Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

Data Mining for XML Query Answering Support

Rohan Mane[1], Pooja Pawar[2], Amarjeet Kachawa[3], Akshada Tapkir[4]

Be Computer, G.H.Raisoni Institute Of Engineering & Technology,Computer Engineering, University Of Pune, India

Abstract- The database research field has concentrated on the Extensible Markup Language (XML) due to its flexible hierarchical nature which can use to represent huge amounts of data. XMLdoesn't have a fixed schema, while having a possibly irregular and incomplete structure.

In this recent world, we have seen digital information available on the web like e-business transaction, e -shopping, e-learning etc., These XML documents are enormous and so the datasets returned an answer query is also huge, which in turn is a complicated process for the retrieval of interpretable knowledge. In this paper, we use an enhanced Tree based Association Rule (TAR) to retrieve the results for the queries posed by the users on the XML document. TAR files provide intensional information on the structure and content of XML document. In addition we propose a method to dynamically update the TAR files when the dataset changes.

Keywords- XML, Query-answering, Data mining, Intensional information, Tree-based association rule.

I. INTRODUCTION

Nowadays web pages are commonly XML based and most of this information does not have well-defined schema. They are comparatively huge document and incomplete in practice. Therefore effective answering of the XML document is tedious as it leads to irrelevant and more generic results. Conventional systems are well good at storing data into databases safe and very fast but not good at providing meaningful analysis. Hence, to extract interesting knowledge from large amounts of data stored in databases data mining or knowledge discovery in database (KDD) is used. Data mining is used to extract the interesting knowledge from large amounts of data stored in databases or data warehouses. The mined knowledge can be represented in many different ways such as clusters, associations, classifications, decision trees, decision rules etc., In which, association rules is much more effective for discovering important relationships in great amounts of data. It is responsible to find correlation relationships among different data attributes in a large set of items in a database. Association rules can be mined from XML documents with three different methods. First method is to mine rules between XML document tags. Second is to mine association from the element content. Third method is to discover the interesting relationship between the XML documents which is otherwise known as frequent substructure.

In attempt to standardize the format of data exchanged over the web and to achieve interoperability between different technologies and tools involved, World Wide Web Consortium [2] introduced XML (Extensible Markup Language). There are two main approaches to access XML document: keyword-based search and query-answering. In this paper we use query-answering system to access XML documents. The users are supposed to post queries and this system is much efficient to make questions and retrieve answers respectively. Discovering recurrent patterns inside XML documents provide high-quality knowledge about the document content frequent patterns are in fact intensional information about the data, that they specify the document in terms of a set of properties rather than by means of data. In this paper, we describe a method for extracting interesting patterns using association rule mining from the XML documents. This give us the summarized views of the XML documents from which queries can be made over the extracted XML documents

II. REVIEW OF RELATED WORKS

There are number of researches available in the literature for XML document mining to efficiently store, index and search on XML data. In the recent years, several researchers are focused on mining based encoding the XML nodes. The following description gives some of the recent researches to mine information from the XML documents. Apriori [4] algorithm and AprioriTid [4] algorithm use multiple passes over data. Present in the databases for finding the frequent pattern from the data. The support value of each data is counted and the items were determined which have minimum support count given by the user. In succeeding passes, it proceeds with items passed in the previous passes. The items were stored into new large itemsets are called candidate item sets. This process continues until no large item sets were retrieved. XMINE RULE [3] operator is used for mining association rules with the relational data. This intends that, after dropping of unneeded data, XML document is converted into relational form. XQuery along with Apriori [1], extracts association rules and Query Answering system made by XQuery used only in simple XML documents. XML document can be mined using XQuery language without pre-processing or post-processing for association rules. XPATH [6], the most important component of XQuery was used in the field of mining association rules. It allows the user to specify the rules. It retrieves the answers for the queries made. Frequent subtree mining [6] technique was used in the discovery process which does not ignore the structure of the tree in the rules. The frequent subtree is split into several subtrees based on the value of support provided by the user and the tree mining algorithm that illustrates the depth of the frequent patterns to identify and branching factor were the key actors. It retrieves the embedded subtrees that sustain the relationships between the node pairs. But it does not distinguish the pairs.

III. EXISTING APPROACH

There is no existing approach has yet studied the problem of relevance oriented result ranking in depth. The search intention for a keyword based query is not easy to determine and can be ambiguous, because the search through condition is not unique; hence, to measure the confidence of each search intention candidate, and to rank the individual matches of all these candidates is a challenging task. Subsisting methods cannot resolve this ranking strategy to rank the individual matches challenge, thus it return low quality result in term of query relevance. 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.

IV. PROPOSED APPROACH

The proposed XML query answering support framework is to perform data mining on XML and obtain intentional knowledge. The intentional knowledge mined is also in the form of XML. This is nothing but rules with a support and confidence. In other words, the result of data that is mined is TARs (Tree-based Association Rules).

As can be seen in fig.1, the framework is to have data mining for XML query answering support. When XML file is given as input to the DOM parser, it will parse until it is well- formed and validated. If the XML document is valid, it is parsed and loaded into a DOM object which can be navigated easily. The parsed XML file is given to data mining sub system which is responsible for sub tree generation and also TAR extraction. The generated TARs are used by query processor sub system. This module takes XML query from the end user and makes use of mined knowledge to answer the query quickly.

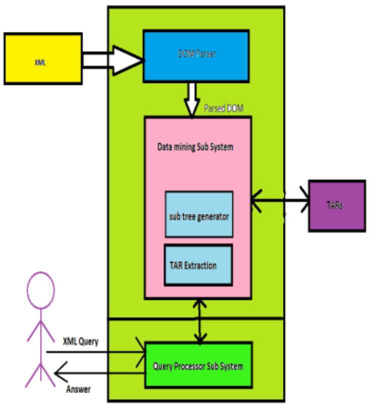

Fig 1: XML query answering support framework

a. TAR EXTRACTION

Association rules [1] describe the frequent occurrence of data items in a large amount of data collected. A and B are the two data items. They are represented in the form of A∩B. Association rule is measured by means of Support and Confidence. Support represents the frequency of the set (A and B) found in the data set. Confidence represents the conditional probability of finding B, having got A. The interesting patterns among the subtrees of the given XML document can be identified. TAR mining is a process composed of two steps:
1) Mining frequent sub trees, that is, sub trees with a support above a user defined threshold value, from the XML document; 2) Computing interesting rules, defines finding the interesting rules that are with user-defined confidence value. The frequent pattern of subtrees had been extracted into TAR files. The TAR files are stored in XML document. These TAR files contain the rules which are computed over confidence values. Each rule is saved inside the <rule> element. These files represent the intensional knowledge about the XML document. This process of mining TAR eases the exploitation of the query- answering system. TARs are mined by generating the rules with the more number of nodes in the body tree. This reduces the complexity.

Algorithm 1 Get-Interesting-Rules (D, minsupp, minconf**)**

```
1:// frequent subtrees
2:FS = FindFrequentSubtrees (D, minsupp)
3:ruleSet =∊ϕ
4:for all s  FS do
5:// rules computed from s
6:tempSet = Compute-Rules(s, minconf)
7:// all rules
8:ruleSet = ruleSet U tempSet
9:end for
10:      return ruleSet
```

Function 1 Compute-Rules (s, minconf**)**

```
1:ruleSet = ϕ ; blackList = ϕ
2:for all cs, subtrees of s do
3:if cs not a subtree of any element in blackList then
4:conf = supp(s) / supp(cs)
5:if conf ≥ minconf then
6:newRule = {cs, s, conf, supp(s)}
7:ruleSet = ruleSet U {newRule}
8:else
```

```
9:blackList = blackList U cs
10:      end if
11:      end if
12:      end for
13:      return ruleSet
```

The rules obtained from [1] Algorithm 1 is written to an XML file. Then indexing is done to the XML file. Afterwards when XML queries are prepared, the proposed system uses index and TARs to quickly answer the query. An index [3] is created for the extracted TAR file to make fast access of the document when queries are posted. This index file is also created in XML format. This contains the set of trees and every node in the tree contains references to the rules generated. The query will be made to the original XML document. This will be automatically translated into TAR files. By using the translated TAR files, the XML documents can be queried easily compared to the other operators [3]. This is because the XQuery, the XML query language which is specifically designed for XML documents. This consists of three class queries [1] to be transformed.

They are as follows:

Class 1: σ/π- queries: This query is used to bring down simple and complex operators with restrictions on them.

Class 2: count-queries: This query is used to count the number of elements having a specific data mentioned in the query.

Class 3: top-k queries: This query is used to select the top k queries which satisfy a grouping condition. By using these class queries, the users can pose a query over the XML document.

b. MODIFICATIONS OF XML DOCUMENT

The answers were retrieved for the given XQuery over the entire XML document which gets transformed to TAR files. When the original XML document gets modified, the TAR file has been updated by using XPATH. This is used to update the XML files dynamically instead of creating the new TAR file for the modified XML document. This also updates the index files already created. This simplifies the work when the input XML document changes at any time. Also by using with XML DOM parser the TAR files and its index can be updated.

V. TREE RULER PROTOTYPE

TreeRuler [1] is a tool used in our approach. When the XML document is given, it makes users to retrieve the intensional information for the queries. Users formulate XQueries over the original data, and these queries are automatically translated and executed on the intensional knowledge. Get the Idea that allows intensional information extraction from an XML document, when given the supports, confidence and the files where the extracted TARs and their index are to be stored.
The tree ruler interface offers three tabs : Get the Idea: this allows showing the intensional information as well as the original document, to give users the possibilities to compare the two kinds of information. Get the Answers: it allows querying the intensional knowledge and the original xml document

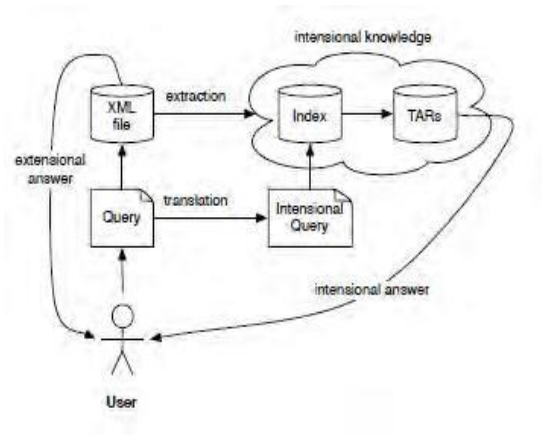

Fig 2.Tree ruler architecture

Users have to write an extensional query. When the query belongs to the classes have analyzed, then it is translated and applied to the intensional knowledge. Finally, once the query is executed, the TARs that reflect the search criteria are shown in Fig.2

VI. CONCLUSION AND FUTURE WORK

In this paper we proposed a method for extracting TAR (Tree-based Association Rule) files from the original XML document which was given as input. This TAR files were formed by the frequent pattern of XML document. This TAR files were used by the query-answering system for retrieving the answers from the XML document. The query-answering system is used in the fields of information representation and reasoning, multimedia, sentimental analysis, information retrieval and natural language processing. In the future we will enhance the process of mining TAR files and also use link to update XML documents when original dataset changes.

REFERENCES

[1] Mirjana Mazuran, Elisa Quintarelli, and Letizia tanca. Optimized Data Mining for XML query-answering support. IEEE Transactions on Knowledge Data Engineering, Volume:PP Issue:99, 2011
[2] World Wide Web Consortium, Extensible Markup Language(XML) 1.0, http://www.w3C.org/TR/REC-xml/, 1998
[3] D. Braga, A. Campi, S. Ceri, M.Klemettinen, and P. Lanzi, "Discovering of Interesting Information in XML Data with Association Rules," Proc.
[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Proc. of the 20th Int. Conf. on Very Large Data Bases. Morgan Kaufmann Publishers Inc., 1994.
[5] J.W.W. Wan and G.Dobbie, "Mining Association Rules from XML Data using XQuery," Proc. Fifth ACM Int Workshop Web Information and Data Management, pp.95-97, 2003.
[6] Liang Huai Yang Mong Li Lee Wynne Hsu Sumit Acharya, "Mining Frequent Query Patterns from XML Queries,"