Bayesian Network and Network Pruning Strategy for XML Duplicate Detection

[1]Ms. Trupti Patil, [2]Siddheshwar Patil, [3]Ms. Swapnali Patil, [4]Sadanand S. Howal

[1,3] Lecturer Computer Science, Latthe Education Society's Polytechnic, Sangli, Maharashtra, India
[2] Asst. Prof. Information Technology, Annasaheb Dange College of engineering & Technology Ashta, Sangli, Maharashtra, India
[3] Asst. Prof. Information Technology, Rajarambapu Institute of Technology, Sakhrale, Maharashtra, India
[1]tpatil.7137@gmail.com, [2]siddheshwar.patil@gmail.com, [3]swapnali.patil9@gmail.com
[4]sadanand.howal@ritindia.edu

Abstract: Data Duplication causes excess use of redundant storage, excess time and inconsistency. Duplicate detection will help to ensure accurate data by identifying and preventing identical or similar records. There is a long work on identifying duplicates in relational data, but only a slight solution focused on duplicate detection in more complex hierarchical structures, like XML data. Hierarchical data are defined as a set of data items that are related to each other by hierarchical relationships such as XML .In the world of XML there are not necessarily uniform and clearly defined structures like tables. Duplicate detection has been studied extensively for relational data. Methods devised for duplicate detection in a single relation do not directly apply to XML data. Therefore there is a need to develop a method to detect duplicate objects in nested XML data. In proposed system duplicates are detected by using duplicate detection algorithm called as XMLDup. Proposed XMLDup method will be using Bayesian network. It determine the probability of two XML elements being duplicates by considering the information within the elements and the structure of information. In order to improve the Bayesian Network evaluation time, pruning strategy is used. Finally work will be analyzed by measuring Precision and Recall value.

Keywords – *Data Duplication, Bayesian Network (BN), XML Duplicate (XMLDup), Network Pruning, XML*

*****

I. INTRODUCTION

Hierarchical data is distinct set of data items that are related to each other by hierarchical relationships. In Hierarchical relationships one data item is the parent of another item. The structure is represented by using the information of parent and child relationships in which each parent can have many children, but each child has only one parent. All attributes of a specific record are listed under an entity type. Hierarchical data is distinct set of data items that are related to each other by hierarchical relationships. In Hierarchical relationships one data item is the parent of another item. The structure is represented by using the information of parent and child relationships in which each parent can have many children, but each child has only one parent. All attributes of a specific record are listed under an entity type.

An XML document is a tree and consequently a single XML data type instance can
represent a complete hierarchy. XML is used both for large scale electronic publishing of data, and for the
exchange of data on the Web and elsewhere. The two main features of XML are that the data is organized hierarchically, and is semi-structured, mixing content, e.g., text and structure, using so called XML tags.

Several problems arise in the context of data integration where data from distributed and heterogeneous data source is combined. One of these problems is the possibly inconsistent representation of the same real world object in the different data sources.When combining the data from heterogeneous sources the ideal result is a unique complete and correct representation for every object such that data quality can only be achieved through data cleansings, where the most important task is to ensure that an object only has one representation in the result. This requires the identification of object and is referred to as "Duplicate detection".

II. RELATED WORK

Lus Leitao, Pavel Calado, and Melanie Herschel, presented a novel method for XML duplicate detection called XMLDup. XMLDup uses the Bayesian Network to determine the probability of two XML elements being duplicates. To improve the efficiency of network evaluation Network Pruning Strategy will be presented. When compared to another state-of-the-art XML duplicate detection algorithm, XMLDup constantly showed better results concerning both efficiency and Effectiveness [1].

F. Naumann and M. Herschel presented Similarity measures are used to automatically identify duplicates when comparing two records. Well-chosen similarity measures and Well-designed algorithms improve the efficiency and effectiveness of duplicate detection. Algorithms are developed to perform on very large volumes of data in search for duplicates [2].

L. Leitao and P. Calado, proposed a novel method that automatically restructures database objects in order to take full advantage of the relations between its attributes and avoids the need to perform a manual selection. They argued that structure can indeed have a significant impact on the process of duplicate detection[3].

P. Calado, M. Herschel, and L. Leitao presented a description and analysis of the different approaches, and a relative experimental evaluation performed on both artificial and real-world data[4]. M.Weis and F. Naumann presents generalized framework for object identification and an XML specific specialization. DogMatix algorithm was introduced for object identification in XML that uses the heuristics to determine candidate descriptions domain-independently [5].

L. Leitao, P. Calado, and M. Weis, proposed a novel method for fuzzy duplicate detection in hierarchical and semi-structured XML data and also proposed a Bayesian network. A Bayesian network model, are able to accurately determine the probability of two XML objects in a given database being duplicates.The model also provides great flexibility in its configuration, allowing the use of different similarity measures for the field values and different conditional probabilities to combine the similarity probabilities of the XML elements [6].

III. Proposed System

Fig.1 shows architecture of proposed system, the XML files will be selected by user which contains duplicate record. The XML files will be parsed by using DOM (Document to Object Modeling) API parser. DOM API parses the XML document and maps the records in the file to the java objects. Now, to construct the Bayesian Network model mapped objects will be used.

Bayesian network provide brief specification of joint probability distribution. In the acyclic graph, where each node represent random variable and edges represent dependencies between two variables. For identification of duplicates in XML structure, a BN is formed. If two XML elements being duplicates it is depends on values of duplicates and children nodes are duplicates.

Bayesian Network will have a node labeled with the records parent node and a binary random variable will be assigned. A binary random variable takes the value 1 to represent the reality that the XML nodes are duplicates. It takes the value 0 to represent the reality that the nodes are not duplicates. Then parent node will have sub parent nodes for child nodes in XML tree, and this continues until the leaf nodes.

A list containing parent nodes will be given to XMLDup method for calculating probabilities. The probability will be obtained by the prior probabilities correlated with the BN leaf nodes, which will set the in-between node probabilities; in anticipation of the root probability is found. To calculate root probability that propagates the similarity of leaf nodes up the tree until reach to the root node, probabilities will be used to calculate final probability. If the calculated probability is more than threshold then the two records in BN structure are duplicates.

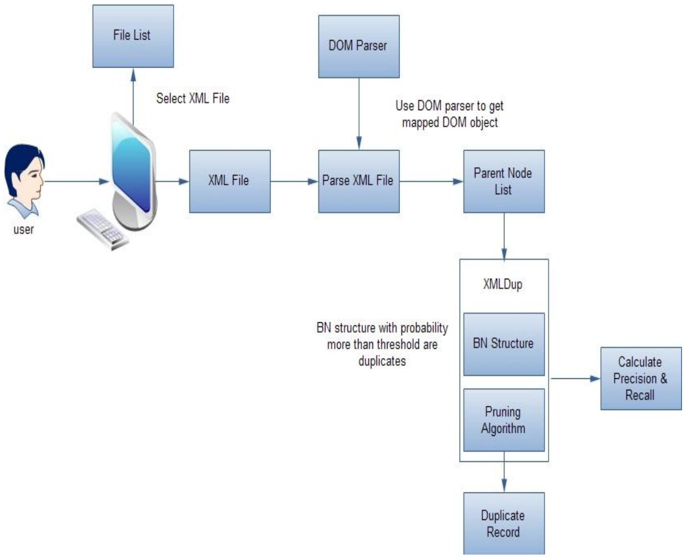
File List
Select XML File
DOM Parser
Use DOM parser to get mapped DOM object
user
XML File
Parse XML File
Parent Node List
XMLDup
BN structure with probability more than threshold are duplicates
BN Structure
Pruning Algorithm
Calculate Precision & Recall
Duplicate Record

Fig.1 System architecture of proposed System

Duplicate records then separated from the list. In order to improve the Bayesian Network evaluation time, the proposed lossless pruning strategy will be used. The meaning of is that no duplicate objects are lost. Pruning algorithm will be implemented and pass it the list of Bayesian Network models calculated at the beginning. This algorithm takes each Bayesian Network model for evaluation. Traversing through Bayesian Network Structure, it propagates the probabilities of values to the parent nodes. If probability of parent is more than threshold value then records represented by Bayesian Network structure are considered as duplicates. Likewise algorithm checks the entire list and finds duplicates. Finally work will be analyzed by measuring Precision and Recall value.

IV. Methodology

The proposed system can be divided into following modules:

1. Parsing XML files Module
2. Bayesian Network (XMLDup) Module
3. Network Pruning
4. Testing of System
5. Analysis of Work

A. Parsing XML files Module

In Parsing XML files module, it accesses the files from user. The file contains hierarchical data such as XML files etc.XML files will be selected for comparison. XML files are parsed that contains duplicate records. The proposed system uses DOM (Document to Object Modeling) API. DOM API parses the xml document and maps the records in the file to the java objects.

B. Bayesian Network (XMLDup) Module

XMLDup uses a Bayesian network to determine the probability of two XML elements being duplicates. Mapped objects will be used to construct Bayesian Network. Bayesian Network will have a node labeled with the records, parent node and a binary random variable is assigned. This variable takes the value 1 to represent the reality that the XML nodes are duplicates. It takes the value 0 to represent the reality that the nodes are not duplicates. Then the parent node will have sub parent nodes for child nodes in XML tree, and this continues until the leaf nodes. Compare the nodes with each other and check whether these are greater than threshold value or not. If they are then take those nodes.

C. Network Pruning

This algorithm takes each Bayesian Network model for evaluation. By traversing through Bayesian Network structure, it propagates the probabilities of values to the parent nodes. If probability of parent node is more than threshold values then records represented by Bayesian Network structure are considered as duplicates. Likewise algorithm checks the entire list and finds duplicates.

D. Testing of System

For Testing of system Country, Cora, Employee Data sets will be used.

E. Analysis of Work

For Analysis of work we will calculate two measures, Precision and Recall measure.

Precision measures the percentage of properly identified duplicates, over the total set of objects determined as duplicates by the system.

Precision = Relevant results / Retrieved results

Recall measures the percentage of duplicates properly identified by the system, over the total set of duplicate objects.

Recall = Relevant results / Total no. of results

V. Conclusion

We introduced a novel method for XML duplicate detection called XMLDup. The algorithm uses a Bayesian Network to determine the probability of two XML objects being duplicates. In conclusion, to improve the runtime efficiency of XMLDup, and network pruning strategy also exists. XMLDup will show better results both in terms of effectiveness and efficiency by measuring precision and recall values.

References

[1] Lus Leitao, Pavel Calado, and Melanie Hersche, “Efficient and Effective Duplicate Detection in Hierarchical Data,” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,VOL. 25, NO. 5, MAY 2013

[2] F. Naumann and M. Herschel, and V. Ganti, “An Introduction to Duplicate Detection Morgan and Claypool, 2010.

[3] L. Leitao and P. Calado, “Duplicate Detection through Structure Optimization,” Proc. 20th ACM Intl Conf. Information and Knowledge Management, pp. 443-452, 2011.

[4] P. Calado, M. Herschel, and L. Leitao, “An Overview of XML Duplicate Detection Algorithms,” Soft Computing in XML Data Management, Studies in Fuzziness and Soft Computing, vol. 255, pp. 193-224, 2010.

[5] M. Weis and F. Naumann, “Dogmatix Tracks Down Duplicates in XML,” Proc. ACM SIGMOD Conf. Management of Data, pp. 431-442, 2005.

[6] L. Leitao, P. Calado, and M.Weis, “Structure-Based Inference of XML Similarity for Fuzzy Duplicate Detection,” Proc. 16th ACM Intl Conf. Information and Knowledge Management, pp. 293-302, 2007.