Volume 3, Issue 12, December-2016, pp. 613-619 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Intelligent XML Query-Answering Support with Efficiently Updating XML Data in Data Mining

[1]Parag Zaware, [2]Prabhudev.I

[1, 2] Vishwabharati Academy College of Engineering, Ahmednagar

Email ID- zaware.parag@gmail.com ,irabashetti@gmail.com

***Abstract*—** Data is present in various unstructured format. Extracting information from non structured documents is a very difficult task and it is become more and more critical when the amount of digital information available over the internet increases. This paper is based on design of Branch Organization Rule (BOR) results in approximate answer of queries for mining. XML is popular portable language best suitable for many web technologies hence we prefer XML. While implementing XML Query Answering we are going to implement Naïve Bayes as Machine learning algorithm which we will use specially for Query Classification. We are also implementing same concept for rules classifications by using which the trees are generated after applying queries. Due to creating classification of queries our accuracy of results will increase.

***Index Terms*—** XML, Mining, query answer, Machine Learning.

I. INTRODUCTION

From past few years' data over the internet is increasing so rapidly which most of the time is in scattered and not flexible for query processing, parsing, storage It uses databases to overcome those problem XML came into the picture which represents large amount of data without any schema and specific structure. To retrieve information from XML document two techniques are exists first based on keyword search and other is based on query retrieval.

The first technique matches exact phrase or word hence it provides no support of exact query. While in second technique follows certain schema but its availability of document with schema is 50%[5]. So when we search query over document without schema it fails because of unnecessary data included into the document and query formulation become a tedious task. If at query format goes wrong the results will leads to failure than expected.

In our proposed work , we identified problems with query i.e. query formulation structure data over unstructured document mining patterns and techniques to retrieve information in given document i.e frequent data which shows an expected answer. We focused on branch organization rule to solve above stated problems with unstructured data. XML's Xquery is used by most of popular to fire query on XML document has specific schema

. In this propose work Branch Organization Rule(BOR) for representation of frequent answers in given XML document knowledge in Tree based Association Rules [1],[7] (TAR's) provides answer support in following cases.

1*) Frequency pattern from data set.*

2) Mostly XML data unable to follow DTD or XML Schema [8]. That's why user unable to specify Xquery[6] over XML document. And hence BOR will helpful to query formulation.

3) Query optimization is done construct indexes and pattern with related constraints.

4) Privacy of the data is maintained.

5) BOR will allow fast support leads to reusable when user requires quick response to his query. Information is extracted by BOR.

BOR gives useful data at abstract level of information from unstructured XML document. For testing our work we used odyssey. EO projects our training dataset. Odysssey EO project contain information of crimes in Europe[2]. By querying large information we must able to test system effectively. It will save cost and time. If old data is lost we can able to work with BOR to retrieve information.

We are taking XMLs as input to our system, process it to generate BORs.After implementing BORs, Additional task we are going to develop machine learning, using machine learning we are going to develop a system which will detect domain of that particular XML and update that XML with domain tag. So XML will classify accordingly when we use Machine learning concepts within implementation. We will use Naive Bayes al-

 DOI:10.22362/ijcert/2016/v3/i12/xxxx

gorithm for this. Naïve Bayes is Classification Machine Learning Algorithm which we focused on the implement for Rules classification. We also provide training sets of Queries those results in high accuracy during query results. The system will contain tree structure view to represent xml hierarchy.

```
<name>

<apple>
<color> red </color>
<taste> sweet </taste>
<size> 15 </size>
<price> 60</price>
</apple>

<dell>
<color> yellow</color>
<taste> sweet </taste>
<size> 12 </size>
<price> 30</price>
</dell>

</name>
```

```
<name>

<apple>
<color> red </color>
<cpu> i5 </cpu>
<size> 15 </size>
<price> 60</price>
</apple>

<dell>
<color> yellow</color>
<cpu> i3 </cpu>
<size> 12 </size>
<price> 30</price>
</dell>

</name>
```

```
<domain>fruits</domain>
<name>

<apple>
<color> red </color>
<taste> sweet </taste>
<size> 15 </size>
<price> 60</price>
</apple>

<dell>
<color> yellow</color>
<taste> sweet </taste>
<size> 12 </size>
<price> 30</price>
</dell>
```

```
<domain>computer</domain>
<name>

<apple>
<color> red </color>
<cpu> i5 </cpu>
<size> 15 </size>
<price> 60</price>
</apple>

<dell>
<color> yellow</color>
<cpu> i3 </cpu>
<size> 12 </size>
<price> 30</price>
</dell>
```

II. LITRATURE SURVEY

To create process which allows to generates BOR over unstructured XML document [3]. Those are used.Further for intensional query retrieval [1].

To work with XML file into other formats [4]. BOR also stores information in XML format so that it can be easily be reused.

Let us take simple code for XML file.[1]

```
<A>
  <B>
    <E></E>
    <F> x </F>
  </B>
  <C>
    <D></D>
  </C>
  <D>
    <F>
      <B> y </B>
      <C></C>
    </F>
  </D>
</A>
```

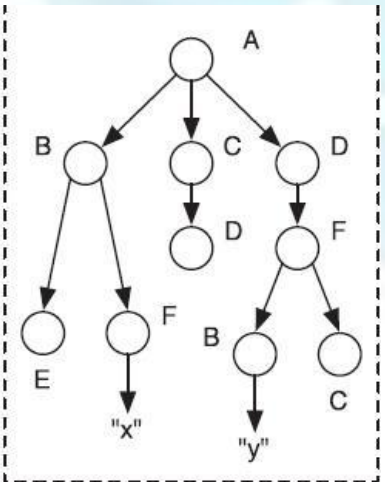

Fig 1: Simple code for XML fille

Results of association rule [3],[4],[7] were tested by using two factors support and confidence[2]. Support refers to frequency of set. Confidence refers to probability of subset at given node.

To understand support and confidence lets take one example

We are using following notation

S_t-is for subtree

D is for main tree

Cardinality (D)-number of nodes in D.

Count (D)-Number of occurances of S in D.

Support and confidence is defined as ,

Support (S_t, D)=count(S_t,D)/cardinality(D).

Confidence (S_t,D)=count(S_t,D)/count(S'_t, D).

Cardinality=33,

There are 6 times when s1 is occurred in tree D.
So,
Support (St,D)=6/33=0.1875----------(1)
Confidence=6/7=0.85---------- (2)

As branch organization rule provide overview of content and structure of document and so BOR will be guide towords in XML document .It also provide in base of intensional queries.

Take above example used to calculate support and confidence.From equation (2) confidence is 85 percent means that if there is node labeled as an in document so there is 85 percent chances child node will be B.

From equation (1) if there is any node in given graph then is 18% chances that next node will be A.So by using this user can easily find structure of document without any approximate schema and DTD, so support and confidence can be used for optimization of query.

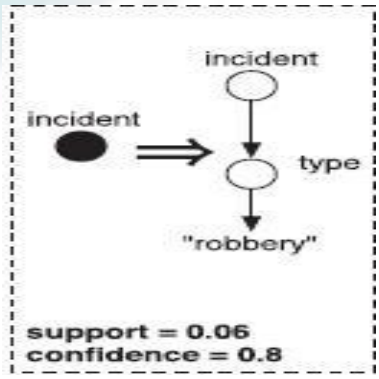

Fig 2: Rule (1)

Rule (1) is having confidence of 80% indicates that if there is a node in given tree is labeled "incident" then its having 80%

 DOI:10.22362/ijcert/2016/v3/i12/xxxx

chances that child of type is "robbery".This also states that in given area there is more probability of happening robbery.

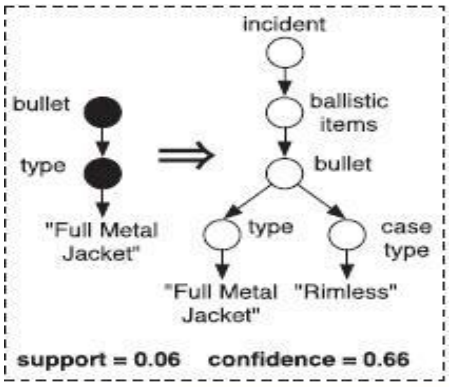

Fig 2: Rule (2)

Rule (2) states that if there is path containing sequence of node "bullet" type then its content type "full metal jacket" with confidence of 66 percents has another child labeled "case type" with content "rimless".

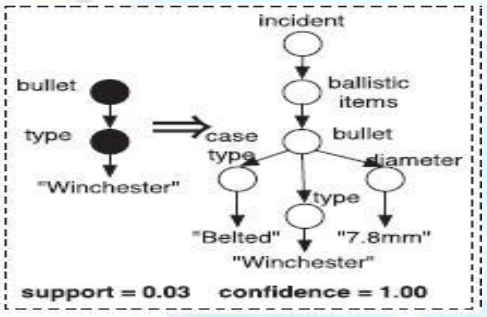

Fig 4: Rule (3)

Rule (3) shows that if there is path composed of sequence of type "Winchester" then node "bullet" with 100% confidence have 2 child labeled "diameter" and "case type" whose contents are respectively "7.8mm" and "belted".

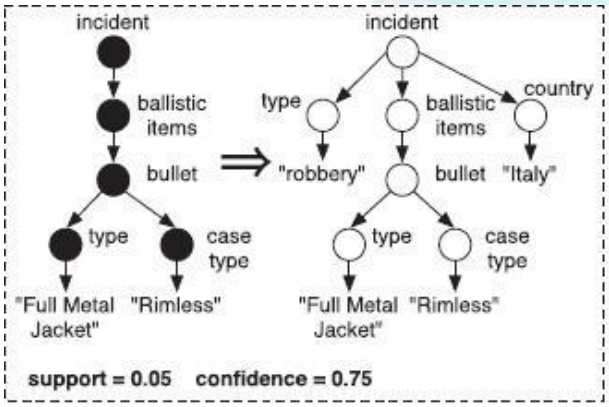

Fig 5: Rule(4)

Rule (4) shows that if there is a path of sequence of nodes "incident/ballistic item/bullet" and bullet has two children labeled "type" and "case type" whose contents are "full metal jacket" and "rimless" respectively then node incident with 75% of confidence has two more children labeled type and counting whose content are "robbery" and "Italy".

III. PROPOSED SYSTEM

Branch oraganization Rule

Branch Organization Rule (BOR) for representation of frequent answers in given XML document. BOR will helpful to query formulation. BOR will allow fast support leads to reusable when user requires quick response to his query.information is extracted by BOR.

BOR is divided into two parts.

1) Mining of subtree.find subtree.

2) Computing interesting rule.find confidence.

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

1: frequent subtrees

2: FS_t = Find Frequent Subtrees (D, min supp)

3: ruleSet = α;

4: for all s_t ε $F S_t$ do

5: // rules computed from s_t

6: tempSet = Compute-Rules (s_t; minconf S)

7: // all rules

8: ruleSet = ruleSet U tempSet

9: end for

10: return ruleSet

By providing XML document Xd as input, support threshold is minsupp and confidence threshold is minconf. The use of this document is faster than firing query on main document.

In particular, for each query set now we need to do next.for each set the first tree is reference of its root are added to the reference of roots other tree and some procedure is applied to children of two root recursively.So we need to create two sets A and C i.e antecedent and consequent tree of all BOR index.

Algorithm 2. Create-Index (D)

1: for all Di 2 D do

 DOI:10.22362/ijcert/2016/v3/i12/xxxx

2: for all dj 2 Di with j 2 f2; 3; . . . ng do

3: references(root(d1)) =

4: references(root(d1)) [references(root(dj))

5: sumChildren (d1; dj)

6: end for

7: end for

8: return D

Function 4. sumChildren (T1; T2)

1: for all x 2 children(root(T2)) do

2: if 9 c 2 children(root(T1)) j c ¼ x then

3: references(root(c)) = references(root(c))

[references(root(x))

4: c ¼ sumChildren(c,x)
5: else
6: addChild(root(T1),x)

7: end if
8: end for
9: return T1

Once Algorithm is applied for file, output is tree whose node contains references to one or more rule stored in XML file.

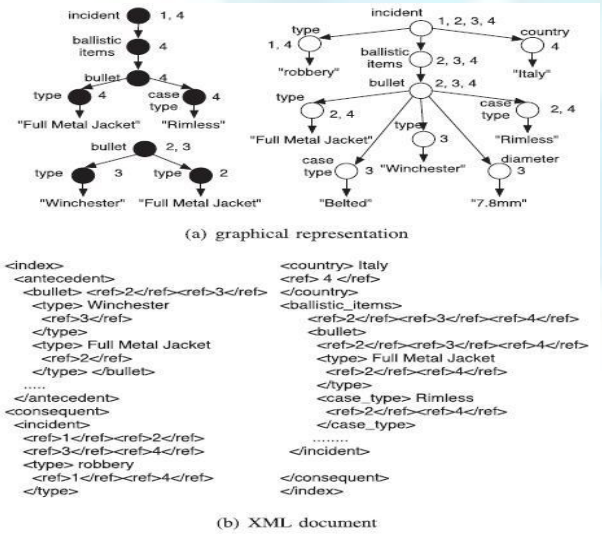

Fig 6: BOR index

Naïve Bayes Algorithm:

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems. [10] Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for Understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data.[9]

A naive Bayes classifier applies Bayes' Theorem in an attempt to suggest possible classes for any given text. To do this, it needs a number of previously classified documents of the same type. The theorem is as follows:

$$P(h/D) = \frac{P(D/h)\,P(h)}{P(D)}$$

P(h) : Prior probability of hypothesis h
P(D) : Prior probability of training data D
P(h/D) : Probability of h given D
P(D/h) : Probability of D given h

Naïve Bayesian Classification

It is based on the Bayesian theorem It is particularly suited when the dimensionality of the Inputs are high. Parameter estimation for naive Bayes models uses the method of maximum Likelihood. In spite over-simplified assumptions, it often performs better in many complex realworld Situations

Advantage: Requires a small amount of training data to estimate the parameters

Theory:
Derivation:
D : Set of tuples

- Each Tuple is an 'n' dimensional attribute vector
- X : (x1,x2,x3,…. xn)

Let there be 'm' Classes: C1, C2, C3…Cm
Naïve Bayes classifier predicts X belongs to Class Ci iff

- P (Ci/X) > P(Cj/X) for 1<= j <= m , j <> i

Maximum Posteriori Hypothesis

- P(Ci/X) = P(X/Ci) P(Ci) / P(X)
- Maximize P(X/Ci) P(Ci) as P(X) is constant

With many attributes, it is computationally expensive to evaluate P(X/Ci). Naïve Assumption of "class conditional independence"

$$P(X/.Ci) = \prod_{k=1}^{n} P(x_k / Ci)$$

P(X/Ci) = P(x1/Ci) * P(x2/Ci) *…* P(xn/ Ci)

Query answering:-

 DOI:10.22362/ijcert/2016/v3/i12/xxxx

User query main XML file can be easily transform because of (1) system will take less time to answer (2) approximate intensional answer are more useful in some cases.For query for XML file Xquery is used. Xquery indexes path expression FlOWR and Count for our purpose to retrieve approximate description satisfying the query with modifying new element.

IV. MATH:

MATHEMATICAL MODEL:

Memory utilization: The experiment was done to find space saving capability. We implemented Miner using Java (jdk 1.6.0) compilers. The sizes of the application and the web application were 1175KB and 5160KB, respectively. It was required for server having Glassfish installed.

Consider a XML with X Tags. The software/resource to be installed is V Variable (size of V in MB).

In general the total memory occupied in the application is the sum of all the memory installed on each Tag

Y = X*V MB.

Where,

Y is Temporary variable

X is Number of Tags.

XML occupy: (X-1)*Vc MB.

One (At least One) system must be installed with server program. So

Server occupy : (1) * Vs MB + V MB

Then the total memory occupied in the system is

Yvc = (X-1)Vc + Vs + V MB.

Then the total memory saved in this system is

Ysvc = XV - ((X-1)Vc = Vs +V)MB.

Percentage utilization of memory on this system using our concept:

((XV-((X-1)Vc + Vs + V))/XV)100

The database size at server side will grow dynamically so it is not taken into consideration in above calculations.

Number of systems in the lab, X = 5; Size of the software to be worked

,V=162MB(jdk1.6.0)+1175KB + 5160KB =172.22MB

So, in normal lab if each system has installed this software, then total memory occupied i

Y = 5 *50KB = 250KB;

Utilization of memory for updating of BOR for new update tags =5 with out time stamp

Y2 = 5*50 = 250KB

Utilization of memory for updating of BOR for new update tags =5 with out time stamp just scan 1 root element

Y3 = 1*50 = 50KB

% Performance updating when no updates are done

$P = Y_2/Y_1 = 250/50 = 5$ x

RESULTS:

The first diagram shows the input to the project.

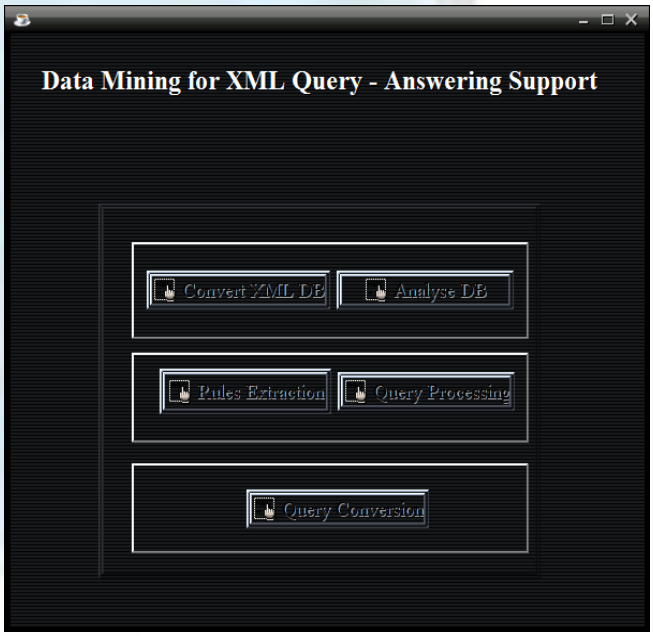

Fig 7: Input

 DOI:10.22362/ijcert/2016/v3/i12/xxxx

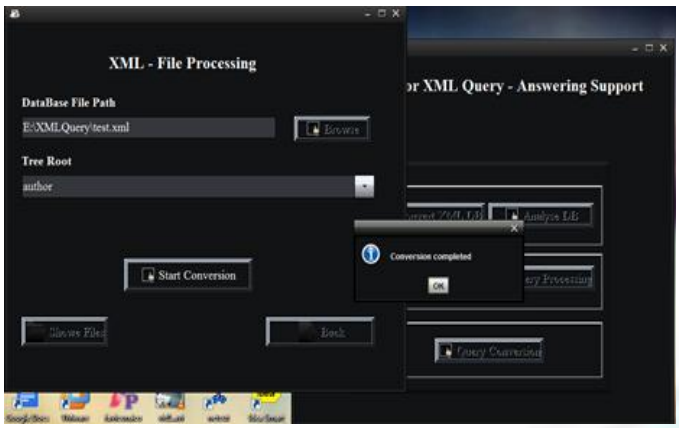

Fig 8: Output screen

Fig 9 Database Analysis

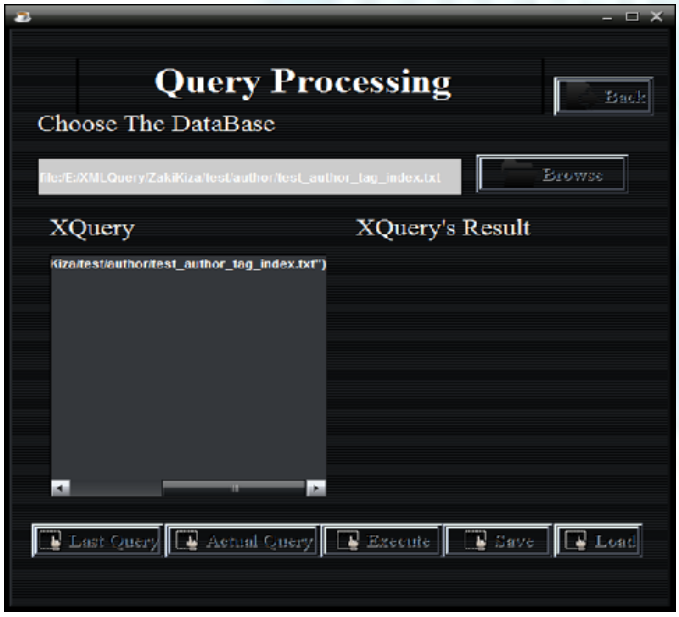

Fig 10. Query Processing

Analysis:

As in existing systems TARs are implemented while in our proposed system we are going to use BOR along with Machine Learning concepts such that it increases accuracy of results too.

In second chart given below we find Count of XML queries execution with respect to execution time. In existing system it requires less time as in our system we are not only executing XML queries but are updating datasets too. So that it requires more time for execution. And it was our drawback of system but as result accuracy increases this drawback overcomes. And we made it possible to reduce that execution time for some time.

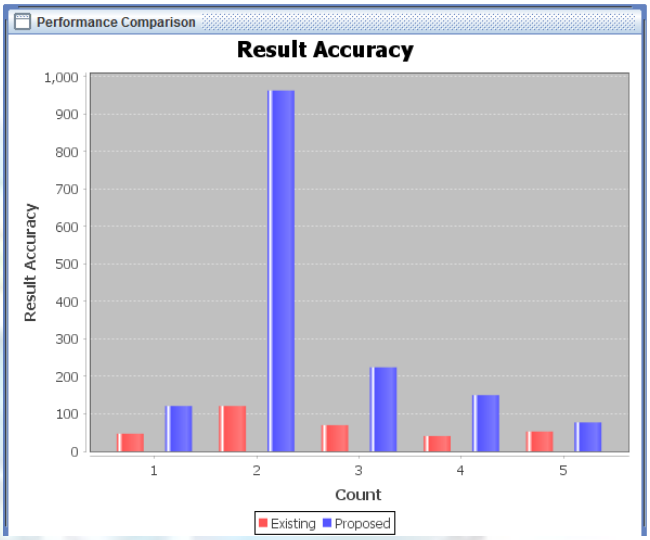

Fig11. Result Accuracy

Above analysis graph represent the Result Accuracy with respect to No. of Queries provided to system. Here Result accuracy represents no. of accurate results with respect to no. of input queries for answer module. Group of queries are given input to module with 'n' of iteration. After training phase it will outs average results or similar results. So It requires training with maximum XML document data which has tree like structure.

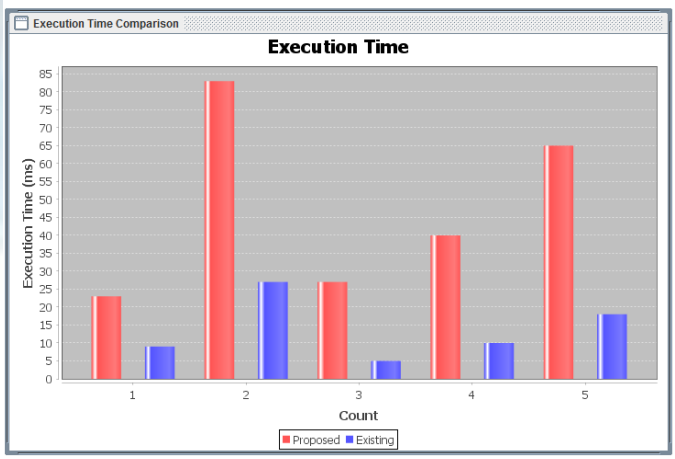

Fig 12. Execution Time for XML Query answering model

Above Chart represents no. of Query Count with respect to execution time. This Execution time taken in ms. Count represents iteration of same group of queries which are given as in-

 DOI:10.22362/ijcert/2016/v3/i12/xxxx

put to our Intelligent Query Answering Project.

Contribution to society: Applying these concepts we can implement projects like Law Query Answering or i-Help Portals to get quick responses for queries what people asked in very short time and in accurate percentage. It helps to improve quick response time once proper training done by datasets what used for training purposes in Naïve Bayes classifiers.

V. CONCLUSION:

Our developed system concludes to following benefits

- Intelligent XML Query-Answering work will update the XML data with frequent new XML data.

-Mine frequent association rules without imposing any a-priori restriction on the structure and the content of the rules

- We implement Naïve Bayes Algorithm for Rule Classification and New Rule Design

-We also implement Machine learning concepts while generating Tree structures and Query Answer support. Similar kind of queries is duly classified between various classes which increases accuracy to answer.

-Stores mined information in specific XML format

-Uses the extracted knowledge to gain information about the original data sets.

REFERENCES AND BIBLIOGRAPHY:

[1] Mirjana Mazuran, Elisa Quintarelli, and Letizia Tanca "Data Mining for XML Query-Answering Support", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 8, AUGUST 2012

[2]A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy Preserving Mining of Association Rules," Proc. Eighth ACM Int'l Conf. Knowledge Discovery and Data Mining, pp. 217-228, 2002.

[3]C. Combi, B. Oliboni, and R. Rossato, "Querying XML Documents by Using Association Rules," Proc. 16th Int'l Conf. Database and Expert Systems Applications, pp. 1020-1024, 2005

[4]D. Braga, A. Campi, S. Ceri, M. Klemettinen, and P. Lanzi, "Discovering Interesting Information in XML Data with Association Rules," Proc. ACM Symp. Applied Computing, pp. 450-454, 2003.

[5]D. Barbosa, L. Mignet, and P. Veltri, "Studying the XML Web: Gathering Statistics from an XML Sample," World Wide Web, vol. 8, no. 4, pp. 413-438, 2005.

[6]E. Baralis, P. Garza, E. Quintarelli, and L. Tanca, "AnsweringXMLQueries by Means of Data Summaries," ACM Trans.InformationSystems, vol. 25, no. 3, p. 10, 2007.

[7]L. Feng, T.S. Dillon, H. Weigand, and E. Chang, "An XMLEnabled Association Rule Framework," Proc. 14th Int'l Conf. Database and Expert Systems Applications, pp. 88-97, 2003.

[8] World Wide Web Consortium, XML Schema, http:// www.w3C.org/TR/xmlschema-1/, 2001.

[9]Chai, K.; H. T. Hn, H. L. Chieu; "Bayesian Online Classifiers for Text Classification and Filtering", Proceedings of the 25th annual international ACM SIGIR conference on Research and Development in Information Retrieval, August 2002.

[10] DATA MINING Concepts and Techniques, Jiawei Han, Micheline Kamber Morgan Kaufman Publishers, 2003

ABOUT AUTHORS

Zaware Parag B received B.E Computer engg degree from ZES Dnyanganga college of engineering pune in 2012 and persuing M.E Comp from VACOE Ahmednagar.

Prabhudev S: Received M.Tech degree in Computer Science and engineering from University of B D T College or Engineering in 2012, since 2012 he has been an assistant professor in the Vishwabharati Academy's college of Engineering (Pune University) His current research areas are cloud computing and data mining. He has published International and National papers in various journals. He is an Editorial board reviewer of American Journal of computer science and engineering. He is member of **AIE, IAENG**