Due to the intrinsic flexibility of XML, discovering knowledge from XML data is faced with new challenges as well as benefits.

Usually people have attempted to handle the XML processing problems using one of four methods as follows:

(1) focus on techniques for "shredding" XML into tables, followed by combining the tables, and later re-joining the results to produce required XML output;

(2) make a few modifications to object oriented or semi-structured databases, which also inherently supports hierarchy, hence they support the XML;

(3) use a top-down tree-traversal strategy for executing queries;

(4) use a custom wrapper at the source end for index like retrieval of only the necessary content. Before we describe the Tukwila architecture, it is useful to briefly examine these previous approaches, including their relative strengths and weaknesses.

4. Objectives of the Proposed System:

1.Query answering system using Tree Based Association Rules on XML document to extract the most relevant feeds from the large file directly.

2.To provide an approach which actually find frequent pattern and Tree Based Association Rules (TAR) from the XML file from various sources/documents which will help to improve performance and retrieve relevant information from the XML document.

5. Proposed System – Experimental Design:

The proposed method for query-answering system is based on the association mining. The XML document (rss, XML or doc) is given as input. The XML tree is constructed using XML parser which is extracted for the user defined values of support and confidence. This system provides the intentional knowledge in the form of XML document for the entire XML documents.

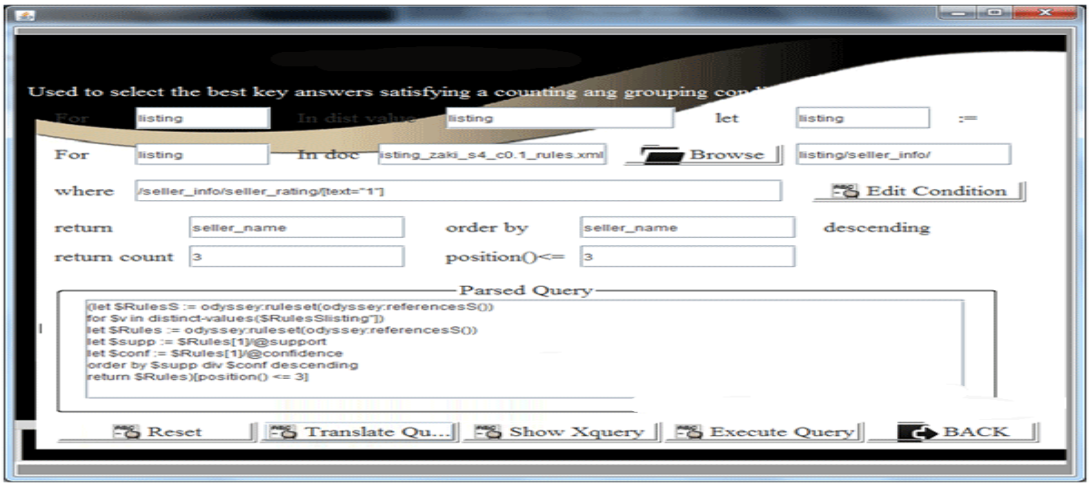


Fig. 1: XML Query Processing using WHERE Clause

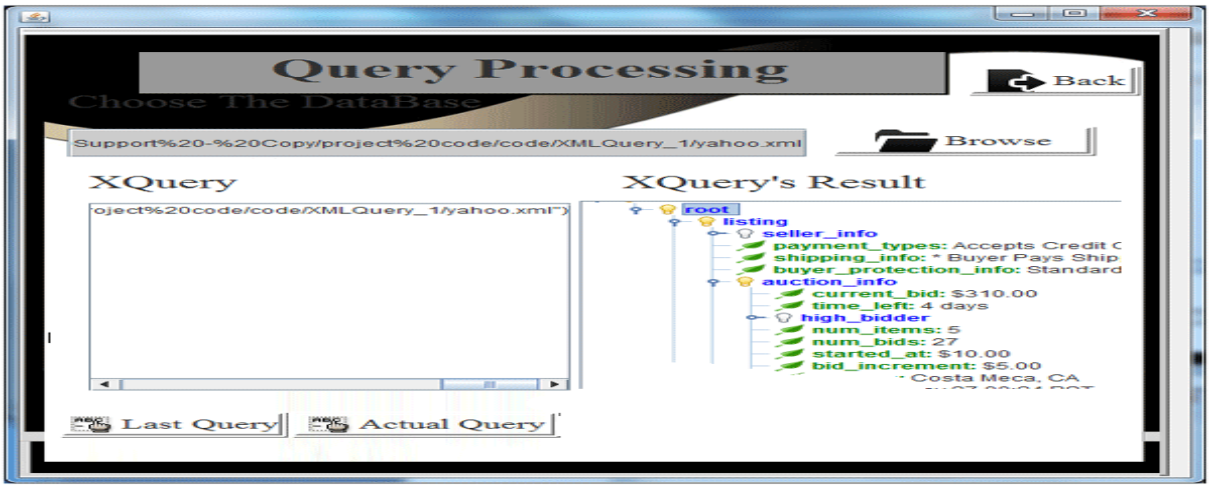


Fig. 2: X Query processing results

Then the query will be applied or XML and appropriate clause, document which was sit for by the users. This gives the result related to the query. Then the XML document is modified and the process is repeated until goal reached. The modifications are stored in the same XML document which was already formed by the values of support and confidence combine all this document parsed is evaluate as again when required.

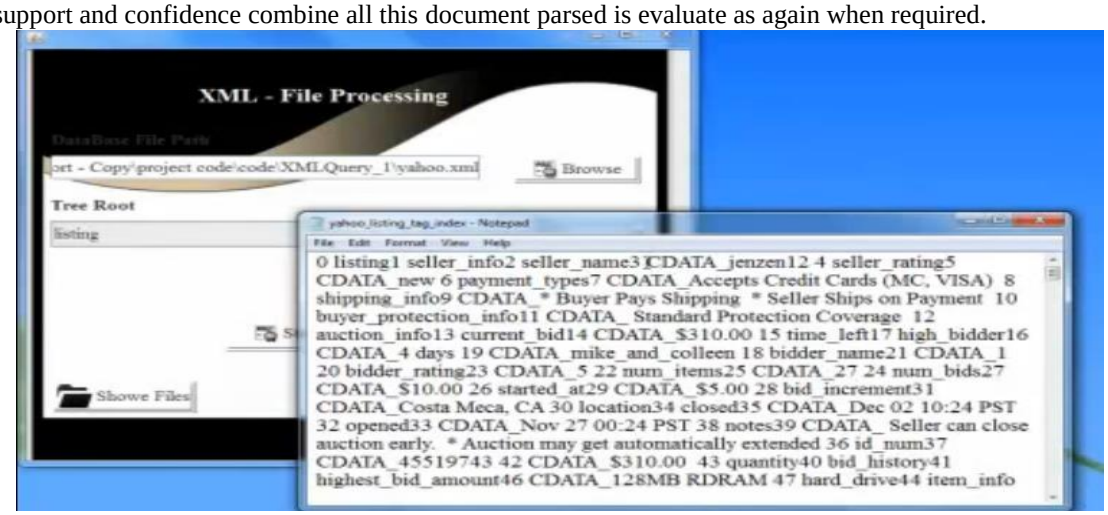


Fig. 3: XML File Processing using Data File

5.1 Construction of Trees:

The XML documents have a flexible architecture. XML documents can be pre-processed. XML pre-processing is done by XML parser. The DOM (Document Object Model) parser is used here which is used to construct the tree from the XML document. Before a XML document can be accessed, it must be loaded into an XML DOM object. The DOM creates a tree structure in the internal memory from the given XML document. It stores the all the document into memory before processing those documents. It allows the users to traverse the document using top-down approach XML trees, access, insert, update the content, style and structure of the document and also to delete the nodes from the tree. Therefore XML document forms a tree structure. Also the XML document should be validated (i.e) the tags should be started and ended correctly without leaving any tag without its pair.

5.2 Extraction of Frequent Pattern:

Association rules [8] describe the frequent occurrence of data items in a large amount of data collected. X and Y are the two considered data items. Now they are represented in the form of X∪Y. Here Association rule is measured by means of Support and Confidence. In this the Support represents the frequency of the set (X and Y) found in the data set and Confidence represents the conditional probability of finding Y, having got X. The interesting patterns among the sub trees of the given XML document can be identified. A Tree-based Association Rule (TAR)[8], the form of XML document defines pattern of sub trees in the XML document. The

entire XML document can be accessed by providing the values of support and confidence. Mining TAR is a two step process[8].

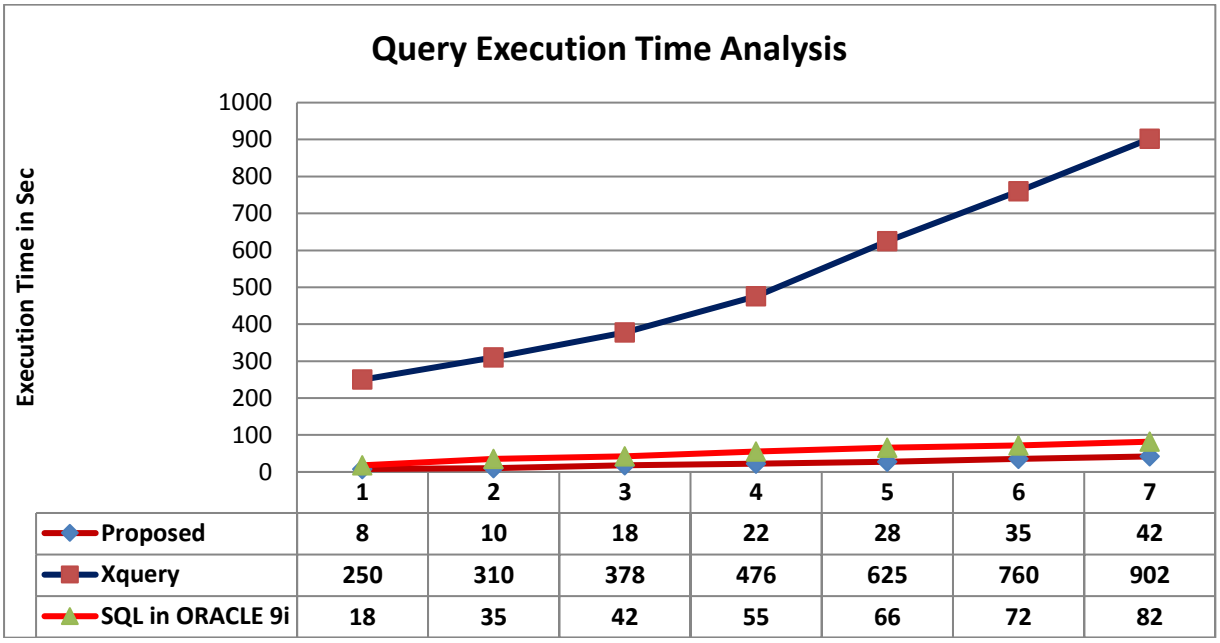
This includes:
1. Mining frequent sub trees
2. Computing interesting rules

After obtaining the set of files the Proposed model will merge them to obtain one XML document which contains data from all the included files. The step next to this is to obtain the TAR of all the files. Once it is done, the new novel Proposed model will give the most frequent feeds of all the files. After that the feed search will be performed by the Proposed model which will then give us the filtered result. The obtained results is better than the other methods. The Comparative results are analyzed in the last section.

6. Experimental Results:

Here in this research used Oracle 9i (Enterprise Edition Release9.2.0.8.0) and Stylus Studio 2008 XML Enterprise Suite Release 2, to evaluate the different query results in the case of a centralized database system. Here in this research used the Visual C++ language using Borland C compiler (32 bit that supports up to 4 GB RAM) to implement our Proposed technique. It is used an Intel Processor with 2.13 GHz,4 GB of RAM under the Windows 7UltimateOperating system.

To support the Oracle 9i database is used the Windows 7 Ultimate operating system. Here in this work used the (well structured) XML datasets in [20] to run comparisons of the X Query language and our Proposed technique using file sizes of 2.5MB, 5.6MB, 10.8MB, 23.48MB and 110.35MB.



	1	2	3	4	5	6	7
Proposed	8	10	18	22	28	35	42
Xquery	250	310	378	476	625	760	902
SQL in ORACLE 9i	18	35	42	55	66	72	82

Fig. 4: Query execution time analysis.(Number of predicates using AND Condition WHERE Clause)

The experimental results for X Query execution time, execution time in Oracle and proposed are presented in Figure 4, which for the purpose of clarity only depicts the results for 110.35MB files, but which show comparative performance that is typical of all the tested file sizes. It can be seen that proposed method's execution times were substantially lower than those of XQuery, and also that proposed method's outperforms the highly regarded Oracle execution times across the range of predicates tested. The results show that the improvement achieved by Proposed method increases with the number of predicates. A second set of experiments was performed to compare the query execution times of Proposed Method, Range Cube and Bit cube[21].These experiments used 5, 10, 15, 20 and 25 element

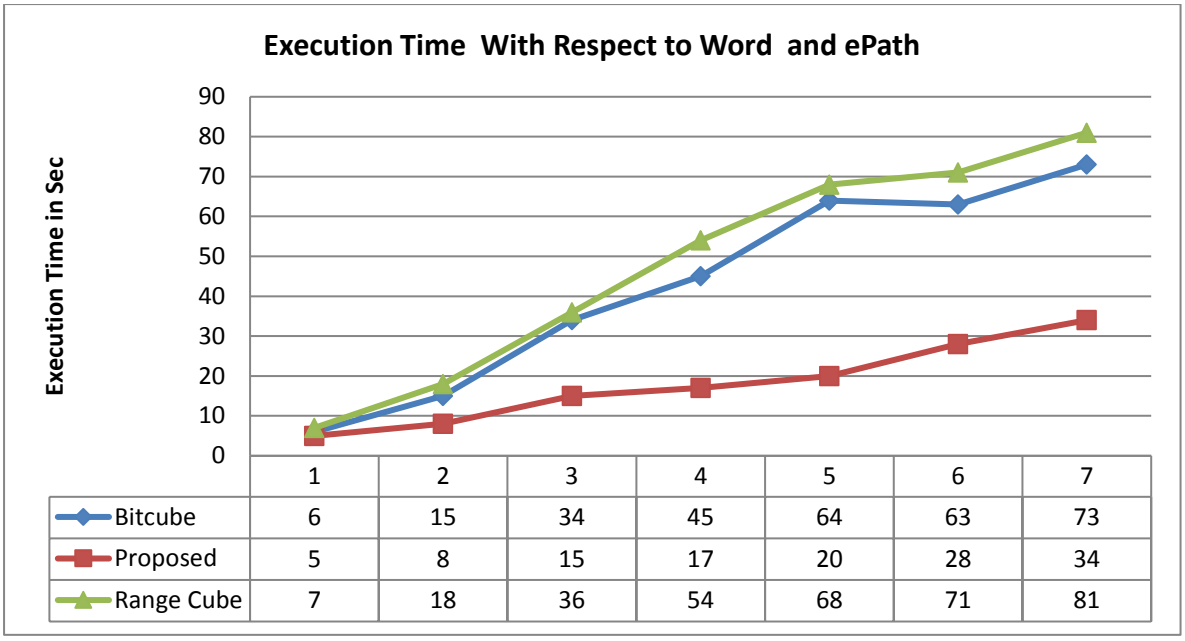


Fig. 5: Execution time wrt word/ePath.

paths (ePaths) per document, and a variety of words per element path. For all numbers of ePaths per document, and all numbers of documents, proposed method outperformed. Bitcube as shown in Figure 5, which for the purpose of clarity only depicts the results for 25 e Paths per document, and 20 words per ePath.

III. Implementation Details:

In particular, the idea of mining association rules [1] to provide summarized representations of XML documents has been investigated in many proposals either by using languages (e.g., XQuery [2]) and techniques developed in the XML context, or by implementing graph-or-tree-based algorithms. This work proposes to develop an application to mine frequently occurring sub trees from the generated XML data set by RSS Feed links of different newspaper links. Then storing them in another XML document. These rules are extracted only if support and confidence of nodes is greater than the provided threshold values. Then apply our Frequent Pattern Miner algorithm for XML. The idea of using association rules as summarized representations of XML documents was also introduced in [4] where the XML summary is based on the extraction of rules both on the structure (schema patterns) and on content (instance patterns) of XML data sets. The limitations of this approach are: 1) the root of the rule is established a-priori and 2) the patterns, used to describe general properties of the schema applying to all instances, are not mined, but derived as an abstraction of similar instance patterns and are less precise and reliable.

A. Fundamental Concepts:

The idea of the project basically revolves around the concept of Tree base Association rule. Mining tree-based association rules is a process composed of two steps:

1. Mining frequent sub trees from the XML document;
2. Computing interesting rules from the previously mined frequent subtrees.

Association rule [1] is an hint of the form A U B, where the rule A and B are subset of the set C of element in a set of transactions D and A ł B= Ø. A rule X Y states that the transaction T that has the elements in A are likely to contain also the elements in B. Association rules are distinguished by two survey: the support, which gives the percentage of transactions present in D that contain both items A and B (AU B); the confidence, which calculates the percentage of transactions in D containing the element A that also contain the elements B (support (A U B)/support (B)). In XML context, both D and C are group of trees. In this work TAR is generated using Natural Language Processing. The project works for RSS news links of multiple newspapers. RSSst ands for "Really Simple Syndication". It is a way to easily distribute a list of headlines, update notices, and sometimes content to a wide number of people. It is used by computer programs that organize those headlines and notices for easy reading. RSS works by having the website author maintain a list of notifications on their website in a standard way. This

list of notifications is called an "RSS Feed". People who are interested in finding out the latest headlines or changes can check this list. Special computer programs called "RSS aggregators" have been developed that automatically access the RSS feeds of websites you care about on your behalf and organize the results for you.

Producing an RSS feed is very simple and hundreds of thousands of websites now provide this feature, including major news organizations like the New York Times, the BBC, and Reuters, as well as many weblogs. RSS provides very basic information to do its notification. It is made up of a list of items presented in order from newest to oldest. Each item usually consists of a simple title describing the item along with a more complete description and a link to a webpage with the actual information being described. Sometimes this description is the full information you want to read (such as the content of a weblog post) and sometimes it is just a summary.

B. Architecture Details:

This paper introduces a proposal for mining and storing Tree-Based Association Rules (TARs) as a means to represent intentional knowledge as an alternative, synthetic data set would be queried for providing quick and summarized answers. For calculating the TAR we make use of Natural Language Processing. The algorithm for finding TAR is given below:

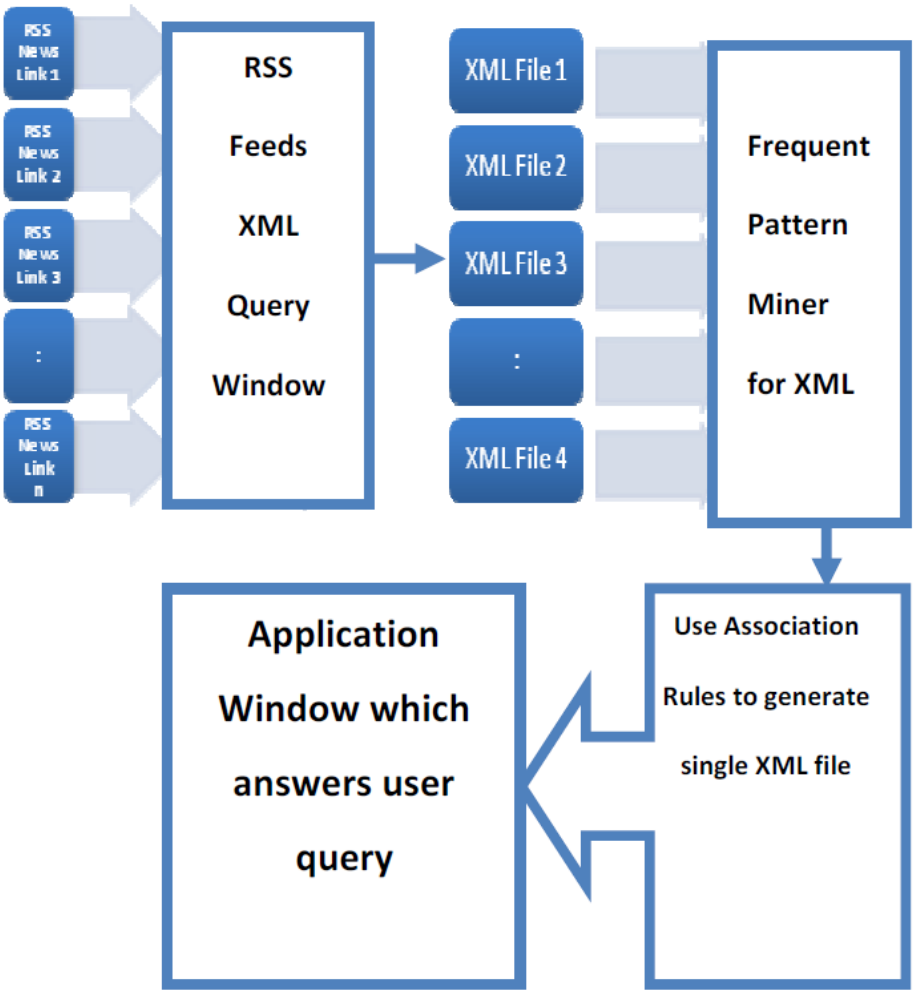


Fig. 1: Operation Flow for Frequent Pattern Miner for XML

Algorithm: Extracting Tree Based Association Rules from XML Files
Input: RSS Feed Files R_n, Similarity Threshold T_h, Support S, Confidence C.
Output: XML Tree Based Association Rules
1. Read all Rn by using XML Reader.
2. Extract Inner Element <Title> Tags within all Rn and insert them into a new XML File xf.
3. Initialize Node List Lst to contain Visited T_{sub} in xf.
4. Initialize Item Set, a Key Value Pair structure to contain Element e and its support fs.
5. For Each Subtree Tsub in Xf
6. Extract <Description> Tag I_{desc} from T_{sub}
7. If(Count(Lst)==0)
8. Then Add I_{desc} into L_{st}

9. Else For Each element ei in L_{st}
10. Similarity Score = Compare(I_{desc}, ei)
11. If(Similarity Score>= T_h)
12. Then If Item Set contains ei
13. Then Increment support of eiby 1 from ItemSet
14. Else Add ei to Item Set and increment support of ei by 2
15. Else Add ei to Item Set and increment support of ei by 1
16. Add I_{desc} into L_{st}
17. Using Item Set as reference data structure containing Item ei and its support fs , desired support C and confidence C, generate frequent ItemSet F.
18. For Each Item Set Ifr in F
19. Insert Ifr in XML Document XTAR
20. Return X_{tar}

Using the concept of TAR in conjunction with Natural Language Processing is very effective. Here since all patterns are matched against each other a detailed comparison is made resulting in an effective frequent pattern matching.

Table 1: Comparative Analysis of Implemented Approach with existing Document summarization and Information Retrieval Methods

Methods	IR Ranking	Type of Input information	Generate Frequent Patterns?	Use of Dictionary	Query Independent	Applicable for Smart Phone Devices?	Applicable on Web Pages?
Automatic Document Summarization By Sentence Extraction	Statistical	Structured Only		No	No	No	No
Query – Specific Document Summarization	Statistical	Structured Only		No	No	No	No
A Language Independent Algorithm for Single and Multiple Document summarization	NLP	Structured Only		Yes	Yes	No	No
OLEDOT	NLP	Structured and Non structured data		Yes	Yes	No	Yes
Implemented Approach	NLP	Structured and Non structured data		Yes	Yes	Yes	Yes

Conclusion:

Mine all frequent association rules restriction on the structure and the content of the rules; Store mined information in XML format. Use the extracted knowledge to gain information about the original data sets. Update mined Tree based Association Rules when the original XML datasets change.

REFERENCES

1. Nirmalrani V and Sakthivel P. (2015), A Hybrid Access Control Model with Multilevel Authentication and Delegation to Protect the Distributed Resources. Journal of Pure and Applied Microbiology (JPAM). 2015 November; 9(Spl. Edn. 2); 595-609. ISSN 0973-7510.
2. Jabez J, Muthu Kumar B. Intrusion Detection System: time probability method and hyperbolic hop field neural network. Journal of Theoretical and Applied Information Technology. 2014 September; 67(1):65-77. Paik, J., H.Y. Youn and U.M. Kim, 2005. “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.
3. V. Kasthuri Muthu, A. Sameera T. “Mining Algorithm for XML Query Answering Support”, ijcset, vol4, 2013.
4. Arundhati Birari1, Prof. Ranjit Gawande2, 2013. “Mining Tree-Based Association Rules for XML Query Answering”, ijettcsvol 2: 3.
5. Wan, J.W.W. and G. Dobbie, 2003. “Extracting Association Rules from XML Documents Using XQuery,” Proc. Fifth ACM Int’l WorkshopWeb Information and Data Management, pp: 94-97.
6. Barbosa, D., L. Mignet and P. Veltri, 2005. “Studying the XML Web:Gathering Statistics from an XML Sample,” World Wide Web, 8(4): 413-438.

7. Suganya, I., N. Velmurugan, Dr.P. Ganeshkumar, 2013. "XML Query-Answering Support System using AssociationMining Technique" IEEE conference on ICT.
8. MarouaneHachicha and Je´ roˆmeDarmont, Member, 2013. IEEE Computer Society, "A Survey of XML TreePatterns", IEEE 25.
9. Yun Chi, Yirong Yang, Yi Xia and Richard R. Muntz, 2010. "CMTreeMiner: Mining Both Closed and MaximalFrequent Subtrees", International Conference on Knowledge Discovery and Data Mining.
10. Wang Lian, Nikos Mamoulis, David Wai-lokCheung, 2005. "Indexing Useful Structural Patterns for XML QueryProcessing", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 17.
11. Neoklis Polyzotis, Minos Garofalakis, Yannis Ioannidis, 2003. "Approximate XML Query Answers",Intl. Conf. onVery Large Data Bases.
12. Devi Mahalakshmi, S., Dr. K. Vijayalakshmi, Dr. K. Muneeswaran, G. Priyanka, 2010. "Mining Intensional Information for answering XML-Queries using Tree-based Association Rules Approach", IEEE.
13. Chandra Sekhar, K, Dhanasree, 2012. "Extracting TARs from XML for Efficient Query Answering", International Journal of Computer Science and Network (IJCSN) 1: 6.
14. Agarwal, R. and R. Srikant, 1994. "Fast Algorithms for Mining Association Rules in Large Databases,"Proc.20th Int'l Conf. Very Large Data Bases, pp: 478-499.
15. Borgwardt, K.M., H.P. Kriegel, P. Wackersreuther, 2006. "Pattern Mining in Frequent Dynamic Subgraphs,"Proc. ICDM.
16. Braga, D., A. Campi, S. Ceri, M. Klemettinen and P. Lanzi, 2003. "Discovering Interesting Information in XMLData with Association Rules," Proc. ACM SYmp. Applied Computing, pp: 450-454.
17. Gasparini, S. and E. Quintarelli, 2005. "Intensional Query Answering to XQuery Expressions," Proc. 16thInt'l Conf. Database and Expert Systems Applocations, pp: 544-553.
18. Goldman, R. and J. Widom, 1997. "DataGuides: Enabling Query Formulation and Optimization in SemistructuredDatabases," Proc. 23rd Int'l Conf. Very Large Data Bases, pp: 436-445.
19. Mazuran, M., E. Quintarelli and L.Tanca, 2012. "Data Mining for XML Query-Answering Support," IEEETans. Knowledge and Data Eng., 17(8): 1021-1035.
20. Paik, J., H.Y. Youn and U.M. Kim, 2005. "A New Method for Mining Association Rules from a Collectionof XML Documents," Proc.Int'l Conf. Computational Science and Its Applications, pp: 936-945.
21. Asai, T., K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto and S. Arikawa, 2002. Efficient Substructure Discovery from Large Semi-Structured Data,Proc. SIAM Int Conf. Data Mining.
22. Xiao, Y., J.F. Yao, Z. Li and M.H. Dunham, 2003. Efficient Data Mining for Maximal Frequent Subtrees,Proc. IEEE Third Int. Conf. Data Mining, pp: 379-386.
23. Zaki, M.J., 2005. Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications, IEEE Trans. Knowledge and Data Eng.,17(8): 1021-1035.
24. Chi, Y., Y. Yang, Y. Xia and R.R. Muntz, 2004. CMTreeMiner: Mining both Closed and Maximal Frequent Subtrees, Proc. Eighth Pacific- Asia Conf. Knowledge Discovery and Data Mining, pp: 63-73.
25. Termier, A., M. Rousset, M. Sebag, K. Ohara, T. Washio and H. Motoda, 2008. DryadeParent, an Efficient and Robust Closed Attribute Tree Mining Algorithm ,IEEE Trans. Knowledge and Data Eng, 20(3): 300-320.