ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

A Novel Method to Managing Semi Structured Data in Distributed Environment usingModified Tree based Association Rules(TAR)

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

[1]*Research ScholarManonmaniamSundaranar UniversityTirunelveli.*
[2]*Assistant ProfessorDepartment of Computer Science and ApplicationL.N. Government Arts CollegePonneri*

ARTICLE INFO

Article history:
Received 10 October 2015
Accepted 30 November 2015
Available online 24 December 2015

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

ABSTRACT

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 or data, 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 reduced processing of big 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 Modified Tree based Association Rules(TAR) files which were shaped by frequent patterns on the original document. 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(i.e., MTAR)method with several existing approaches.

ToCite ThisArticle:E. Seshatheri and Dr.T. Bhuvaneshwari.,A Novel Method to Managing Semi Structured Data in Distributed Environment usingModified Tree based Association Rules(TAR). ***Aust. J. Basic & Appl. Sci.,*** *9(35): 277-286, 2015*

INTRODUCTION

The goal of data mining is to extract or mineknowledgefrom large amounts of data. Consequently, data miningconsists of more than gathering and managing data, it alsoincludes analysis and prediction. The eXtensible MarkupLanguage (XML) (Agrawal, R. and R. Srikant, 1994) has become a standard language for datadepiction and exchange XML is a Standard,flexible syntax for data exchanging Regular, Structured and Semi Structured data.Mining of XML documents significantly differs fromstructured 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 characteristic flexibility of XML, in both structure and semantics, discovering knowledge from XML data is faced with new challenges as well as assistances. 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 in a file to carryits semantics, in order for query formulation to be actualusers need to know this structure in advance, which is often not thecase. This limitation is a crucial problem which did notarise in the context of relational database managementsystems (RDBMS). As a consequence, when accessing for the first timea large dataset, gaining some general information about itsmain structural and semantic characteristics helpsstudy on more specific details. This research work addressesthe need of getting the essence of the document earlierquerying it, both in terms of content and structure. Determining regular patterns inside XML documents provides high-quality knowledge about the document content:frequent patterns are in fact intentional information about thedata contained in the document itself, that is, they specify thedocument in terms of a set of properties rather than by meansof data. As different to detailed and precise informationconveyed by the data information, this information is partial and oftenapproximate, but synthetic, and concerns both the documentstructure and its content.

1.1 Tree-Based Association Rules from XML Document:

Association rules 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,

Corresponding Author: E. Seshatheri, Research Scholar ManonmaniamSundaranar University Tirunelveli.
E-mail: seshathriphd@gmail.com

such that X∪Y =Φ;.The quality of an association rule is usually measured by means of support and confidence. Support corresponds tothe frequency of the set X∪Y in the dataset, while self-assurance corresponds to the conditional probability of findingY, having found X and is given by sup(X∪Y)=*sup*(X).In this work here it is extend the notion of association rule originallyintroduced in the context of relational databases, inorder to adapt it to the hierarchical nature of XML documents.In particular, it is consider the element-only Info set contentmodel, which allows an XML nonterminal tag to includeonly other elements and/or attributes, while the text is confined to terminal elements. Furthermore, without loss ofgeneralization, here it is do not consider some features of the Info setthat are not relevant to the present work, such as namesID-IDREF attributes, URIs, and Links.

Following the Infoset conventions, here it is represent an XML documentby a labeled tree (N,E,γ) where N is the set ofnodes, γ€N is the root of the tree (i.e. the root of theXML document), E is the set of edges. Moreover, the followingproperties on nodes and edges hold:

1) Each noden_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 theroot, an element, text, or attribute, whereas the labelN*content*$_i$ can assume as value a PCDATA or ⊥(undefined, for no terminals).

2) Each edge $e_j = h(n_h,n_k),EL_j)$,with nh 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 XMLdocument, thus edges do not have names.Here in this work interested in finding relationships among subtreesof 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.

1.2 Fundamental concepts:

Given two labeled trees T =<NT;E_T;r_T> and S=<N_S;E_S; r_S>,S is said to be an induced subtree of T if and only if there exists a mapping ɵ : N_S-->N_T such that$\forall_{ni}$ЄN_S;NL_i= NL_j , where ɵ(ni) = n_j and for each edgee_j = h(n1; n2);EL_{ji}Є E_S; h(ɵ(n_1); ɵ(n_2));EL_{ji}Є E_T.

Figure 1 shows an example of an XML document (Figure1(a)), its tree-based representation (Figure 1(b)) and threeinduced subtrees of the document (Figure 1(c)).

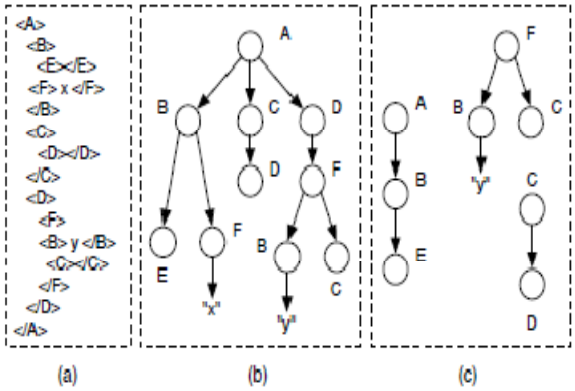


Fig.1: a) an example of XML document, b) its tree-based representation, and c) three induced sub-trees

A Tree-based Association Rule (TAR) is a tuple of theformT_r = ⟨S_B; S_H; s_{Tr} ;c_{Tri}, where S_B =< N_B;E_B; r_B>and S_H =< N_H;E_H; r_H> are trees and s_{Tr} and c_{Tr} arereal numbers representing the support and confidence of therule respectively. Furthermore, S_B is an induced subtree ofS_H with an additional property on the node labels. Indeed,the following properties must hold:

N_B⊆ N_H

• E_B⊆ E_H and $\forall_n$;mЄ N_B; ⟨(n;m);EL›Є$E_B$iff⟨(n,m),EL›ЄE_H

• the set of tags of S_B is equal to the set of tags of S_H with the addition ofthe empty label “Є".

Australian Journal of Basic and Applied Sciences, 9(35) November 2015, Pages: 277-286

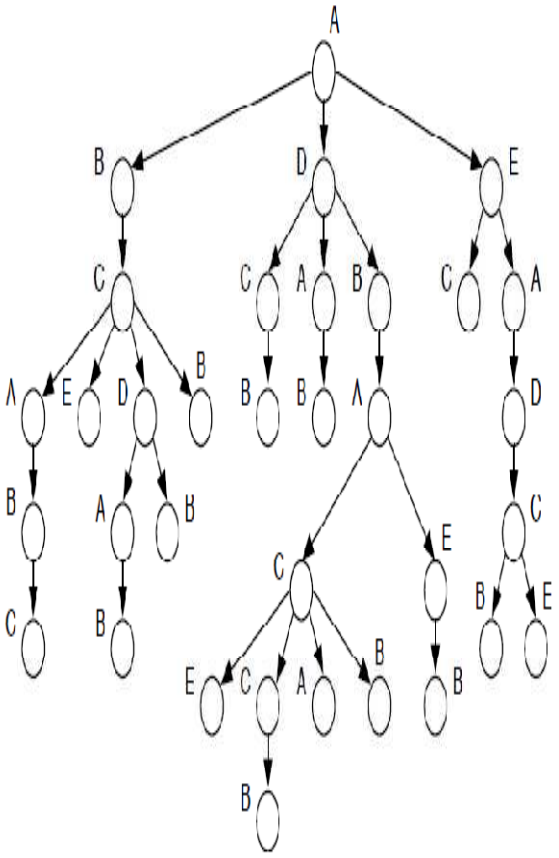


Fig.2: Sample dataset

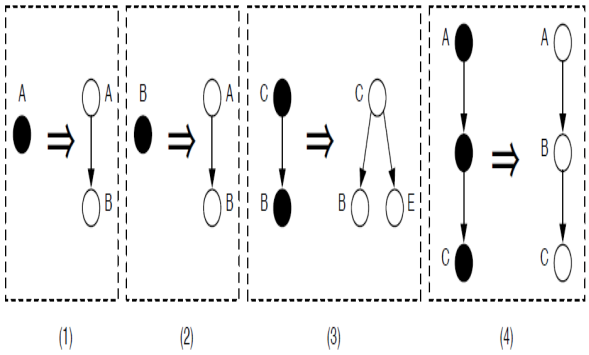


Fig.3: Sample sTARs (structure Tree-based Association Rules)

The empty label is introduced because the body of a rulemay contain nodes with unspecified tags (a sort of placeholdernodes) and these tags will be declared in the headpart of the rule (see the rule (2) of Figure 3). For the sakeof clarity the label _ is omitted in the figures and all nodeswith empty labels do not present any label at all.It is worth noticing that Tree-based association rules aredifferent from XML association rules, because the firstrequire that (X 6_ Y) ^ (Y 6_ X), i.e. the two trees X andY have to be disjunct, while Tree-based association rulesrequire that X is an induced subtree of Y

A TAR represents intensional knowledge in the form S_B)=>S_H, where S_B is the body tree and SH the head tree ofthe rule. Indeed, the rule S_B) S_H states that if the treeS_B appears in an XML document D, it is likely that thebigger", or more specific", tree S_H also appears. In ourgraphical representation, we will render the nodes of thebody of a rule by black circles, and the nodes of the headby empty circles.

Thus, every tree-based association rule is characterized bytwo measures:s_{Tr} support, measures the frequency of the tree S_H in theXML documentc_{Tr} confidence, measures the reliability of a rule, that is thefrequency of the tree S_H, once S_B has already beenfoundGiven function count(S;D) denoting the number of occurrencesof a subtree S in the tree D and function cardinality(D) denoting the number of nodes of D, it is possible todefine formally the two measures as:

2. Literature Survey:

The problem of association rule mining was initiallyproposed and many implementations of the algorithms weredeveloped in the database literature. Some methods useXQuery to extract association rules from simple XMLdocuments. They propose a set of functions, written inXQuery, 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 XQuery 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 XQuery is a language which is used for finding & extracting element, attributes from XML documents.

An algorithm PATHJOIN is Proposed by Y. Xiao et.al to discover all maximal frequent sub trees given some minimum support threshold. Mohammed J. Zaki, has introduced the mining of embedded sub trees in a (forest) database of trees and introduced a novel algorithm, TREEMINER, for tree mining. TREEMINER uses depth-first search; it also uses the scope-list vertical representation of trees. The limitation of this algorithm is that it cannot find structure of XML document.

Another algorithm is presented by Yun Chi, Yirong Yang, Yi Xia, and Richard R. Muntz, called CMTreeMiner, a computationally efficient algorithm that discovers all closed and maximal frequent sub trees in a database of rooted unordered trees.

Termier*et al*. show that DRYADEPARENT is currently the fastest tree mining algorithm. However, Dryadeparent extracts embedded sub trees which are trees that maintain the ancestor relationship between nodes but do not distinguish, among the ancestor-descendant pairs and the parent- child ones.

Wan and Dobbie 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*, introduced 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 semanticsfor the operator and propose some examples of complexassociation 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 havea Document Type Definition. A document type definition (DTD) is a setof markup declarations that define a document type foran SGML-family markup language (SGML, XML andHTML). A DTD uses a terse formal syntax that declaresprecisely which elements and references may appear wherein the document of the particular type, and what theelements' 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 arechallenging.

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 miningconsists of more than collecting and managing data. Sometimes, it also includes analysis and prediction of thedata. The XML allows the representation of hierarchal and semi-structured data containing not only the values ofindividual 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-joiningthe results to produce required XML output;

(2) make a few modifications to object oriented or semi-structured databases, which also inherently supportshierarchy, 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 here it is describe the Tukwila architecture, it is useful to briefly examine these previous approaches, including theirrelative 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 filedirectly.

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 XMLdocument (rss, XML or doc) is given as input. The XML tree is constructed using XML parser which isextracted for the user defined values of support and confidence. This systemprovides the intentionalknowledge in the form of XML document for the entire XML documents.

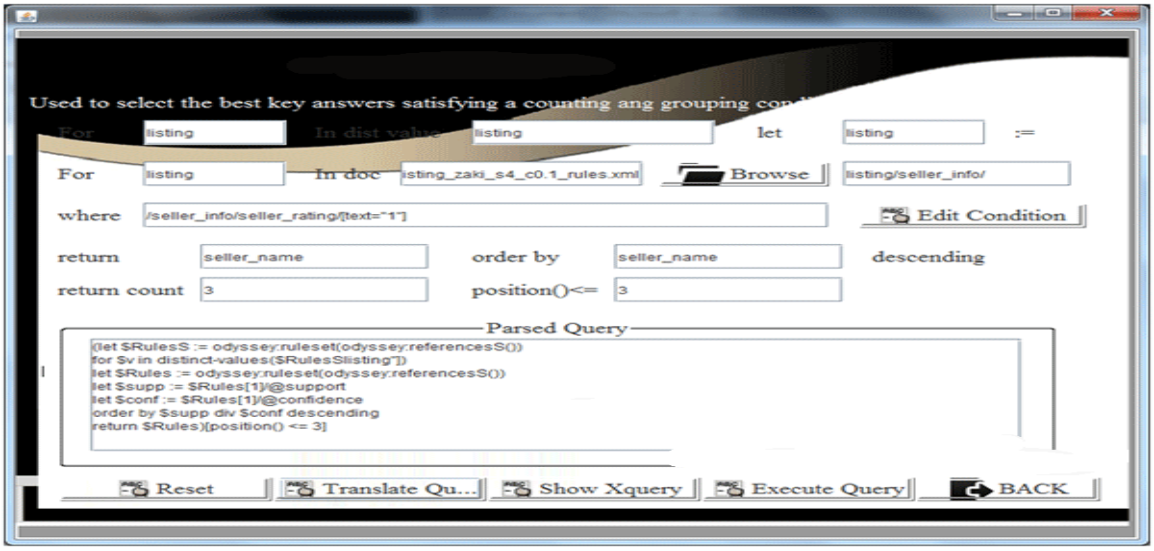


Fig.4:XML Query Processing using WHERE Clause

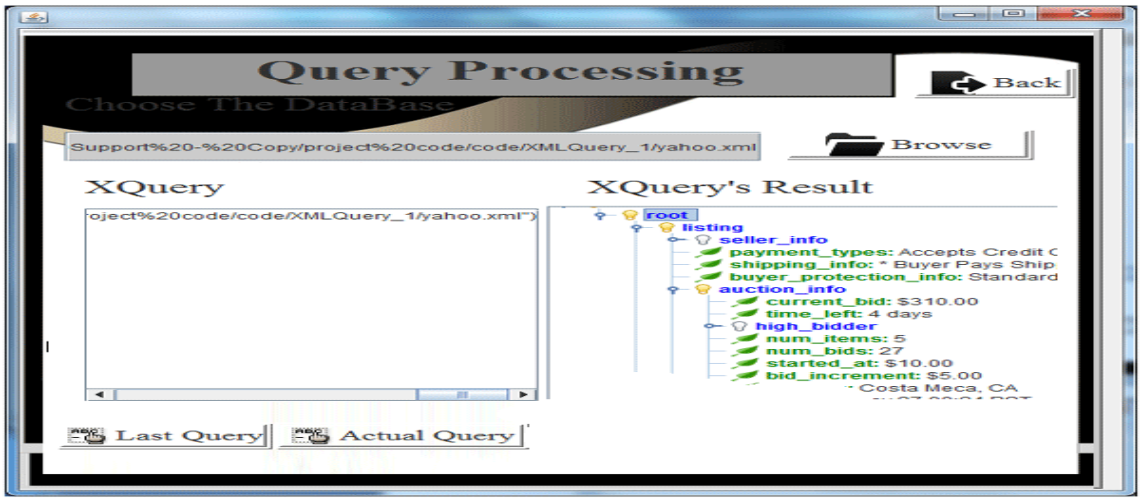


Fig.5: XQuery processingresults

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

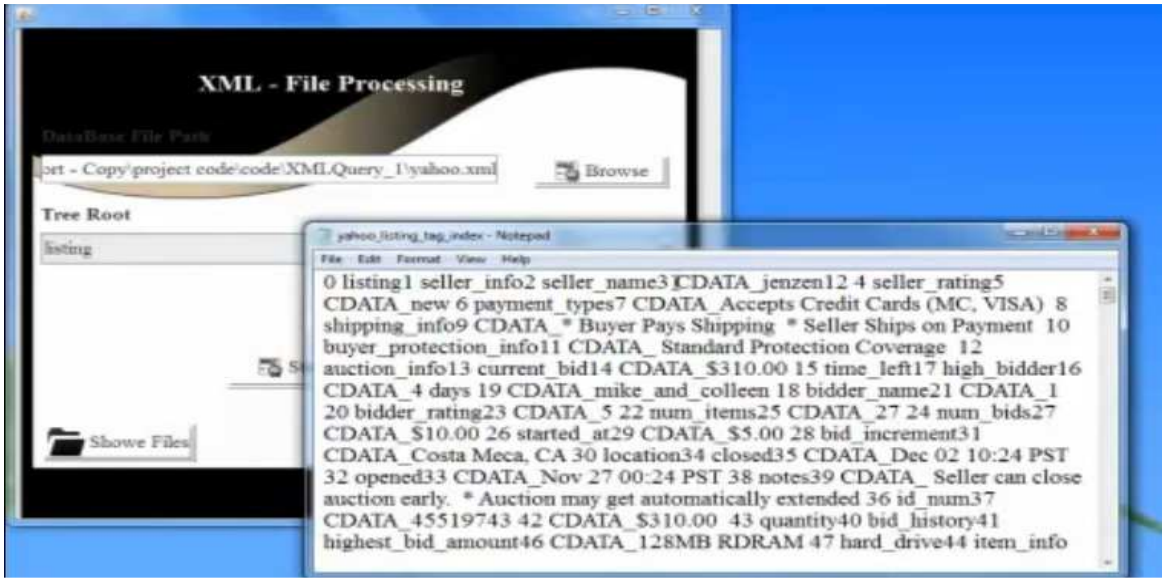


Fig.6:XML File Processing using Data File

5.1 Construction of Trees:

The XML documents have a flexible architecture. XML documents can be pre-processed. XMLpre-processing is done by XML parser. The DOM (Document Object Model) parser is used herewhich issued to construct the tree from the XML document. Before a XML document can be accessed, it mustbe loaded into an XML DOM object. The DOM creates a tree structure in the internal memory from thegiven XML document. It stores the all the document into memory before processing those documents. Itallows the users to traverse the document using top-down approach XML trees, access, insert, update the content, style and structure of thedocument 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 correctlywithout leaving any tag without its pair.

5.2 Extraction of Frequent Pattern:

Association rules describe the frequent occurrence of data items in a large amount of data collected. Xand Y are the two considered data items. Now they are represented in the form of X∪Y. HereAssociation 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 conditionalprobability of finding Y, having got X. The interesting patterns among the subtrees of the givenXML document can be identified. A Tree-based Association Rule (TAR) (Suganya, I.,*et al*., 2013), the form of XMLdocument defines pattern of subtrees in the XML document. The entire XML documentcan be accessed by providing the values of support and confidence. Mining TAR is a two stepprocess (Suganya, I.,*et al*., 2013).

This includes:

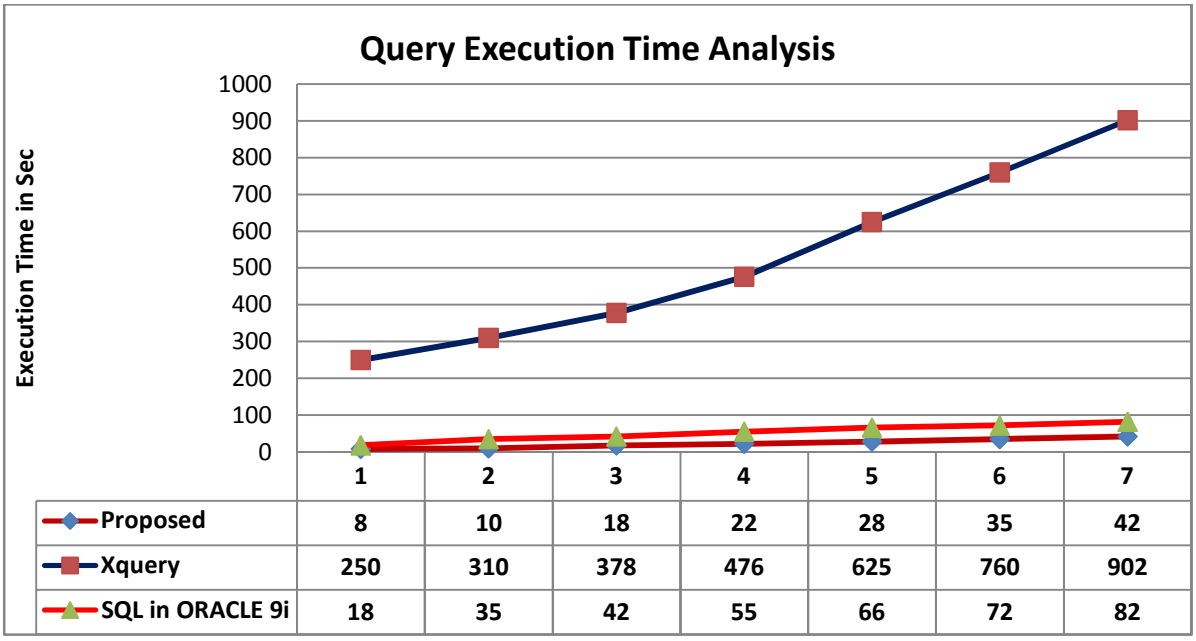
1. Mining frequent sub trees
2. Computing interesting rules

After obtaining the set of files the Proposed model will merge them to obtain one XML document which containsdata 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 modelwill give the most frequent feeds of all the files. After that the feed search will be performed by the Proposedmodel 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 SuiteRelease 2, to evaluate the different query results in thecase of a centralized database system. Here in this research used theVisual C++ language using BorlandC compiler (32 bit thatsupports up to 4 GB RAM) to implement our Proposedtechnique. It is used an Intel Processor with 2.13 GHz,4 GB of RAM under the Windows 7UltimateOperating system.

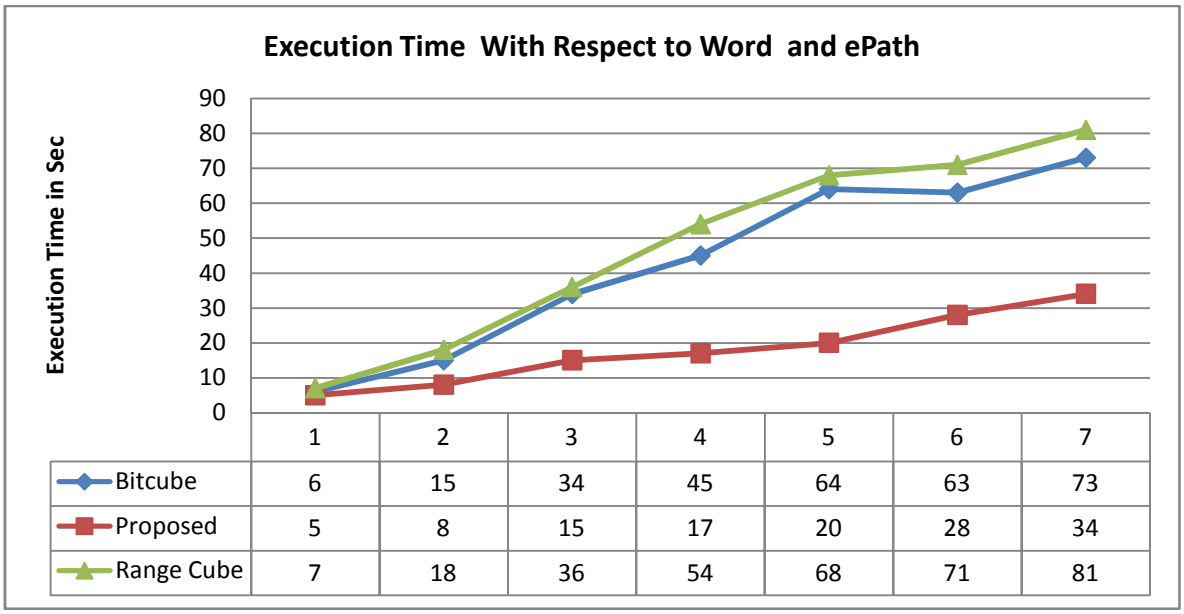
To support the Oracle 9i database is used the Windows 7 Ultimate operating system. Here in this work used the (well structured) XMLdatasets in (Mazuran, M.,*et al*., 2012) to run comparisons of the XQuerylanguage and our Proposed technique using filesizes of 2.5MB, 5.6MB, 10.8MB, 23.48MB and 110.35MB.



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

Theexperimental results for XQuery execution time,execution time in Oracle and proposed are presented inFigure 4, which for the purpose of clarity only depictsthe results for 110.35MB files, but which show comparativeperformance that is typical of all the tested file sizes.It can be seen that proposed method's execution times weresubstantially lower than those of XQuery, and also thatproposedmethod's outperforms the highly regarded Oracle executiontimes across the range of predicates tested. The resultsshow that the improvement achieved by Proposed methodincreaseswith the number of predicates.A second set of experiments was performed to comparethe query execution times of ProposedMethod, Range Cube and Bitcube (Paik, J.,*et al*., 2005).These experiments used 5, 10, 15, 20 and 25 elementpaths (ePaths) per document, and a variety of words perelement 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 ofclarity only depicts the results for 25 ePaths perdocument, and 20 words per ePath.



	1	2	3	4	5	6	7
Bitcube	6	15	34	45	64	63	73
Proposed	5	8	15	17	20	28	34
Range Cube	7	18	36	54	68	71	81

Fig.5: Execution time wrt word/ePath.

Implementation Details:

In particular, the idea of mining association rules toprovide summarized representations of XML documents hasbeen investigated in many proposals either by usinglanguages (e.g., XQuery) and techniques developed inthe XML context, or by implementing graph-or-tree-basedalgorithms. This work proposes to develop an application tomine frequently occurring subtrees from the generated XMLdata set by RSS Feed links of different newspaper links.Then storing them in another XML document. These rulesare extracted only if support and confidence of nodes isgreater than the provided threshold values. Then apply ourFrequent Pattern Miner algorithm for XML. The idea ofusing association rules as summarized representations ofXML documents was also introduced in where the XMLsummary is based on the extraction of rules both on thestructure (schema patterns) and on content (instancepatterns) of XML data sets. The limitations of this approachare: 1) the root of the rule is established a-priori and 2) thepatterns, used to describe general properties of the schemaapplying to all instances, are not mined, but derived as anabstraction of similar instance patterns and are less preciseand reliable.

A. Fundamental Concepts:

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

1. Mining frequent subtrees from the XML document;
2. Computing interesting rules from the previously minedfrequent subtrees.

Association rule (Agrawal, R. and R. Srikant, 1994) is an hint of the form A U B, where therule A and B are subset of the set C of element in a set oftransactions D and A ł B= Ø. A rule X Y states that thetransaction T that has the elements in A are likely to containalso the elements in B. Association rules are distinguishedby two survey: the support, which gives the percentage oftransactions present in D that contain both items A and B (AU B); the confidence, which calculates the percentage oftransactions in D containing the element A that also containthe elements B (support (A U B)/support (B)). In XMLcontext, both D and C are group of trees. In this work TARis generated using Natural Language Processing. The projectworks for RSS news links of multiple newspapers. RSSstands for "Really Simple Syndication". It is a way to easilydistribute a list of headlines, update notices, and sometimescontent to a wide number of people. It is used by computerprograms that organize those headlines and notices for easyreading. RSS works by having the website author maintain alist of notifications on their website in a standard way. Thislist of

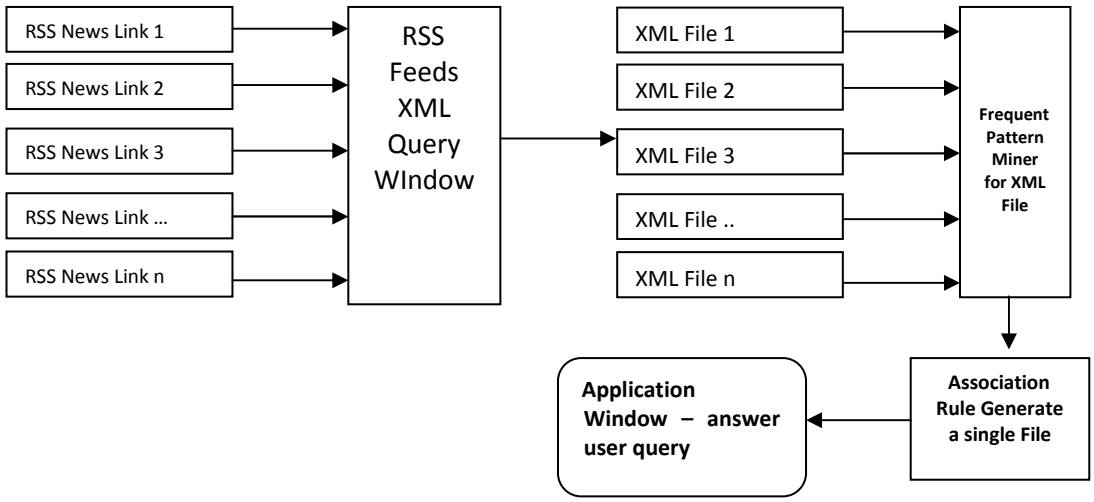


Fig. 6: Operation flow for frequent pattern Miner for XML.

notifications is called an "RSS Feed". People who areinterested in finding out the latest headlines or changes cancheck this list. Special computer programs called "RSSaggregators" have been developed that automatically accessthe RSS feeds of websites you care about on your behalf andorganize the results for you.

Producing an RSS feed is very simple and hundreds ofthousands of websites now provide this feature, includingmajor news organizations like the New York Times, theBBC, and Reuters, as well as many weblogs. RSS providesvery basic information to do its notification. It is made up ofa list of items presented in order from newest to oldest. Eachitem usually consists of a simple title describing the

itemalong 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 wantto read (such as the content of a weblog post) and sometimesit is just a summary.

B. Architecture Details:

This research work introduces a proposal for mining and storingModified Tree-Based Association Rules (MTARs) as a means torepresent intensional knowledge as an alternative, syntheticdata set would be queried for providing quick andsummarized answers. For calculating the MTAR we make useof Natural Language Processing. The algorithm for findingMTAR is given below:

Algorithm: Extracting Tree Based Association Rules fromXML Files

Input: RSS Feed Files R_n, Similarity Threshold T_h, SupportS, Confidence C.

Output: XML Tree Based Association Rules

1. Read all R_n by using XML Reader.
2. Extract Inner Element <Title> Tags within all $R_n$and insert them into a new XML File xf.
3. Initialize NodeListLst to contain Visited T_{sub} in x_f.
4. Initialize ItemSet, a Key Value Pair structure tocontain Element e and its support fs.
5. For Each SubtreeT_{sub} in X_f
6. Extract <Description> Tag I_{desc} from T_{sub}
7. If(Count(L_{st})==0)
8. Then Add I_{desc} into Lst
9. Else For Each element e_i in L_{st}
10. SimilarityScore = Compare(I_{desc}, e_i)
11. If(SimilarityScore>= T_h)
12. Then If ItemSet contains e_i
13. Then Increment support of $e_i$by 1 from ItemSet
14. Else Add e_i to ItemSet andincrement support of e_i by 2
15. Else Add e_i to ItemSet and incrementsupport of e_i by 1
16. Add I_{desc} into L_{st}
17. Using ItemSet as reference data structurecontaining Item e_i and its support f_s , desired support C and confidence C, generate frequentItemSet F.
18. For Each ItemSetIfr in F
19. Insert Ifr in XML Document XTAR
20. Return X_{tar}

Conclusion:

The Comparative Analysis of Implemented Approach with existing Document summarization and Information Retrieval Methods as mentioned below in the Table1;

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	No
Query – Specific Document Summarization	Statistical	Structured Only	No	No	No	No	No
A Language Independent Algorithm for Single and Multiple Document summarization	NLP	Structured Only	No	Yes	Yes	No	No
OLEDOT	NLP	Structured and Non structured data	Yes	Yes	Yes	No	Yes
Implemented Approach(Modified TAR)	NLP	Structured and Non structured data	Yes	Yes	Yes	Yes	Yes

The main goals of the work are to achieve:

1. Mine all frequent association rules restriction onthe structure and the content of the rules;
2. Store mined information in XML format;
3. Use the extracted knowledge to gain informationabout the original data sets;
4. Modified mined Tree based Association Rules when the original XML datasets change.

REFERENCES

Agrawal,R.and R. Srikant, 1994.“Fast Algorithms for MiningAssociation Rules in Large Databases,” Proc. 20th Int’l Conf. VeryLarge Data Bases, pp: 478-499.

Combi, C., B. Oliboni and R. Rossato, 2005.“Querying XMLDocuments by Using Association Rules,” Proc. 16th Int’l Conf.Database and Expert Systems Applications, pp: 1020-1024.

Paik, J., H.Y. Youn and U.M. Kim, 2005.“A New Method for MiningAssociation Rules from a

Collection of XML Documents," Proc. Int'lConf. Computational Science and Its Applications, pp: 936-945.

YogeshR.Rochlani, Prof. A.R. Itkikar, 2012. "Integrating HeterogeneousData Sources Using XML Mediator", ijcsn, 1: 3.

ArundhatiBirari, Prof. RanjitGawande, 2013. "Mining Tree-BasedAssociation Rules for XML Query Answering", ijettcsvol., 2: 3.

Wan,J.W.W.and G. Dobbie, 2003."Extracting Association Rules fromXML Documents Using XQuery," Proc. Fifth ACM Int'l WorkshopWeb Information and Data Management, pp: 94-97.

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.

Suganya,I., N.Velmurugan, Dr.P.Ganeshkumar, 2013. "XML Query-Answering Support System using AssociationMining Technique" IEEE conference on ICT.

MarouaneHachicha and Je´ roˆmeDarmont, Member, 2013.IEEE Computer Society, "A Survey of XML TreePatterns", IEEE 25.

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.

Wang Lian, Nikos Mamoulis, David Wai-lokCheung, 2005. "Indexing Useful Structural Patterns for XML QueryProcessing", Ieee Transactions On Knowledge And Data Engineering,17: 7.

NeoklisPolyzotis,MinosGarofalakis,YannisIoannidis, 2003. "Approximate XML Query Answers",Intl. Conf. onVery Large Data Bases.

Devi Mahalakshmi, S., Dr. K. Vijayalakshmi,Dr. K. Muneeswaran, G.Priyanka, 2010."Mining IntensionalInformation for answering XML-Queries using Tree-based Association Rules Approach", IEEE.

Chandra Sekhar.K, Dhanasree, 2012. "Extracting TARs from XML for Efficient Query Answering",InternationalJournal of Computer Science and Network (IJCSN) 1: 6.

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.

Borgwardt, K.M., H.P. Kriegel, P. Wackersreuther, 2006."Pattern Mining in Frequent Dynamic Subgraphs,"Proc. ICDM.

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.

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

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.

Mazuran, M., E. Quintarelli and L.Tanca, 2012."Data Mining for XML Query-Answering Support," IEEETans. Knowledge and Data Eng.,17(8): 1021-1035.

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.

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.

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.

Zaki, M.J., 2005. Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications, IEEE Trans. Knowledge and Data Eng., 17(8): 1021-1035.

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.

Termier, A., M.Rousset, M.Sebag, K.Ohara, T.Washioand H.Motoda, 2008.DryadeParent, an Efficient and Robust Closed Attribute Tree Mining Algorithm ,IEEE Trans. Knowledge and Data Eng,20(3): 300-320, Mar.