International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600

http://www.ijcsjournal.com Volume 5, Issue 1, No 28, 2017 ISSN: 2348-6600
Reference ID: IJCS-291 PAGE NO: 1918-1926

Extensible markup Language approximate query answering Using data mining, intentional based on Tree-Based Association Rules

A Mohan Kumar M. Tech,,[#1] Dr. G Rajendra Kumar M. Tech, Ph.D.,[#2]

[# 1 2] Computers Science & Engineering,
SRI SIVANI ENGINEERING COLLEGE, NH-5, CHILAKAPALEM, 532001,
SRIKAKULAM Dst.(AP)
[1] mohankumar7arangi@gmail.com,9440063579.
[2] rajendragk@rediffmail.com,9573355762.

Abstract - With the increasing popularity of XML for data representations, there is a lot of interest in searching XML data. Due to the structural heterogeneity and textual content's diversity of XML, it is daunting for users to formulate exact queries and search accurate answers. Therefore, approximate matching is introduced to deal with the difficulty in answering users' queries, and this matching could be addressed by first relaxing the structure and content of a given query, and then looking for answers that match the relaxed queries. Ranking and returning the most relevant results of a query have become the most popular paradigm in XML query processing. However, the existing proposals do not adequately take structures into account and they therefore lack the strength to elegantly combine structures with contents to answer the relaxed queries. To address this problem, we first propose a sophisticated framework of query relaxations for supporting approximate queries over XML data. The answers underlying this framework are not compelled to strictly satisfy the given query formulation, instead they can be founded on properties inferable from the original query. We then develop a novel top-k retrieval approach which can smartly generate the most promising answers in an order correlated with the ranking measure

Keywords: - Extensible markup Language (XML),approximate query answering, data mining, intentionalinformation, Tree-Based Association Rules.

INTRODUCTION

The In recent years the database research field hasconcentrated on XML (eXtensible Markup Language as aflexible hierarchical model suitable to represent hugeamounts of data with no absolute and fixed schema, and apossibly irregular and incomplete structure. There are twomain approaches to XML document access: keyword-basedsearch and query-answering. The first one comes from thetradition of information retrieval, where most searches areperformed on the textual content of the document; this meansthat no advantage is derived from the semantics conveyed by the document structure .As for query-answering, since query languages for semi

structured data rely the on document structure to convey itssemantics, in order for query formulation to be effective usersneed to know this structure in advance, which is often not thecase. In fact, it is not mandatory for an XML document thave a defined schema: 50% of the documents on the web do not possess one . When users specify queries withoutknowing the document structure, they may fail to retrieveinformation which was there, but under a different structure.This limitation is a crucial problem which did not emerge inthe context of relational database management systems .Frequent, dramatic outcomes of this situation are either theinformation overload problem, where too much data areincluded in the answer because the set of keywords specifiedfor the search captures too many meanings, or the informationdeprivation problem, where either the use of inappropriatekeywords, or the wrong formulation of the query, prevent theuser from receiving the correct answer. As aconsequence,when accessing for the first time a large dataset,gaining some general information about its main structuraland semantic characteristics helps investigation on more specific details .This paper addresses the need of getting the gist of thedocument before querying it, both in terms of content andstructure. Discovering recurrent patterns inside XMLdocuments provides high-quality knowledge about thedocument content: frequent patterns are in fact intentionalinformation about the data contained in the document itself,that is, they specify the document in terms of a set ofproperties rather than by means of data. As opposed to thedetailed and precise information conveyed by the data, thisinformation is partial and often approximate, but synthetic,and concerns both the document structure and its content. Inparticular, the idea of mining association rules to providesummarized representations of XML documents has beeninvestigated in many proposals either by using languages andtechniques developed in the XML context, or byimplementing graph- or tree-based algorithms .

RELATED WORK

Existing System

In, Weigel et al. study the relationship between scoring methods and XML indices for efficient ranking, and propose IR-CADG, an extension to data guides to account for keywords, which integrates ranking on structures and contents. In, Yan et al. propose a preference-based ranking model to deal with approximate queries in XML. Their approach relies on a user-specified interest score to generate the top-k answers, which increases users' query burdens. It fails in discovering the potential answers which users may have interest in, for the lack of a similarity assessment designed for extracting the inherent semantics presented in XML data sources. In, Maio et al. presented an ontology-based retrieval approach, which supports data organization and visualization, and provides a friendly navigation model. Built on the availability of a bulk of ontologies (knowledge), existing commercial solutions (e.g. Google knowledge graph, etc.) accomplish the ontology-based information retrieval and question answering (IR&QA) on structured and unstructured data. However, as introduced in, developing ontologies is a time-consuming task which often needs an accurate domain expertise to tackle structural and logical difficulties of concepts as well as conceivable relationships.

Proposed System

We propose a query relaxation method incorporating structures and contents, as well as the factors that users are more concerned about, for supporting approximate queries over XML data. In particular, our method surmises the factors that users are more concerned about based on the analysis of user's original query for supporting query relaxations. In addition, our approach differentiates the relaxation ordering instead of giving an equal importance to each node to be relaxed. In particular, the first relaxed structure to be considered is the one that has the highest similarity coefficient with original query and the first node to be relaxed is the least important node.

We customize the similarity relation assessment by analyzing the inherent semantics

IMPLEMENTATION

Answer Score

The scoring function S applied to an approximate match<AQ(T)> combines the approximations occurring at both structures and contents (including the approximations occurring at query conditions), and it returns a score such that the higher is S(<AQ(T)>), the higher is the overall approximation required for the matching. Are functions which evaluate the approximations occurring at structures and contents respectively. In which, wi is the normalized weight of the corresponding attribute node, CQi and CAQi are corresponding attribute values, and k is the number of attribute nodes specified by the query. Note that, since there exists a 1-1 mapping relationship between a tree pattern query and its match in data trees, the answer score of could be also defined by using the given tree pattern query and its relaxations. In other words, the score of Intuitively, the coefficient in the above definition reflects the user's preference on structures and contents. Theoretically, for the XML documents with different characters, the value of should be different, whereas this value could be obtained by means of data mining on the database workload-log of past users' queries during the off-line preprocessing step. The answer score of an answer (a match) measures the relevance of that answer to the user's query. For a given parameter k, the top k problem is searching the best top-k answers (matches) ordered from best (highest answer score) to the worst. In particular, we have the following top-k problem definition: (Top-k Answers)Given a tree pattern query (or its representative query)Q and Q's approximate queries AQ, the top-k answers to Q, and to any of AQ, is the set of answers (matches) to Q, and to any of AQ, with the k highest answer scores.

Structure Relaxation Planning

Our structure relaxation planning relies on query-rewriting. To avoid generating a potentially large number of relaxed queries, our approach decides the structural relaxations to be considered in the priority list depending on their corresponding tree similarity coefficients. More specifically, the relaxation with the highest tree similarity coefficient value will be first taken into consideration. As mentioned earlier, for a tree pattern query Q, it is sufficient to associate the corresponding tree similarity coefficient, which can be accessed in a constant time by using the CDAG during the query processing with each approximate query of Q. At each CDAG node, we keep the maximum theoretical upper bound for an approximate match that satisfies the tree pattern query associated with that node. If the query at that node includes all nodes of the original query, then a match that satisfies this tree pattern query cannot be further extended and its coefficient upper bound value is equal to its tree similarity coefficient. If the tree pattern query does not include all the nodes from the original query (e.g., it is a relaxation of the original query where some leaf deletion operations are applied), we store a pointer in the CDAG to the CDAG node containing the best relaxation that approximate (partial) match could satisfy. We could keep pointers in the CDAG to access information such as the coefficient upper bound values of all possible configurations of approximate queries. During query evaluations, tree similarity coefficients are accessed in constant time by using a hash table to check the approximate matches against the tree pattern queries stored in CDAG.

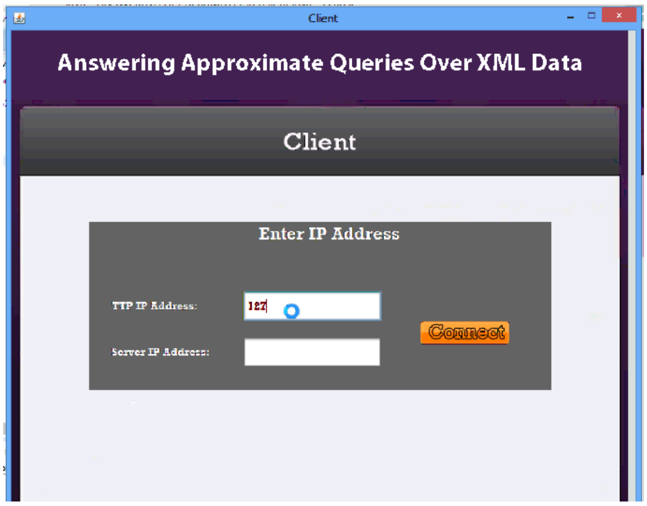
Content Relaxation Planning

Our content relaxation planning relies on query-rewriting. In particular, the sub-threshold for each specified attribute node Given a tree pattern query Q, assume that wi is the weight of a specified attribute node In our content rewriting solution, we rewrite the given textual content into a set of expanded contents (content substitutes), and then evaluate the union of these contents on XML data sources. In particular, for each condition Ci occurring in the attribute node Ai = q, if Ai is a categorical attribute node, based on the Semantic Trees, we will first extract the expanded contents of Ai having the similarity coefficient above the sub-threshold i, and then generate the relaxed query conditions by using these expanded contents. If Ai is a numerical attribute node, then we will replace the original value q with a relaxation range to generate the relaxed query conditions. The time consumption of our content rewriting consists of two components: computing the similarity coefficients of textual contents and replacing the textual contents. From the complexity analysis introduced, we can easily find that, the time consumption of our content relaxation is mainly used in computing the similarity coefficients of categorical textual contents and the standard deviation of numerical attribute nodes.

Top-k Query Execution

Ranking and returning the most relevant results of a query becomes the most popular paradigm in XML query processing. Next we will describe our solution, which employs the preparations made in the off-line step, to efficiently search the top-k most relevant answers from XML data sources. The execution of our top-k query processing is as follows. Our approach starts by evaluating all the structure relaxations and content relaxations, which are maintained by using the structure and content relaxation plans in advance. Then it determines the matches with the highest score potential, which relies on answer scores calculated by using our proposed scoring function, to prioritize approximate matches. It then expands top-k answers by adding the next best matches. If the number of answers is less than the given parameter k, it will repeatedly execute the loop until all the top-k answers are obtained.

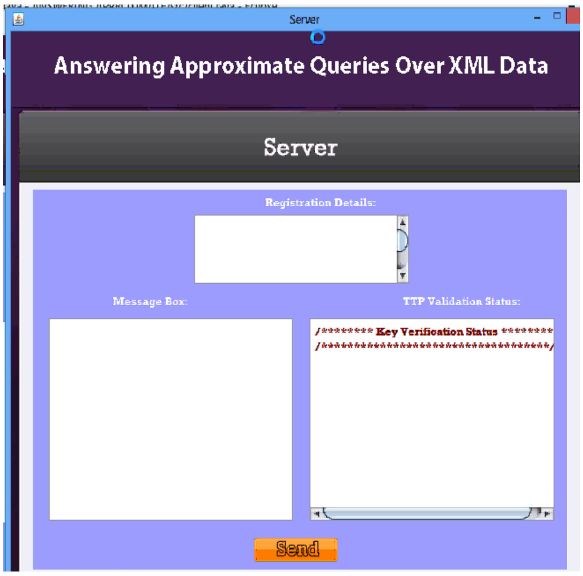
EXPERIMENTAL RESULTS



TTP IP Address: 127

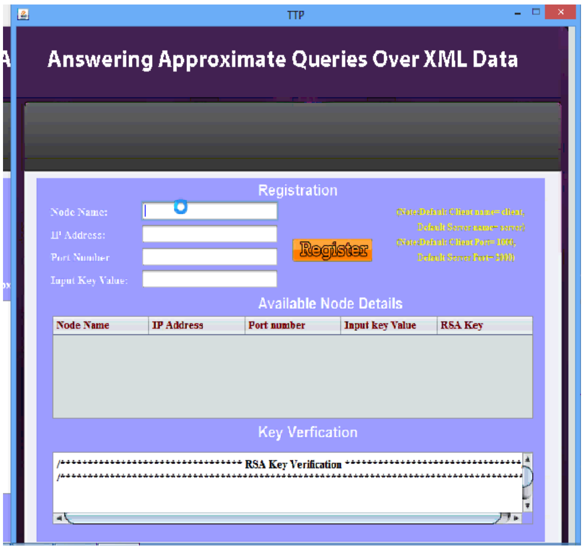
Server IP Address:

Fig:-1 Client Layout



```
/******** Key Verification Status ********
/*********************************/
```

Fig:-2 Server



TTP

Answering Approximate Queries Over XML Data

Registration

Node Name:

IP Address:

Port Number

Input Key Value:

(Note:Default Client name= client,
Default Server name= server)
(Note:Default Client Port= 1000,
Default Server Port= 2000)

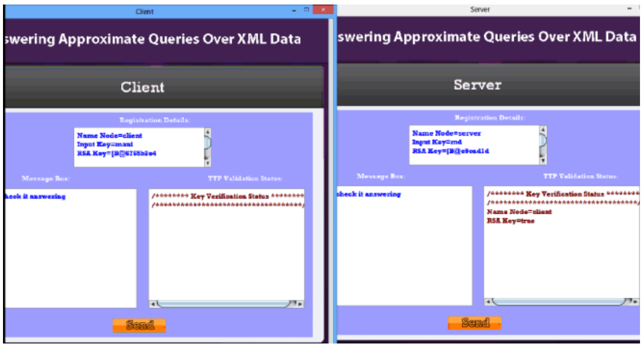
Available Node Details

Node Name	IP Address	Port number	Input key Value	RSA Key

Key Verfication

```
/********************************* RSA Key Verification *********************************
/****************************************************************************************
```

Fig:-3 Nodes Registration



Client

swering Approximate Queries Over XML Data

Client

Registration Details:

Name Node=client
Input Key=mani
RSA Key=[B@6758b2e4

Message Box:

heck it answering

TTP Validation Status:

```
/******** Key Verification Status ********
/***************************************/
```

Server

swering Approximate Queries Over XML Data

Server

Registration Details:

Name Node=server
Input Key=md
RSA Key=[B@e9ead1d

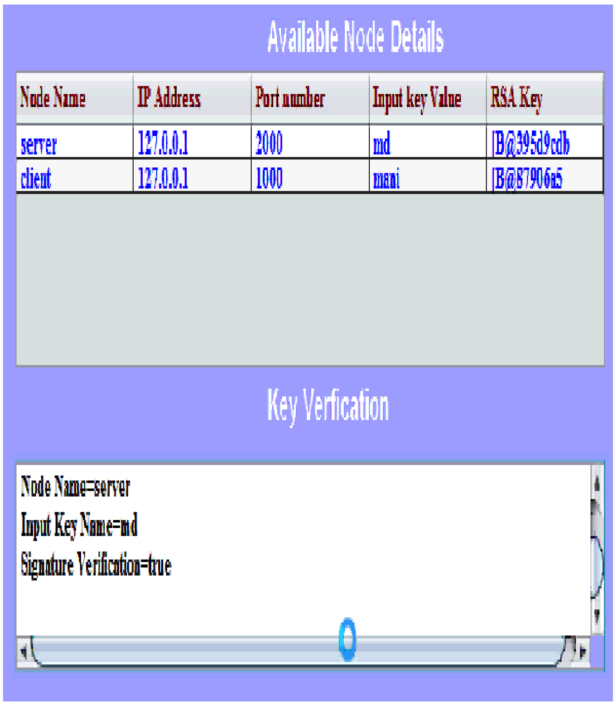
Message Box:

check it answering

TTP Validation Status:

```
/******** Key Verification Status ********
/***************************************/
Name Node=client
RSA Key=true
```

Fig:-4 Data Sending Process



Available Node Details

Node Name	IP Address	Port number	Input key Value	RSA Key
server	127.0.0.1	2000	md	[B@395d9cdb
client	127.0.0.1	1000	mani	[B@87906a5

Key Verfication

Node Name=server
Input Key Name=md
Signature Verification=true

Fig:-5 Key Verification

CONCLUSION

We propose a sophisticated frame work of query relaxations for supporting approximate queries over XML data. Typically, our approach surmise factors that users are more concerned about based on the analysis of the user's original query and assigns a corresponding weight to each attribute node for supporting query relaxations. In addition, our approach adequately takes structures into account and it therefore has the ability to elegantly combine structure with contents to answer approximate queries. We also present an effective approach for searching the top-k most relevant answers from a large number of XML data sources along with our query relaxation framework. The rule learning algorithm searches for these patterns and creates a trinity tree, which is then used to learn a regular expression that represents the template that was used to generate input web documents. We implement our web sites proved that our technique achieves very high precision and recall with a learning and extraction time that is almost negligible. We also identified a means to introduce a bias to the search procedure that improves its efficiency Without Negative Impact.

REFERENCES

[1]Arasu, A. and Garcia-Molina, H. Extracting Structured Data from Web Pages.In Proc.of the ACM SIGMOD Int. Conf. on Management of Data, 2003.

[2] Baumgartner, R., Flesca, S. and Gottlob, G. Visual Web Information Extraction with Lixto. In Proc. of Very Large DataBases (VLDB), 2001.

[3]Chakrabarti, S. Mining the Web: Discovering Knowledge from Hypertext Data. ISBN: -155860-754-4. Morgan Kaufmann Publishers, 2003.

[4] Chang, C. and Lui, S. IEPAD: Information extraction based on pattern discovery. In Proc. of 2001 Int. World Wide Web Conf., pp. 681–688, 2001.

[5] Crescenzi, V., Mecca, G. and P. Merialdo. ROADRUNNER: Towards automaticdataextraction from large web sites. In Proc. of the 2001 Int. VLDB Conf, pp. 109–118, 2001

[6]Laender, A. H. F., Ribeiro-Neto, B. A., Soares da Silva, A. and Teixeira, J. S. A Brief Survey of Web Data Extraction Tools. ACM SIGMOD Record 31(2), pp 84-93. 2002.

[7]Zhai, Y. and Liu, B. Extracting Web Data Using Instance-Based Learning. In Proc. of Web Information Systems Engineering (WISE), pp. 318-331, 2005.

[8] Zhai, Y. and Liu, B. Structured Data Extraction from the Web Based on Partial Tree Alignment. IEEE Trans. Knowl. Data Eng. 18(12), pp. 1614-1628, 2006.

[9] Shasha, D., Wang, J.T.L, Shan, H., Zhang, K. ATreeGrep: Approximate Searching in Unordered Trees. In 14th International Conference on Scientific and Statistical Database Management. Edinburgh, Scotland, 2002

[10] Soderland, S. Learning to Extract Text-based Information from the World Wide Web, In Proceedings of Third International Conference on KnowledgeDiscovery and Data Mining (KDD-97). 1997.