Proceedings of
INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION AND ENERGY SYSTEMS
(ICCCES-16)
In Association with IET, UK & Sponsored by TEQIP-II
29th -30th, Jan. 2016

Paper ID: CSEIT02

DUPLICATE DETECTION IN XML DATA USING BAYESIAN NETWORK AND NETWORK PRUNING STRATEGY

Ms. Trupti A. Patil
CSE Dept, ADCET, Ashta, Sangli, Maharashtra
Siddheshwar V. Patil
IT, Dept BIGCE, Kegaon Solapur, Maharashtra
Ms. Swapnali G. Patil
IT Dept, ADCET, Ashta Sangli, Maharashtra
Sadanand S. Howal
CSE, Dept RIT, Islampur Sangli, Maharashtra

Abstract: Data Duplication causes excess use of storage, excess time and inconsistency. Duplicate detection will help to make sure that accurate data is displayed by identifying and preventing identical or parallel records. On identifying duplicates in relational data, an extensive work has been done so far. But only minor solutions are focused on duplicate detection in additional complex hierarchical structures, like XML data. Hierarchical data means a set of data items that are related to each other by hierarchical relationships such as XML. In the world of XML, no automatically consistent and clearly defined structures like tables are available. Duplicate detection has been studied broadly for relational data. For a single relation some methods are used for duplicate detection, they do not directly apply to XML data. So it is required 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 method'. Bayesian network will be used by proposed XMLDup method. It determines the probability of two XML elements being duplicates by considering the information of elements and the structure of information. To improve the Bayesian Network evaluation time, pruning strategy is used. Finally work will be analyzed by measuring accuracy.

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

I. INTRODUCTION

Hierarchical data is a distinct set of data items that are related to each other by hierarchical relationships. In Hierarchical relationships one data item represents the parent of another data item. The structure is represented by parent and child relationships in which each parent can have a number of children, but each child has a single parent.

An XML document is a tree. A single XML data type instance can represent a complete hierarchy. XML is used for interchanging data over internet, Web development and data transport. The features of XML are the sharing of data and simplifying data storage.

Several problems occur in the context of data integration where data from distributed and diverse data source is collected. One of these problems is possible incompatible representation of the same real world object in the different data sources. When combining the data from various sources the perfect result is unique and correct representation of every object such that data quality can only be achieved through data cleansing, where the most important task is to make sure that an object simply has one representation in the result. This requires the detection 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. The Network Pruning Strategy will be presented to improve the efficiency of network evaluation. 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 used to identify duplicates automatically by comparing two records. Well-chosen similarity measures and Well-designed algorithms improve the efficiency and effectiveness of duplicate detection. The algorithms are developed to perform duplicate detection 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 the 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 presented a universal framework for object identification and an XML specific specialization. DogMatiX algorithm was introduced for object identification in XML that uses heuristics to determine the 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 is able to determine the probability of two XML objects in a given database being duplicates. The model also provides greater flexibility in its configuration. It allows 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 the proposed system, the XML files will be selected by the user which contains duplicate records. The XML files will be parsed by using DOM (Document to Object Modelling) API parser. DOM API parses the XML document. After parsing, it maps the records in the file to the Java objects, parse the XML files using the XML Parser API, and get the individual nodes Then Mapped objects will be used to construct BN. It starts with the values in two records which are used for comparison. Bayesian network provides a brief specification of joint probability distribution. For identification of duplicates in the XML structure, a BN is formed. Two XML elements are duplicates, if their values are duplicates as well as child nodes are duplicates. Bayesian Network will have a node labelled with the records 'parent node' and a binary arbitrary variable will be assigned to it. A binary arbitrary variable represents the reality that, if the XML nodes are duplicates it takes the value one otherwise it takes the value zero. Then, parent node will have child nodes in the XML tree and the process of assigning binary arbitrary variables will be continued until the leaf node is reached. A list containing the parent nodes will be given to the XMLDup method for calculating probabilities.

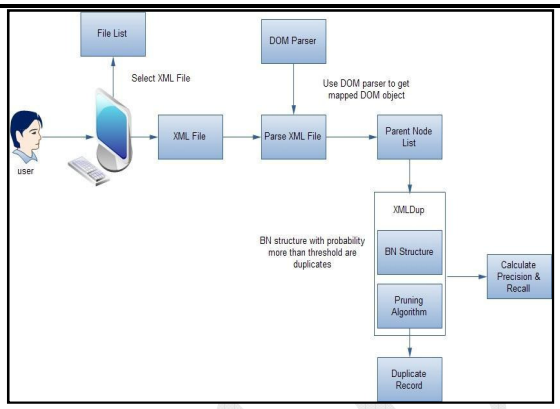

Fig. 1 System Architecture

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 the root probability is found. To calculate root probability that transmits the similarity of leaf nodes up the tree until it reaches to the root node, the probabilities will be used to calculate final probability. The two records in BN structure are duplicates when calculated probability is more than a threshold.

Duplicate records will be separated from the list. The Bayesian Network evaluation time will be improved by the proposed lossless pruning strategy. Pruning algorithm will be implemented by using the list of Bayesian Network models calculated at the start and the Bayesian Network model will be used for evaluation.

Traversing through Bayesian Network Structure, the probabilities of values are transmitted to the parent nodes. If parent's probability is more than threshold value, then records are considered as duplicates in the Bayesian Network structure. Then, the algorithm checks the entire list and finds the duplicates. The speed of the algorithm is increased by removing leading and trailing spaces of values and skips some of the attributes. Precision and Recall values will be measured and compared for both network pruning & improved network pruning algorithm.

IV. METHODOLOGY

The proposed system can be divided into following modules:

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

A. Parsing XML files

The Parsing XML files module, accept the files from user. The XML file contains hierarchical data. XML files are parsed that contains duplicate records. The proposed system uses DOM (Document to Object Modeling) API that parses the

XML document and maps the records to the java objects in the file.

B. Bayesian Network (XMLDup)

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 labelled with the records 'parent node' and a binary arbitrary variable will be assigned to it. A binary arbitrary variable represents the reality that, if the XML nodes are duplicates it takes the value one otherwise it takes the value zero. Then, parent node will have child nodes in the XML tree and the process of assigning binary arbitrary variables will be continued until the leaf node is reached. 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 uses each Bayesian Network model for evaluation. Traversing through Bayesian Network structure, it calculates the probabilities of values to the parent nodes. If the probability of parent is more than threshold value, then records are considered as duplicates in the Bayesian Network structure. Likewise algorithm checks the entire list and finds duplicate. To boost the algorithm, it removes leading and trailing spaces of values and skips some of the attributes to boost speed of the algorithm.

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. EXPERIMENTATION AND RESULTS

The experimentation is carried out keeping in mind the Accuracy Loss. Furthermore, the experimentation is carried out for finding the effect of approximations on accuracy. The programs are implemented in JAVA. The experiment is carried out using Eclipse on single machine with windows 7 operating system.

A. Experimentation Results :

5.1 Parsing XML files:

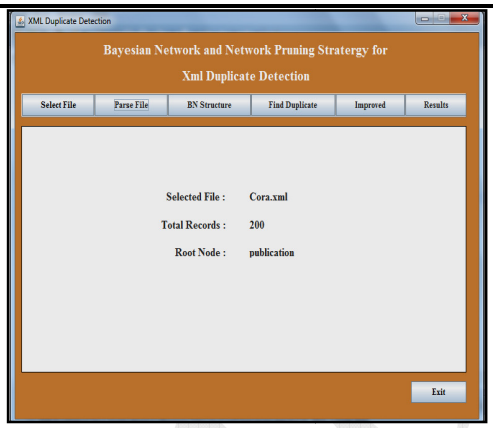

Fig 5.1 Parsing XML file

As shown in Figure 5.1, The XML file is selected by using the select button. The XML file is parsed by using DOM parser. The result displays selected file, total records from selected XML file.

5.2 Bayesian Network (XMLDup):

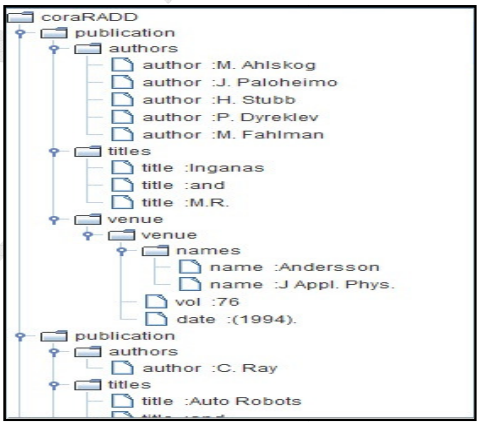

Fig 5.2 Construction of BN Structure

As shown in Figure 5.2, The Mapped objects are used to construct BN. The process starts with the construction of Bayesian Network model. It displays BN structure tree for each parent node.

1. 5.3. Network Pruning :

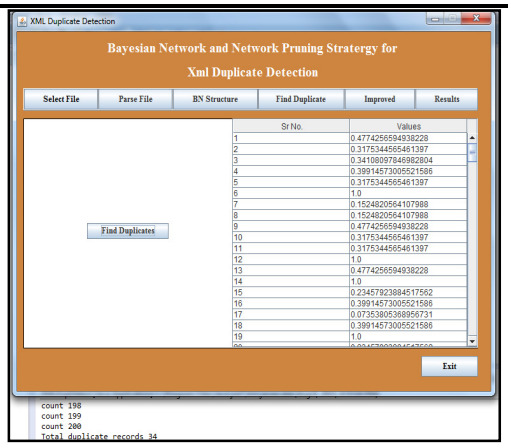

Fig. 5.3 Implementation of Network Pruning Algorithm

As shown in Figure 5.3, The Network Pruning algorithm is implemented by passing it the list of BN models calculated at the beginning. This algorithm takes each BN model for evaluation. Network pruning algorithm finds total duplicate records 38 from Cora.xml datasets.

5.4. Improved Network Pruning:

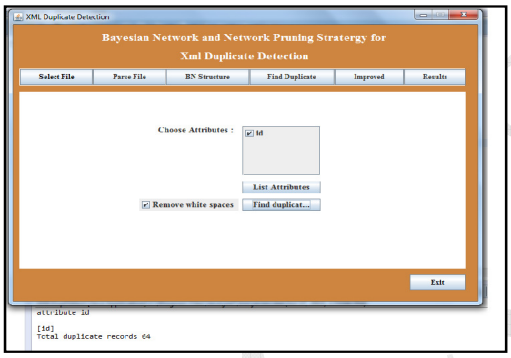

Fig. 5.4 Implementation of Improved Network Pruning Algorithm

As shown in Figure 5.4. The XMLDup algorithm considers all the attributes of each node. But in our system, instead of taking all attributes for comparison, we can consider a limited number of attribute and leave another. So it will give better performance than previous. Any attribute value containing any trailing or leading white space, we first remove the white spaces and then perform the comparison.

5.5 Testing of System:

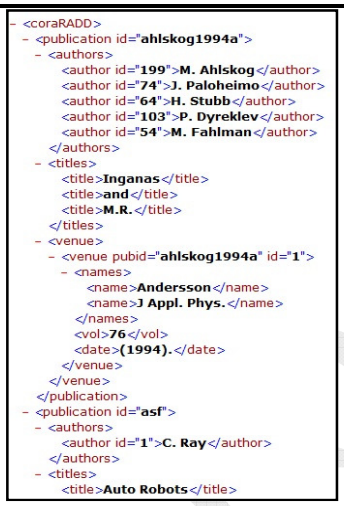

Fig. 5.5 Cora.xml Datasets

As shown in Figure 5.5. This design shows Cora xml data set.

5.6. Analysis of Work:

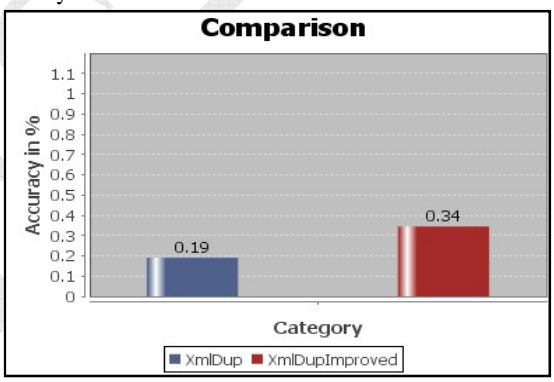

Fig. 5.6 Calculating Accuracy

As shown in Figure 5.6, after all the models were built, the work is analyzed by measuring the accuracy of both network pruning & improved network pruning algorithm. It gives better result than previous algorithms by using improved algorithm.

VI CONCLUSION

We introduced a novel method for XML duplicate detection. 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, network pruning strategy is used. The speed of the algorithm is increased by implementing Improved Network Pruning algorithm. The XMLDup will show better results both in terms of effectiveness and efficiency by measuring accuracy.

ACKNOWLEDGMENT

I would like to express my deep sense of gratitude towards my guide Prof. S. V. Patil, Annasaheb Dange College of Engineering, Ashta Sangli for his invaluable help and guidance for the project. I am highly indebted to them for constantly encouraging me by giving critics on my work. I am grateful to them for having given me the support and confidence.

REFERENCES

[1] Lus Leitao, Pavel Calado, and Melanie Herschel, "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, and 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