Volume : 3 | Issue : 1 | Jan 2014 • ISSN No 2277 - 8160

Research Paper | Engineering

Data Mining For XML Query-Answering Support

Prof. Dr. V.D. Kulkarni	Department of Computer Engineering, University of Pune, India
Bansi Kaneria	Department of Computer Engineering, University of Pune, India
Sayali Kothawade	Department of Computer Engineering, University of Pune, India
Tunoor Rao	Department of Computer Engineering, University of Pune, India
Vishakha Agrawal	Department of Computer Engineering, University of Pune, India

ABSTRACT

Extracting information from semi structured documents is a very difficult task, and is going to become more and more critical as the amount of digital information available on the net grows. Documents are often so large that the data set returned as answer to a query may be too big to convey interpretable knowledge. In this paper, we describe an approach based on Tree-Based Association Rules (TARs): mined rules, which provide approximate, intentional information on both the structure and the contents of Extensible Markup Language (XML) documents, and can be stored in XML format as well. This mined knowledge is later used to provide:

1) a concise idea and the gist of both the structure and the content of the XML document

2) quick, approximate answers to queries.

In this paper, we focus on the second feature. In recent years, the database research field has concentrated on the Extensible Markup Language (XML) as a flexible hierarchical model suitable to represent huge amounts of data with no absolute and fixed schema, and a possibly irregular and incomplete structure. There are two main approaches to XML document access: keyword-based search and query-answering. The first one comes from the convention of information retrieval , where most searches are performed on the textual content of the document; this means that no advantage is derived from the semantics conveyed b content of the document; this means that no advantage is derived from the semantics conveyed by the document structure.

KEYWORDS : DataMining, XML ,XQuery ,support, confidence, association rules, iTars, sTars

I. INTRODUCTION

In brief, the project addresses the need of getting the gist of the document before querying it, both in terms of content and structure. Discovering 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 attributes rather than by means of data.

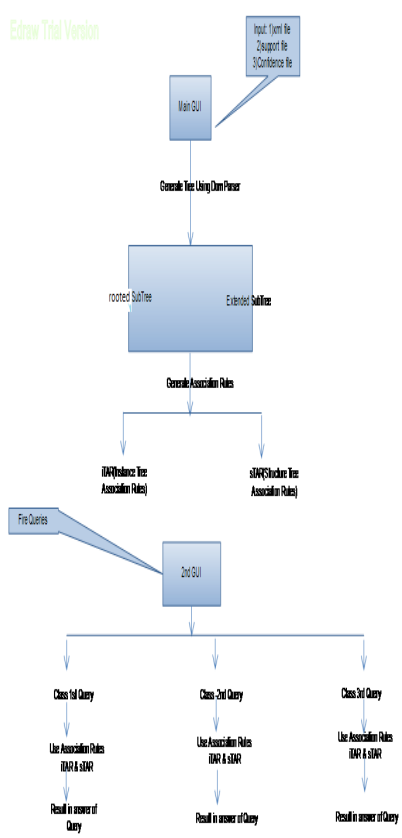
In this system, we propose a data mining for XML document. This can be used on any operating system. We don't need to install any external plug-ins or software.

In this, we are using XML as a database for the data mining. It is stored in the form of tree structure which is easy to understand.

II.RELATED WORK

Previously, people used Oracle, DB2 which is the database with software installation and memory consumption . These Databases have their query language to do operation with them. Thus it lead to the problem of how to perform CRUD operations with these different Databases. Also, earlier data mining was done on the basis of either keyword based search or query answering and both these methods resulted in problems like information overloading and information deprivation.

III. SYSTEM ARCHITECTURE

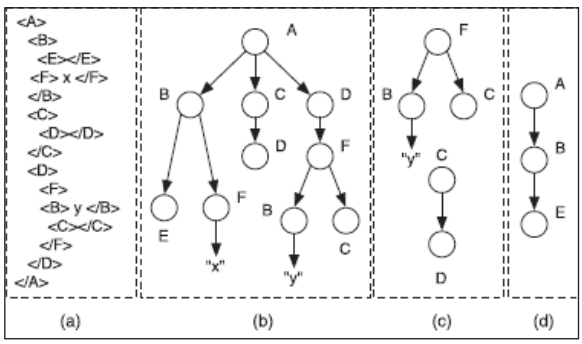


IV.IMPLEMENTATION DETAILS

The various algorithms used in the project are:

iTAR (instance TARs):- They provide information on the structure of the document as well as the data values contained in them.

sTAR(structured TARs) **:-** They provide information only on the structure of the XML document.



(a) An example of XML document.
(b) Its tree-based representation.
(c) Two induced subtrees.
(d) A rooted subtree.

Support:- Support is an indication of how frequently the item appears in the database.

Confidence :- : it indicates the no. of time the If/then statements have been found to be true.

Appriory:- *It* is an algorithm used in data mining for learning association rules.

It is used or works with any XML document as long as the structure of the document is known in advance

V. SYSTEM REQUIREMENTS

A system can be characterized by its functional and non-functional requirements. Functional requirements describe the functionality of a system while non-functional requirements describe attributes like reliability, maintainability and security, etc.

The system's functional requirements are as follows:

1. Software Interface:-

Front End: JAVA.

Back End: MySQL

The system's non-functional requirements are as follows:

Performance Requirement:-

1)Availability:-Our software will be available for almost all kind of data mining projects.

2) Speed:- The speed of our system depends upon the processor as well as the XML document size and the support and confidence given.

Software Quality Attributes:-

1) Reliability:- The project cannot be hampered under any circumstances. The sudden loss of power would not affect the software to such an extent that cannot be recovered.

2) Availability:- The application developed by us will be available all the time to the users .

3)Portability:- This software is portable and adaptable to any system that fulfills the software requirement.

4)Performance:- In software engineering, performance testing is testing that is performed to determine how fast the software performs under a particular workload. The application problem is best at its level. It does not require any frequent refreshing. Once the software is deployed it can be used continuously thereafter, its platform independent and it can be used universally. The system can be restarted if hanged or sudden loss of power supply. **5)Security:-** We are providing the authentication for securing the application and the database.

6)User Friendliness:- This tool is user friendly. As a user there are no complicated steps to use this application.

7)Robustness :-

Minimum human intervention will be required.

VI. SYSTEM TESTING AND IMPLEMENTATION PLAN

Testing is a method to check the functionality of components, sub assemblies, assemblies and/or a finished product.

For successful implementation of our project, it is necessary that the user should first correctly do the login and then enter a valid XML document with support and confidence. Further on the basis of all the inputs various tree based association rules will b generated and then user can finally fire its queries on the form of XQuey.

1.Requirement Specification:-

It includes complete specification of the system (with appropriate assumptions). A document detailing the same should be written and a presentation on that must be made. Attempt should be made to survey more on the system and to check latest updates regarding system currently used for detection.

2.Technology familiarization:-

It includes understanding of the algorithms and technology needed to implement the project. The technology defined should be such that we are able to apply it practically to the project rather than just a theoretical perspective.

3. High-level and detailed Design:-

Listing down all the possible scenarios and then coming up with flowcharts or pseudo code to handle the scenario. For each requirement specified, a corresponding scenario should be there.

4.Implementation of the system:- Implementation of the desktop application and establishing connection with java n my sql for data miming. During this milestone period, it would be a good idea for the team (or one person from the team) to start working on a test plan for the entire system. This test-plan can be updated as and when new scenarios come to mind.

5. Unit Testing:-

The system should be thoroughly tested by running for all the scenarios. Some time should be there to handle any issues found during testing of the system.

6. Final Review:-

Issues found during the previous milestone are fixed and the system is ready for the final review. In the final review of the project, it should be checked that all the requirements specified during milestone are fulfilled (or appropriate reasons given for not fulfilling the same).

SYSTEM IMPLEMENTATION PLAN:

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Topic Finalization	16 days	Mon 6/10/13	Sat 6/29/13		Internet
2		Study of IEEE paper	6 days	Mon 7/1/13	Mon 7/8/13	1	IEEE paper
3		Synopsis submission	5 days	Tue 7/9/13	Mon 7/15/13	2	IEEE paper
4		Defining problem definition	5 days	Tue 7/16/13	Mon 7/22/13	3	IEEE paper
5		Discussion of references	3 days	Tue 7/23/13	Thu 7/25/13	4	Internet and IEEE p
6		Literature survey	11 days	Fri 7/26/13	Fri 8/9/13	4	Internet and IEEE p
7		Presentation1 preparation	6 days	Mon 8/12/13	Mon 8/19/13	4	IEEE paper
8		Presentation2 preparation	7 days	Fri 7/19/13	Sat 7/27/13		IEEE paper
9		Discussion of all reviews	6 days	Tue 8/20/13	Tue 8/27/13		IEEE paper
10		Feasibility study	5 days	Wed 8/28/13	Tue 9/3/13		sahani (DAA)
11		Satisfiablity study	7 days	Wed 9/4/13	Thu 9/12/13		sahani (DAA)
12		SRS- Introduction	2 days	Mon 7/29/13	Tue 7/30/13	8	Booch(OOMD)
13		SRS- System features	2 days	Tue 8/20/13	Wed 8/21/13	8	Booch(OOMD)
14		SRS- External interface	1 day	Tue 8/20/13	Tue 8/20/13	8	
15		SRS- Non-functional req	1 day	Tue 8/20/13	Tue 8/20/13	8	
16		SRS- Analysis models	3 days	Tue 8/20/13	Thu 8/22/13	8	
17		SRS- System plan	2 days	Tue 8/20/13	Wed 8/21/13	8	
18		Mathematical Modeling	7 days	Mon 9/2/13	Tue 9/10/13		
19		Architectural diagram	3 days	Wed 7/31/13	Fri 8/2/13	12	
20		UML Diagrams	10 days	Mon 8/5/13	Fri 8/16/13	19	Rom Patton
21		Design of test cases	6 days	Mon 8/19/13	Mon 8/26/13	20	
22		Assignments of project	6 days	Tue 7/23/13	Tue 7/30/13		
23		Reference Term-work Assessment Sheet 1 Work	9 days	Wed 9/25/13	Mon 10/7/13		
24		Report Preparation	11 days	Wed 9/18/13	Wed 10/2/13		Internet

Page 1

VII. CONCLUSION

The goals we have achieved in this paper are

1) mine all frequent association rules without imposing any restriction on the structure and the content of the rules;
2) Store mined information in XML format; 3) use the extracted knowledge to gain information about the original data sets.

REFERENCES

1. Y. Xiao, J.F. Yao, Z. Li, and M.H. Dunham, "Efficient Data Mining for Maximal Frequent Subtrees," Proc. IEEE Third Int'l Conf. Data Mining, pp. 379-386, 2003. | 2. A. Termier, M. Rousset, and M. Sebag, "Dryade: A New Approach for Discovering Closed Frequent Trees in Heterogeneous Tree Databases," Proc. IEEE Fourth Int'l Conf. Data Mining, pp. 543-546, 2004. | 3. K. Wong, J.X. Yu, and N. Tang, "Answering XML Queries Using Path-Based Indexes: A Survey," World Wide Web, vol. 9, no. 3, pp. 277-299, 2006. | 4. Akihiro Inokuchi, Takashi Washio and Hiroshi Motoda I. "An Apriori-based Algorithm for Mining Frequent Sub-structures from Graph" Data. Sept. 13-16, 2000, | 5. Ali Mohammadzadeh, M. Rahgozar, and A. Zarnani." Documents a New Model for Discovering XML Association Rules from XML." Date14 august 2006 |