RESEARCH ARTICLE OPEN ACCESS

Duplicate Record Detection in XML using AI Techniques

Anagha Pradeep[1], Teena George[2]
[1]Computer Science and Engineering, MG University, ASIET, Kerala, India)
[2]Computer Science and Engineering MG University, ASIET, Kerala, India)

----------**********----------

Abstract:

Duplicate detection multiple representation of same entity. XML is widely used in almost all applications especially data in web. Due to the wide usage of XML it is essential to identify duplicates in it. Various methods like normalization etc are used for duplicate detection in relational database but it cannot be employed in XML due to its complex structure. Detecting and eliminating duplicates correctly has become one of the challenging issues in the areas of places where data integration is performed. Many techniques have been emerged for detecting duplicates in both relational databases and XML data's. By recognizing and eliminating duplicates in XML data could be the solution, for this a strategy based on Bayesian Network called XMLDup to detect duplicates and use machine learning algorithm like SVM, Bee, Bat algorithms for improving its efficiency and compare them to find out the most efficient method to find out duplicates in XML effectively.

Keywords — ***Bayesian Network, DELPHI, dogmatiX, duplicate detection, network pruning, SXNM, XML, XMLDup, Bee, Bat***

----------**********----------

I. INTRODUCTION

Duplicate detection is the major important task in the data mining, for finding duplicate in the data objects. Its purpose is to identify whether the given data is duplicates or not. Real world duplicates are multiple representations of same real world data object. Detection of duplicates can performed in many places, its major important in database.

Duplicate detection is the major important task to determining dissimilar representation of XML data for real world object. Duplicate detection is a essential process in data cleaning and is significant for data integration, individual data management, and several areas. In case of relational database we can use normalization and other conservative approaches can be used but in case of XML due to its complex structure it cannot be applicable. The difficulty of XML duplicate detection is mainly tackling in applications like catalog integration or online data process.

There are different techniques available for identifying duplicates in XML data such as Duplicate object get matched in XML (DogmatiX), XMLDup, , network pruning, NM similarity, SXNM and XML document Integration (XdoI) etc. XMLDup and dogmatiX are good approaches. In these approaches Dogmatix and XMLDup are important approaches where they show most effectiveness and efficiency, for calculating these we use precision and recall in order to analyze them. When large datasets are given there are chances that information not relevant for comparisons will be considered while detecting duplicates. In order to overcome this drawback network pruning has been introduced. The advantage of network pruning is it improves Bayesian network evaluation time. One disadvantage of network pruning is sometimes it will not detect some duplicates. In this paper different technique for detecting duplicates in XML data has been studied and it also compares the efficiency of different techniques in identifying duplicates. Figure1. shows an example for duplicates in real world entity

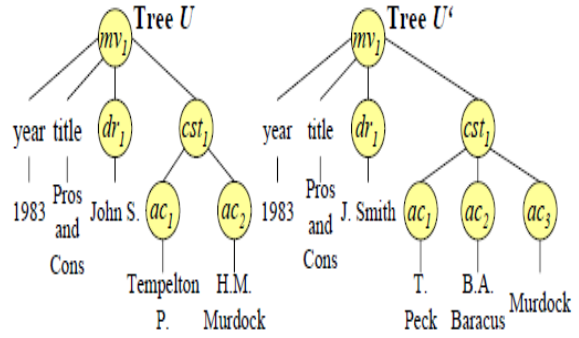

Fig 1: Two XML trees, each representing a movie (*mv*) nesting directors (*dr*), a cast (*cst*) and actors (*ac*) as shown in [3]

In this paper we mainly focus on XMLDup method in detail by using Bayesian network and network pruning method for increasing efficiency and our contribution is to use artificial intelligence methods for further efficiency.

II. STATE OF ART

A. *Methods for Duplicate Detection in XML*

Duplicate record detection is one of the major problem faced in data warehouses during data integration, in order to resolving this various approaches where introduced. They may be either top down or bottom up category. All algorithms that have been developed for XML duplicate detection fall in the category of iterative duplicate detection algorithms. A characteristic of algorithms in this class is that they use a measure that computes a similarity score between two object representations. If the similarity is above a predefined threshold, the pair of object representations is classified as a duplicate.XML duplicate detection algorithms is on the format of algorithms that operate on tree data. First use XML joining operations for these later in 2002 Dogmatix where proposed which is an top up approach have high values for precision and recall. XMLDup is similar to Dogmatix is most effective in low recall values . For increasing its efficiency so many studies were carried out among these network pruning is most important. In network pruning it reduces the number of comparisons. In this paper it compare few important methods and focus on their drawbacks.

1) *Delphi Algorithm*

R. Ananthakrishna S. Chaudhuri and V. Ganti proposed Delphi approach [4] for eliminating duplicates in dimensional tables represented hierarchically in the data warehouse. The authors exploit the dimensional hierarchies associated with the tables stored in data warehouse. The algorithm proves to be efficient and scalable which significantly reduces the number of false positives without missing out on detecting duplicates. Algorithm is based on top down approach. Most of the earlier address the problem of a single relation where tuple represents an object and duplicate record detection is performed combing the object. It is used for 1:N relationship between referenced table (parent) and referencing table(child) eg: state and cities. It starts with parent then goes to child. A dimensional hierarchy consists of a chain of relations linked by key—foreign key dependencies.

2) *Dogmatix Framework*

M. Weis and F. Naumann proposed DogmatiX track down approach [2] for identifying duplicates in XML data. DELPHI uses non symmetrical measure which doesn't compare difference of two elements. DogmatiX overcome the drawback of DELPHI by considering the symmetrical measure which takes into account the difference between the elements. DogmatiX, where Duplicate objects get matched in XML. DogmatiX algorithm for object identification in XML. This algorithm takes an XML document, its XML Schema S, and a file describing a mapping M of element XPaths to a real world type T as input. The type mapping format is (name of the real-world type, set of schema elements). DogmatiX is rendered domain-independent in its description selection by using specialized heuristics. It specializes our framework and successfully overcomes the problems of object definition and structural diversity. The framework consists of three steps of candidate definition , duplicate definition and duplicate detection phase. In candidate definition phase the result be the set of all XML elements that represents actors, duplicate

definition phase it determines the elements that are used for our comparison and finally in duplicate detection phase based on the similarity calculation duplicates are determined. It is used for 1:N relationship.

3) SXNM (Sorted XML Neighborhood Method)

S. Puhlmann, M. Weis, and F. Naumann proposed [6] XML Duplicate Detection Using Sorted Neighborhoods, It is bottom up approach.It solve the problem of M:N relationship. For each object definition if develop a key. For example for an object U and object description {"pros and cons",1983} the key will be PR083. Then sort it in dictionary order. The main defect of this approach is it compare candidates whose key in a fixed size window, It decrease performance when typographical error occurs. Typographical errors in key attribute values can cause similar objects to be places in far away positions and thus never be reached by the comparison window.

4) XMLDup

M. Weis, L. Leitao and P. Calado proposed Bayesian Network to improve the performance [3]. Duplicate detection is performed on hierarchical and semi-structured XML data. Probabilities are computed using Bayesian Network, which is a directed acyclic graph. In this, authors considered both prior and conditional probability values. Prior probability is associated with the leaf node and conditional probability with inner nodes in the network. In this four conditional probabilities are considered, based on which a node is identified as duplicate or non-duplicate. Since, probability of a node being duplicated is not known in advance prior probability is assigned to each parent node. Probability of a node being duplicated is calculated using conditional probability. In conditional probability a node is considered as duplicate if its value nodes are duplicates. A parent node is considered duplicate if all its child nodes are duplicates. In short a node is considered as optimal overlay. Computing the cost involves comparing string values using edit distance.

6) Network Pruning in XMLDup

duplicate is the calculated probability exceeds the prior probability value assigned to a node otherwise it is considered as non- duplicate. All node values are considered as textual string and probability is calculated using a similarity function which uses edit distance. This method proves to be highly flexible but it is not scalable both in time and space. This method gives high recall and precision values.

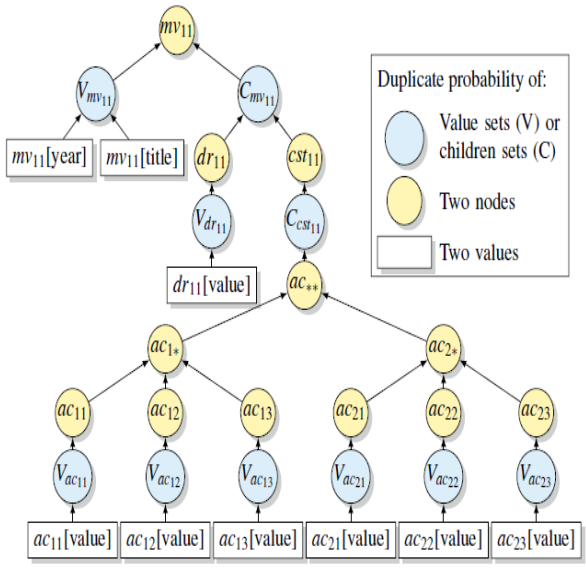

Fig 2. shows the bayesian network of movie dataset as shown in figure1.

5) A Structure-Aware XML Distance Measure

Milano et al.[5] propose a distance measure between two XML candidates that takes into account both their structure and their data values. As is common to all iterative duplicate detection algorithms, this measure is used to perform a pairwise comparison between all candidates. If the distance measure determines that two XML candidates are closer than a given threshold, the pair is classified as a duplicate. Based on overlays that represent a 1:1 matching of XML nodes. The similarity is computed based on the cost of an

In order to improve BN evaluation time network pruning is proposed [8]. It is flexible enough to handle large datasets. Since it performs well on large dataset the problem of DogmatiX was overcome. In this method Bayesian Network is

developed and is evaluated using prior and conditional probabilities. Four types of conditional probabilities are taken for determining the duplicates. Prior probabilities are calculated using similarity function which is normalized to fit between 0 and 1. Network pruning is employed to accelerate the Bayesian Network evaluation. A lossless pruning strategy is used which ensures that no duplicates are missed out.

This method delivers a high degree of recall and precision. Detecting duplicates using this method saves lot of time there by increasing its efficiency in detecting duplicates. Network pruning saves the time spends on finding the correct matched pairs there by eliminating the drawback of other methods. In this work, the author propose three such heuristics: sorting by depth, by average string size, and by distinctiveness.There are so many features used for pruning they are

Format features. Features that provide information about the type of contents in attributes values. These are the ratio of attributes that contain numeric values, alphabetical values, and both.

Content length features. Since we use a string edit distance to compare attribute values, and given that the outcome of this measure is strongly related to the size of the strings, this group contains the features average string size and entropy of the string sizes.

Absence features. This group contains only one feature that measures how many objects are missing the given attribute. Attributes that are missing in many objects should probably be taken less into account.

TABLE I
ADVANTAGES AND DISADVANTAGES

Method	Advantages	Disadvantages	Complexity	Traversal
DELPHI algorithm	solving the problems of object definition and structural heterogeneity inherent to XML data. Simplest method	Only applies to 1:N relationship. It cannot consider structure	$O(n^2)$	Top down
SXNM – The Sorted XML Neighborhood Method	Can be used in M:N relationship and 1:M	High number of pairwise comparisons compromises efficiency. Cannot detect typographical error	O(nlogn)	Bottom up
DOGMATIX Framework	High precision	Not good when dataset is too small or too large	$O(n^2)$	Top down
Bayesian network XMLDup Algorithm	Reduce no of comparisons compared to early approaches. A high quality, scalable, and efficient algorithm.	Not focused on run time efficiency	$O(n^2)$	Bottom up
concept of overlays	The similarity is computed based on the cost of an optimal overlay.	Only suited XML elements having 1:N relationship	$O(n^2)$	Top down
Network Pruning in XMLDup	It increase efficiency and effectiveness by using a network pruning algorithm	Conditional probability values have to be derived manually	$O(n^2)$	Bottom up

III. METHODOLOGY

In this paper a hybrid technique is used for detecting duplicates in hierarchically structured XML data. Most aggressive machine learning techniques and swarm intelligence techniques are used to derive the conditional probabilities for all new structure entered. A method known as binning technique is used to convert the outputs of these into accurate posterior probabilities. To improve the rate of duplicate detection network pruning is also employed. Through experimental analysis it is shown that the proposed work yields a high rate of duplicates thereby achieving an improvement in the value of precision. This method outperforms other duplicate detection solution in terms of effectiveness.

Deriving Conditional Probabilities Using SVM

A XML document is considered as duplicates based on the conditional probability. Using SVM probabilities of different structure can be calculated efficiently. While applying SVM, conditional probability is obtained as the output which is converted into an accurate posterior probability using binning [9], [10].

While using SVM, the dataset was divided into two: training and testing sets. SVMs learn a decision boundary between two classes by mapping the training examples onto a higher dimensional space and then determining the best separating hyper plane between those spaces [9]. Given a test example „x", the SVM outputs a score that measures the distance of „x" from the separating hyper plane.

Modified Bat Algorithm for identifying duplicate

Most microbats(with basic attribute) are insectivores. Microbats use a produce of sonar, called, echolocation, to detect pre(record), avoid obstacles, and locate their roosting crevices in the dark. These bats emit a very loud sound pulse and listen for the echo that bounces back from the surrounding things. Their pulses change in properties and can be linked with their hunting strategies, depending on the type. Most bats use small, frequency-modulated signals to sweep through about an octave, while others more often use constant-frequency signals for echosound. Their signal bandwidth varies depends on the species, and often increased by using more harmonics.

By idealizing some of the echolsound characteristics of micro-bats(small keys), we can develop various bat-inspired algorithms or bat algorithms. Here developed Modified Bat Algorithm with Doppler Effect.

In the proposed system presents an approach new modified bat algorithm to overcome the difficulty and complexity of the later Approaches .The new algorithm finds the best optimization solution for random selection of the input values and removes the duplicate records in the system. The algorithm reduces the number of the steps.

Optimization is nothing but selection of a best element from some set of available objects. An optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function.

In the proposed system, we implement the IBAT (Modified Bat) which is a metaheuristic algorithm that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

The Modified Bat Algorithm is based on the echolocation behavior of micro-bats with varying pulse rates of emission and loudness with Doppler Effect.

Bee Algorithm for Duplicate detection

Tabu Artificial Bee Colony Algorithm repeatedly builds the solutions for the given problem. The intermediate solutions are assigned as solution states. At each iteration of the algorithm, each Bee moves from the position X to Y. For this movement, all the local search of the records has been done using the Tabu search algorithm. After getting the optimization result of the duplicate records, next iteration starts with the global search of duplicate records by using the Artificial Bee Colony algorithm. Duplicate records can be easily identified using Tabu Artificial Bee Colony Algorithm which is combination of meta heuristic algorithms.

IV. PERFORMANCE EVALUATION

For checking accuracy we use
precision = tp/(tp+fp)
recall = tp/(tp+fn)
tp : Correctly identified duplicates
fp: Falsely identified duplicates
fn: Number of duplicate nodes identified as non duplicates

V. CONCLUTION

In this paper, we evaluate different techniques for finding duplicate records in XML document, here a machine learning algorithm known as SVM is proposed for deriving conditional probabilities for the detection of duplicates and a technique known as binning is used to convert the output of SVM to an accurate posterior probability. Estimating the probability using SVM increases the rate of duplicate detection. SVM not only consider contents but it also takes into account XML objects with different structures. The proposed method achieves an improvement in the value of precision on different structured data. After that by using bat and bee algorithms for getting more efficiency.

REFERENCES

[1] P. Calado, M. Herschel, and L. Leita˜o, "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.

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

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

[4] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. "Eliminating fuzzy duplicates in data warehouses," In International Conference on Very Large Databases, Hong Kong, China, 2002.

[5] D. Milano, M. Scannapieco, and T. Catarci, "Structure Aware XML Object Identification," Proc. VLDB Workshop Clean Databases (CleanDB), 2006.

[6] S. Puhlmann, M. Weis, and F. Naumann, "XML Duplicate Detection Using Sorted Neighborhoods," SPRINGER ,Proc. Conf. Extending Database Technology (EDBT), pp. 773-791, 2006.

[7] B. Zadrozny and C. Elkan. "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers". Proceedings of the Eighteenth International Conference on Machine Learning, 2001.

[8] M. Weis, L. Leitao and P. Calado "Efficient and Effective Duplicate Detection in Hierarchical Data," IEEE Transactions on Knowledge and Data Engineering, Vol. 25, May 2013.

[9] J. Drish "Obtaining calibrated probability estimates from support vector classifiers"

[10] J.T. Kwok "Moderating the Outputs of Support Vector Machine Classifiers". In IEEE -NN,1995