A Survey on Various Methods Used for Detecting Duplicates in XML Data

Anju Ann Abraham
PG Student
Department of Computer Science and Technology, Karunya University
India

S. Deepa Kanmani
Assistant Professor
Department of Computer Science and Technology, Karunya University
India

Abstract

Duplicate detection is the process of identifying multiple representations of same real world entity. eXtensible Mark-up Language is widely used in almost all applications especially data in web. Due to the wide usage of eXtensible Mark-up Language it is essential to identify duplicates in XML. Eliminating duplicates correctly has become one of the challenging issues in the areas of Customer Relationship Management (CRM) and other place where data integration is performed. Many techniques have been developed for detecting duplicates in both relational and XML data's. Different methods are used for detecting duplicates in XML data which is represented in different formats. The efficiency will vary depending upon different methodology. This paper made a survey for detecting duplicates. The purpose of this paper is to provide a survey on different methods used for detecting duplicates in XML data.

Keywords— *Bayesian Network, DELPHI, dogmatiX, duplicate detection, network pruning, NM similarity, XdoI, XML, XMLDup.*

1. Introduction

Data mining is the process of extraction of information from a given data source. It is useful in anomaly detection, classification, clustering and summarization etc. Employing data mining in duplicate detection will help in classifying the duplicates and non-duplicates from a given dataset.

eXtensibleMarkup Language (XML) have been widely used in e-business and web for data exchange. Now-a-days the World Wide Web is a huge platform for data and this data is created, stored and transferred. XML documents are one of the best tools for representing and transferring data because of their flexibility and self-description. The main applications of duplicate detection are in the field of customer relationship management (CRM), data integration etc. In case of CRM, if many entries of a person are present it may lead to multiple mailing to same person which create a way for, incorrect aggregation of sales to some customers. Consider an example, a person named Angelin Julie purchased a product from a shop and details of her entries are made by the retailer as A. Julie, New York, USA. Another day while she purchased another product an entry is made as Angeline Julie, NY, United States of America. In both the cases the person is same but they are represented in different way. This multiple representation of the same entity will lead to multiple mailing to the same person which led to incorrect sales. The purpose of detecting duplicates is an essential in this type of problems. Duplicate detection has been performed more frequently in data which is stored in a table [1]. 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 and XML document Integration (XdoI) etc. XMLDup and dogmatiX are good when the information is small. 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.

2. Methodologies

Various methods are used for identifying duplicates in XML data. Among those few methods have studied and their performances are compared.

2.1. Delphi

R. Ananthakrishna S. Chaudhuri and V. Ganti proposed Delphi approach [2] 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. It significantly reduces the number of false positives that are

retrieved during duplicate detection. Dimensional hierarchies are exploited to find out the co-occurrence among the tuples for detecting equivalence error and to reduce the number of false positives. A threshold similarity function is used to define duplicate detection [3]. Textual similarity is computed by dividing the tuples into tokens using some tokenization function. Co-occurrence between the two distinct tuples is determined by the amount of overlap between two children set tuples. Textual similarity is measured using token containment metric and co-occurrence similarity is measured with the help of foreign key containment metric. If the textual or co-occurrence similarity lies above the threshold value then it is considered as duplicates. To reduce the number of pair wise comparison among the tuples a potential duplicate identification filter is used, it will compare only those pair which seems to be duplicate. The main drawback of this method is that it will not compare all pairs of tuples in the hierarchy.

2.2. DogmatiX

M. Weis and F. Naumann proposed DogmatiX track down approach [4] 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 takes XML document; its schema and a file which describes mapping of elements are taken as input and it produce clusters of duplicate object with its identifier as output. One of the closest approaches to dogmatiX is DELPHI. The framework consists of three components: candidate definition which decides the elements for comparison, duplicate definition which specify which of the two candidates compared are duplicates, and duplicate detection which define how to detect duplicates based on the two previous components. Duplicate definition is defined by description element and description selection, in which description selection uses two heuristics called r-distant ancestor and r-distant descendant, where r is depth. After the description instance has been generated, classification is performed to separate the duplicates and non-duplicates. Pruning is performed if the dataset is large, in order the number of candidate element taken for comparison. The process of pruning is supported by blocking and filtering. Filtering prunes out elements that are not duplicates and in blocking elements that are likely to be duplicates are clustered together for comparison. Edit distance and inverse document frequency is used to calculate the similarity between to elements. Effectiveness of this method in identifying duplicates is high when neither too few or nor too much information is selected. This method yields low precision and high recall values.

2.3. XMLDup

M. Weis, L. Leitao and P. Calado proposed Bayesian Network to improve the performance [5]. 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 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.

2.4. NM similarity

Z. Na, Z. Dongzhan, Y. Ye and D. Jiangjiao proposed NM similarity to compute the similarity between XML nodes [6]. This method is used to compute both content and structure similarity. Content similarity is evaluated using cosine measure and structure similarity is evaluated using Euclidean distance [7]. While evaluating structure similarity only those elements with most important properties will be used for matching. During content similarity detection stop words, punctuation and stems are removed from the XML documents. NM similarity is applied to K - Nearest Neighbour machine learning algorithm for classifying the document based on similarity. Performance is evaluated using recall and precision, where recall is the measure of completeness and precision is the measure of exactness. Since no pruning is employed in this method it shows a high accuracy because pruning will miss out some important elements from the document but main disadvantage is that only similar structure document will be considered as duplicates.

2.5. XdoI

W. Viyanon, S. K. Madria, and S. S. Bhowmick proposed XML Document Integration (XdoI) for detecting the duplicates [8] so that data integration

can be made easily without any ambiguities. XdoI considers both structure and content while finding the similarity between two XML documents. This method performs matching in bottom-up fashion using XML keys which reduces the number of comparison being made. XdoI is composed of three stages. In stage1, by taking leaf node parent as clustering point group the base XML tree and target XML tree. In stage2, both the XML sub trees are considered as an independent item and are matched with each other. In stage3, the best matched sub trees are integrated with the first XML tree. The steps involved in integration are: first, find the similarity degree of each pair of sub trees and if there exist more than one matched sub tree the maximum similarity degree is taken then in second step, tree similarity degree is calculated and finally if the tree similarity degree is greater than the threshold value then two documents can be integrated to cluster point. In this method XML document is pre-processed by parsing and then stores the document in relational table based on their structure. For all distinct node leaf nodes matching is performed using Structured Query Language (SQL). This distinct node is used as key. Main drawback of the paper is time spend in finding the keys for sub tree matching. This method even though reduces the number of false positive it consumes more time in detecting similarities among documents.

2.6. Network Pruning

In order to improve BN evaluation time network pruning is proposed [9]. 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. The algorithm computes pruning factor automatically using stimulated annealing [10] depending on the data and those nodes which falls below the threshold value are discarded. 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.

3. Comparison of Various Duplicate Detections Methods

Various methods are used for detecting duplicates in XML data. Each method has their own merits and demerits. Various methods are compared with each other to know their advantages and disadvantages. The comparison is shown in table given below.

Table 1 Comparison of Various Duplicate Detections Methods

Algorithm Used	Advantages	Disadvantages
DELPHI	Efficient and scalable, less false positives	Avoid some tuples from comparison
DogmatiX	High precision	Not good when dataset is too small or too large
XMLDup	Highly flexible, high recall and precision	Not scalable in terms of space and time
NM similarity	High accuracy	Similar structure document will be considered
XdoI	Less false positives	Consumes time in identifying keys
Network Pruning	Flexible, high recall and precision and consume less time	Pruning may miss out some duplicates

The comparison table shown above uses performance metrics to evaluate various methods. The various performance metrics used are recall, precision, false positives etc. From the comparative study performed network pruning is more flexible as it can be applied on both large and small datasets. Even though network pruning miss out some duplicates it is used in most applications.

4. Conclusion

In this paper various algorithms for detecting duplicates in XML data have been compared. The paper discusses six algorithms for detecting duplicates in XML data. Bayesian network using network pruning perform better on large dataset and it is scalable in terms of time. It also gives high precision and recall values there by increasing the accuracy of duplicate detection. From the studies were made, it has been observed that network pruning outperforms other methods.

5. References

[1]F. Naumann and M. Herschel, "An Introduction to Duplicate Detection," Morgan and Claypool, 2010.

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

[3]M. Hernandez and S. Stolfo. "The merge/purge problem for large databases," In Proceedings of the ACM SIGMOD, pp 127-138, San Jose, CA, May 1995.

[4] M. Weis and F. Naumann" DogmatiX Tracks down Duplicates in XML," ACM SIGMOD Conf. Management of data, pp. 431-442, June 2005.

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

[6] Z. Na, Z. Dongzhan, Y. Ye and D. Jiangjiao "Proceedings of the Third International Symposium on Computer Science and Computational Technology," pp. 426-430, August 2010.

[7] G. An and L. Huashan, "Improved Algorithm of XML Document Structural Clustering Based on Edit Distance," Microcomputer Applications, vol 29, pp. 88-91, 2008.

[8] W. Viyanon, S. K. Madria, and S. S. Bhowmick" XML Data Integration Based on Content and Structure Similarity Using Keys," Springer-Verlag Berlin Heidelberg 2008, pp. 484–493.

[9] 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.

[10]S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by Simulated Annealing," Science, vol. 220, pp. 671-680, 1983