International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:11, 2014

A Tree Based Association Rule Approach for XML Data with Semantic Integration

D. Sasikala, K. Premalatha

***Abstract*—**The use of eXtensible Markup Language (XML) in web, business and scientific databases lead to the development of methods, techniques and systems to manage and analyze XML data. Semi-structured documents suffer due to its heterogeneity and dimensionality. XML structure and content mining represent convergence for research in semi-structured data and text mining. As the information available on the internet grows drastically, extracting knowledge from XML documents becomes a harder task. Certainly, documents are often so large that the data set returned as answer to a query may also be very big to convey the required information. To improve the query answering, a Semantic Tree Based Association Rule (STAR) mining method is proposed. This method provides intentional information by considering the structure, content and the semantics of the content. The method is applied on Reuter's dataset and the results show that the proposed method outperforms well.

***Keywords*—**Semi--structured Document, Tree based Association Rule (TAR), Semantic Association Rule Mining.

I. Introduction

SEMI structured data is a structured data format which does not follow strict data models like relational databases. The logical structure showcases document construction and not content. World Wide Web (WWW) information sources need semi structured data. For example e-mails, product catalog information for on-line purchases, news, articles, TV programs listings, journals and programming language codes use semi structured format [1]. Email documents and product catalog use implicit structure without any need to disclose explicit information. Some documents need specific information and use object oriented technologies.

Semi structured data uses simple tags or markers to describe document's logical structure by representing elements and maintaining their hierarchy. For example, Hyper Text Markup Language (HTML) and XML tags design semi structured documents [2]. XML is a mark-up language which is suitable for representing semi-structured data, which are often referred to as self describing that is no pre-imposed schema or type is needed for data interpretation itself. XML was designed to transport and store data. XML is now as important for the Web as HTML was to the foundation of the Web.XML is the most common tool for data transmissions between all sorts of applications. It is used in many aspects of web development, often to simplify data storage and sharing. XML data is stored in plain text format. This provides software and hardware-independent way of storing data. This makes it much easier to create data that can be shared by different applications.

XML documents form a tree structure that starts at the root and branches to the leaves. XML documents must contain a root element. This element is the parent of all other elements. The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree. The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings. All elements can have text content and attributes. A simple XML document that describes a person's contact details is given in Fig. 1.

Fig. 1 Example of an XML document and tree structure

Generally XML documents can be represented in two types of views: document-centric and data-centric views. In document-centric view, XML documents have irregular structure and it is interpreted by a document markup. In data-centric XML, documents use a regular structure, designed in standard format for semi structured data exchange and representation [3]. Hence, XML document suits mining tasks like classification, clustering, concept/entity extraction and a corpus' document summarization.

XML document access has two approaches namely keyword-based search and query-answering. The keyword–based search is from information retrieval tradition, where searches are performed on document's textual content; meaning there is no advantage from semantics conveyed through document structure. In query-answering system the query language for semi structured data rely on document structure to convey semantics. To form queries effectively, users need to know structure in advance, which is not the case usually. In fact, it is not necessary for an XML document to have a defined schema: 50% web documents do not possess this [4]. When user queries are specified without knowing document structure, they fail to retrieve information available. They retrieve information under a different structure.

Information retrieval has to match relevant texts with given query. Sometimes a part of document has enough information

D. Sasikala is working as Associate Professor in the Department of CSE at Bannari Amman Institute of Technology, India (e-mail: dskramesh04@gmail.com).
K. Premalatha is working as Professor in the Department of CSE, Bannari Amman Institute of Technology.

for the user's query and selecting the appropriate parts is useful when documents are long [5]. Document fragmenting and segmenting are the two techniques generally used to select the relevant portions of the document in [15]. The structure of an XML document is a tree, mining XML association rules is different from that in the traditional well-structured world. A transaction in XML context is XML fragment that define the context in which the items must be counted. In other words, the transaction is a sub tree, and the items are the leaf nodes in the sub tree; the root node of sub tree will be used to identify a transaction and use leaf node in the sub tree to identify an item. Association rule is a method for deriving intentional knowledge from XML documents in the form of TARs, and then storing these TARs as an alternative, synthetic data set to be queried for providing quick and summarized answers. This procedure is characterized by the following key aspects.

- It works directly on the XML documents, without transforming the data into any intermediate format
- It looks for general association rules, without the need to impose what should be contained in the antecedent and consequent of the rule.
- It stores association rules in XML format.
- It translates the queries on the original data set into queries on the TARs set.

II. Literature Survey

There are many methods available to perform Tree based Association Rule Mining. This section presents the literature review of some methods of it. In [6] they described a tool for extracting XML association rules from XML documents. XML enables the self-description of hierarchies of semi-structured information, by intermixing data content with semantic tags which describe such content. At the moment, the use of XML within the data mining community is still quite limited. There are some proposals to exploit the XML syntax to express interfaces to knowledge representation artifacts, including association rules, so that they can be easily changed among data mining tools.

Another method called Tree-based Association Rules (TARs) is proposed by [7]. Here mined rules provide required information on structure and content of XML file and the TARs are also stored in XML format. The mined knowledge is used later for XML query answering support. This enables quick and accurate answering. The prototype application is also developed to demonstrate the efficiency of the system. The empirical results are very positive and query answering is expected to be useful in real time applications. The focus is also on mining XML-enabled association rules with the use of templates. It is proposed by [8].

Qui in [9] analyzed various similarity measures, measurement methods and the cost of measuring. Searching the web based XML was similar to information search and mining of XML documents was done by expanding query and assigning weights to the terms given in the query. Experiments proved that the algorithm to extend the XML query for retrieving a document improved the full-search ratio and accurate search ratio.

The method in [10] described about mining association Rules from structured XML data. The XML data is different from relational data in term of structure. Relational data is flat and have a regular structure while XML structure vary and consist of some user defined tags. In order to mine XML data, a data preparation step should take place. Data preparation depends on the nature of the XML document and the type of transformation need to be done on the XML document in order to access and to mine it. Mining XML can be categorized into two parts.

- Indirect mining which means pre-processing of the XML documents.
- Direct mining which means mine the XML file without pre-processing or post processing.

A Tree-based Association Rules (TARs) mined rules based approach that provided approximate, intentional information on both XML documents structure and contents and could be stored in XML format was described in [11].

The method in [12] is used the mined knowledge for XML query answering support enabling accurate and quick answering. The aim of the method is to provide intentional knowledge as a substitute for original document during querying and not improve queries execution time over original XML data set. The suggested method can be developed to optimize mining algorithms.

Tree based query answering prototype for XML documents was introduced by [13] which reduced documents processing. Extracted sub trees were stored in XML format. It would locate patterns to describe schema's general properties applying to all instances not mined but derived as a similar instance patterns abstraction and were thus less precise and reliable. Once extracted, tree based association rules can be stored in a document and accessed independent of data set they were extracted from.

An approach based on TARs: mined rules to ensure approximate, intentional information on both XML documents structure and contents which could be stored in an XML format was described by [10]. This knowledge later provided a concise idea on structure and content and it also provides quick, approximate answers to queries. Reference [16] represents a survey about the various TARs algorithms.

III. Semantic Tree Based Association Rule Mining

Extracting information from semi structured databases becomes harder. This is because of the reason that the data and the content are inherently flexible. XML is a flexible hierarchical model suitable to represent large amount of data. Discovering frequent patterns from XML document provides implicit knowledge about the document. The tree based association rule mining helps to identify this implicit knowledge. TAR provides relevant information for a given query effectively. To provide the relevant information efficiently the content of the document is also taken in the proposed method along with the structure. The STAR retrieves the relevant document for the given query by considering the structure and the semantics of the XML document. The proposed method follows the steps below.

1. Pre-processing is done on the dataset by remove stop words and performs stemming.
2. Then the TF-IDF is computed.
3. The TAR is constructed for frequent terms based on semantic appropriateness between the terms and the minimum support.
4. Finally the performance is measured with different queries.

Pre-processing is done by removing unwanted stop words from documents and stemming replaces each word with its root. Then features are extracted by measure of term frequency, and a TAR is constructed. The proposed Semantic Tree based Association Rule (STAR) method integrates semantic graphs into TAR mining.

A. Pre-processing

A word is an uninterrupted sequence composed of letters (a..z), digits (0..9) and special characters (@ and _). A phrase has multiple words. For example the phrase "la machine IBM-360" counts as four words. Usually space and punctuation symbols delimit words. Stop words are non-significant words and usually are removed before starting document indexing. As stop words have no purpose in information retrieval, a general stop words list is framed for documents. There are 2 reasons to use stop words list [14].

For best information retrieval, matching query and document should be based on good indexing terms. For example document retrieval by using words like 'the' or 'be' is not a good retrieval strategy. These non-significant words are considered noise and reduce performance of relevant documents retrieval.

Stop words list reduces inverted file size from 30% to 50 %. Most search engines do not account for stop words during search to increase speed and reduce space. In many European and Italian languages variant words are formed by adding suffices at end of root word. Stemming removes inflectional and derivational suffixes to conflate word variants into same stem or root. For example, words like "thinking", "thinkers" or "thinks" can be reduced to "think". Indexing words with same route into single index term can increase document retrieval and classification success.

B. Term Frequency

Let D be set of documents and t be set of terms. Terms are selected after removing stop words and finishing stemming. Term Frequent – Inverse Document Frequency (tf-idf) denotes importance of term t_j in document d, in list of documents D.

$$tf - idf(i,d) = tf(i,d) * idf(i) \quad (1)$$

$$idf(i) = \log \frac{N}{df(i)} \quad (2)$$

where, in (1) tf() is frequency of term t_i in the document d. In (2) N is total number of documents and df(i) is number of documents containing the term t_i. The tf-idf evaluates the importance of a term to a document in a collection of documents.

C. Semantic Tree-Based Association Rules (STARs)

Association Rule Mining is a popular method to discover interesting relationship between data items in large databases. It is used to identify the strong association rules from large databases using the various interesting measures. Association rules are implications of the form $X \rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \Phi$. X is antecedent and Y is consequent. The strength of an association rule is mainly quantified by the measures. The following are the basic measures of association rule mining.

The rule has support s if s% of the transactions contain both X and Y. The interestingness issue refers to finding rules that are interesting and useful to users. A rule has confidence c if c% of the transactions that contain X also contain Y. A rule is said to hold on a dataset if the confidence of the rule is greater than a user-specified threshold. This can represented in (3) and (4) as

$$Support\,(X \rightarrow Y) = P(X \cup Y) \quad (3)$$

$$Confidence\,(X \rightarrow Y) = P(Y/X) \quad (4)$$

Tree based association rules (TARs) describes data items co-occurrences from collected data in the form $A \rightarrow B$, where A and B are arbitrary data item sets like $A \cap B = \emptyset$. Support and confidence measure association rules quality. Support is calculated as frequency of set $A \cup B$ in data set whereas confidence is calculated as probability of finding B having found A. Confidence is given by supp $(A \cup B)$ / supp(A).

Mining STAR follows the first step as mining frequent subtrees, i.e. sub trees with the semantic relationship and a support above a user-defined threshold, from XML document; and second step as computing interesting rules, i.e. rules with a confidence above user-defined threshold, from frequent sub trees. When mining process was finished and frequent STARs were extracted. Then the query is given to mine the information for the XML dataset. One of the reasons to use STAR instead of the original document is that processing for query answering is quicker than processing the document.

The algorithm represents extension to a generic frequent subtree mining algorithm as in [10] to compute interesting STAR. Algorithm inputs are XML document D, threshold for support of frequent subtrees minsupp, and threshold for confidence of rules, minconf. The algorithm is as follows.

```
Get-Interesting-Rules (D, minsupp, minconf)
// Finding frequent subtrees
FS = FindFrequentSubtrees (D, minsupp)
ruleSet =∅;
for all s ∈ FS do
// rules computed from s
tempSet = Compute-Rules(s; minconf)
// To compute all rules
ruleSet =ruleSet ∪ tempSet
end for
return ruleSet
Compute-Rules (s; minconf)
ruleSet =∅; blackList =∅ ;
for all c_s, subtrees of s do
if c_s is not a subtree of any element in blackList then
```

conf = supp(s) / supp(c_s)
if conf ≥minconf then
newRule= (c_s, s, *conf, supp(s)*)
ruleSet=ruleSet ∪{newRule}
else
blackList =blackList ∪c_s
end if
end if
end for

In the proposed method, the XML data contained in the body is used to devise suitable notions of semantic features and semantic relatedness among XML. The maximal frequent rooted unordered trees are used to identify associations. The semantic relationship is identified using (5):

$$SemanticVa\ lue = \frac{R^m}{1.1^d} \tag{5}$$

Here R is the tf-idf of mth term and d is the depth of the node from the root. The proposed method integrates semantic graphs into tree based association rule mining. An example is shown in Fig. 2 for the root Acquisition. The semantic related terms with respect to the word acquisition is used to build the tree. Each term is further defined based on the semantics. The tree formed thus improves the overall XML query structure for finding the related documents.

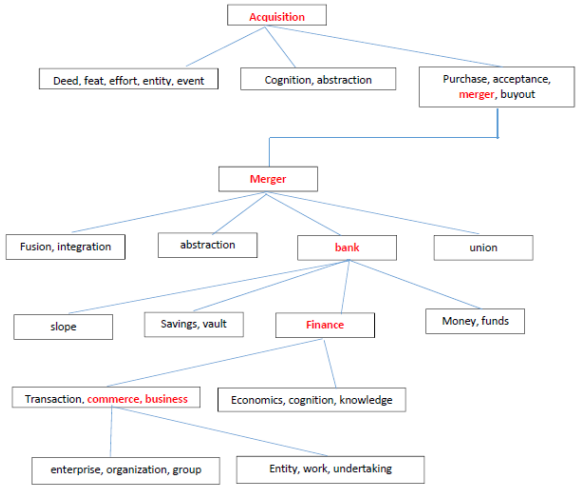

Fig. 2 An example of tree formed based on semantics

As the terms are selected from the body of the XML data, the proposed method mine intentional knowledge from XML datasets using TARs and have successfully resulted in good rules. Mined STARs get a concise idea of structure and XML document content based on semantics.

IV. Experiments and Results

To measure the performance of the proposed method the measures precision and recall is used. Precision and recall are the commonly used measures, to evaluate the accuracy of an approach. These methods are used here to evaluate the accuracy of approach which returns approximate answers.

$$recall = \frac{TP}{TP + FP} \tag{6}$$

$$precision = \frac{TP}{TP + FN} \tag{7}$$

In (6) and (7) TP (True Positive) = Number of correct predictions that an instance is valid; FP (False Positive) = Number of incorrect predictions that an instance is valid; FN (False Negative) = Number of incorrect predictions that an instance is invalid.

Recall depends on the support threshold of the mining process. In fact, the application of this algorithm returns the frequent subtrees and the number of such trees depends on the support threshold. Thus, since the minimum support threshold strongly influences query recall, it is a relevant parameter for tuning the intentional representation of information.

To understand how the support threshold influences the accuracy of the intentional answers, experiments were performed by extensionally querying some real data sets and also by extracting intentional answers from them. The experiments were conducted on the Reuters dataset. The Reuters data set with 4 class labels is used to evaluate the proposed method. It contains 21578 Reuters news documents. They were labeled manually by Reuters personnel. Labels belong to 5 different category classes, such as people, places and topics. The precision and recall calculated for 4 different queries such as Q1 is acquisition, Q2 is grain, Q3 is Crude oil and Q4 is earnings. The following diagrams show the results of these four queries.

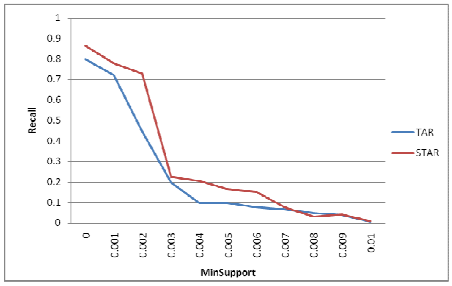

Fig. 2 Recall for Q1

Fig. 3 shows that the proposed method improved recall by 26.37% when compared with TAR. The results of the proposed method as shown in Figs. 3-6 give better recall value even when the support is low. So this shows that the proposed method outperforms well.

Fig. 4 shows that the proposed method improved recall by 18.72% when compared with TAR.

Fig. 5 shows that the proposed method improved recall by 26.23% when compared with TAR.

Fig. 6 shows that the proposed method improved recall by 27.26% when compared with TAR.

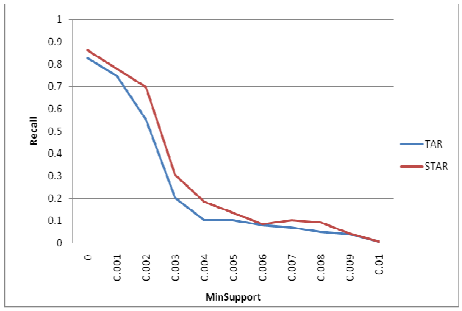

Fig. 3 Recall for Q2

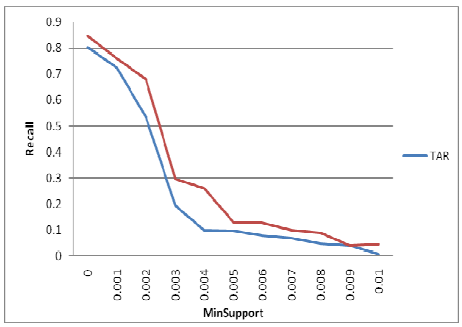

Fig. 4 Recall for Q3

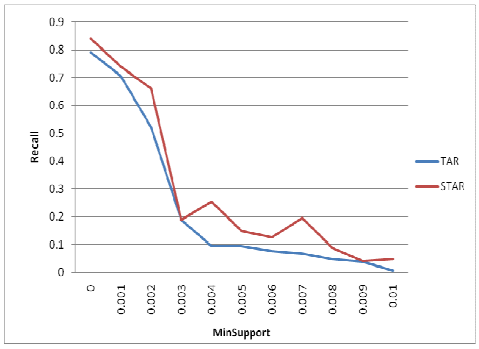

Fig. 5 Recall for Q4

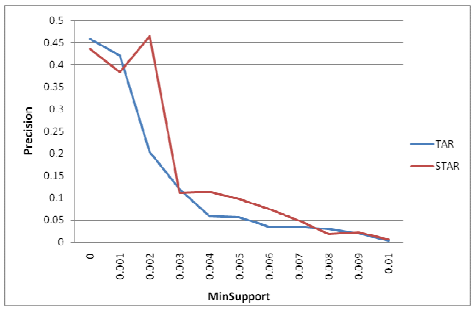

Fig. 6 Precision for Q1

Fig. 7 shows that the proposed method improved precision by 22.86% when compared with TAR. The results of the proposed method as shown in Figs. 7-10 give better Precision value even when the support is low. So this shows that the proposed method outperforms well. It also shows that the existing and proposed method gives the same result when the support is high.

From Fig. 8, it shows that the proposed method decreased precision by 0.3% when compared with TAR.

From Fig. 9, it shows that the proposed method improved precision by 14.44% when compared with TAR.

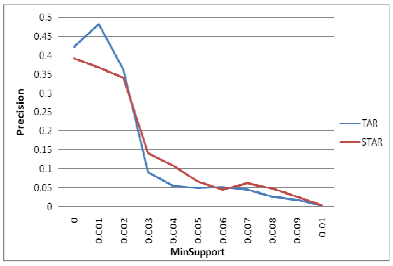

Fig. 7 Precision for Q2

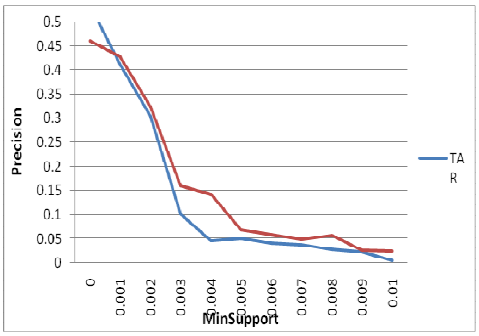

Fig. 8 Precision for Q3

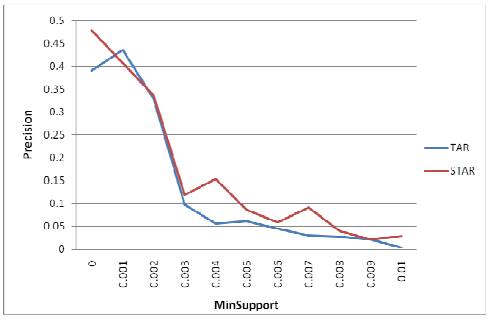

Fig. 9 Precision for Q4

Fig. 10 shows that the proposed method improved precision by 22.65% when compared with TAR. For a given query recall is defined as the number of relevant document correctly predicted to the total number of documents available. The algorithm can be tuned to increase the turnout of relevant documents. In the process, irrelevant documents are also predicted as relevant documents. Therefore as recall increases, precision decreases and vice versa. In the proposed technique the balance between precision and recall is well maintained.

V. Conclusion

XML document's rich and flexible format are attracted by using semi structured contents in many applications. The

proposed STAR method investigates integrating semantic graphs into tree based association rule mining by devising suitable notions of semantic features and semantic relatedness among XML. The semantic related terms with respect to the term is used to build the tree and identify associations. The tree formed thus improves the overall XML query structure for finding the related documents. The Reuters dataset has been taken for validating this method. The results show that the proposed method outperforms well with the help of precision and recall measures.

REFERENCES

[1] Markus Tresch, Neal Palmer and Allen Luniewski (1995), "Type Classification of Semi Structured documents", In the proceedings of the 21st International Conference on Very Large Data Bases,pp.263-274.

[2] Jeonghee Yi and Neel Sundaresan, (2001), "A Classifier for semi structured documents", In the Proceedings of 6th Internatnal Conference on Knowledge Discovery and Data Mining,pp.34-344

[3] Shashirekha H.L., Vanishree K.S., and Sumangala N,(2011),"Content and Structure Based Classification of Xml Documents", International Journal Of Machine Intelligence , Vol. 3, No. 4, pp.376-380.

[4] Sekhar, G. S., and Krishna, S. M. , (2012), "Efficient Data Mining for XML Queries–Answering Support", In the IOSR Journal of Computer Engineering, Vol.4, No.6, pp. 13-22.

[5] F. Llopis A. Ferrandez , J. L. Vicedo and A. Gelbukh,(2002), "Text segmentation for efficient information retrieval", In the Proceedings of 3rd International Conference on Text Processing and Computational Linguistics,LNCS 2276: pp 373-380.

[6] Chen L, Bhowmick, SS, & Chia LT, (2004), "Mining association rules from structural deltas of historical XML documents", In the Proceedings of Pacific-Asia conference on knowledge discovery and data mining, pp. 452-457.

[7] AliMohammadzadeh R, Soltan S & Rahgozar M, (2006), 'Template guided association rule mining from XML documents', Proceedings of 15th International World Wide Web Conference, pp. 963-964

[8] Tekli J, Chbeir R, & Yetongnon K (2007), "Structural similarity evaluation between XML documents and DTDs", Proceedings of the 8th International Conference on Web Information Systems Engineering Nancy, pp. 196-211.

[9] Qiu W (2009), "Research and application of XML documents query based on weight cost", Asia-Pacific Conference on information processing, vol.1, pp.525-528.

[10] Mazuran, M, Quintarelli, E, & Tanca, L 2012, 'Data mining for XML query-answering support', IEEE Transaction on Knowledge and Data Engineering, vol. 24, no. 8, pp. 1393-1407.

[11] Sekhar, G. S., and Krishna, S. M., (2012), "Efficient Data Mining for XML Queries–Answering Support",In the IOSR Journal of Computer Engineering, Vol.4, No.6, pp. 13-22.

[12] Vikhe, P. B., & Gunjal, B. L. (2013), "Extracting Tree Based Association Rules from XML Document", International Journal of Emerging Technology and Advanced Engineering,Vol. 3, No.6.

[13] Chiranjeevi, K., Vasantha, K., & Rao, C. M. (2013),"A Succinct Answering Prototype for XML Data", In the international journal of Advanced and Innovative Research,Vol.2,No.11, pp. 642-651.

[14] Jacques Savoy,(1999) "A Stemming Procedure and Stop word List for General French Corpora", In the Journal of the American Society for Information Science, Vol.50, No.10, pp 944-952.

[15] G. Salton, (1989), "Automatic text processing: the transformation, analysis, and retrieval of information by computer". Addison-Wesley Longman Publishing Co. Boston, MA, USA 1989.

[16] Vurukonda, N., Reddy, G. R., C., Mounika, B., Yogyatha, G., Srujana, N., & Priya, P. K. (2013). "A Survey on Tree based Association Rules (TARs) from XML Documents", In the International Journal of Research and computational Technology,Vol.5, No.1.