IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 6, Ver. I (Nov – Dec. 2015), PP 69-76
www.iosrjournals.org

Duplicate Detection in Hierarchical Data Using XPath

[1]Akash R. Petkar, [2]Vijay B. Patil
[1,2]Computer Science &Engineering, MIT Aurangabad.

***Abstract**: There were many techniques for identifying duplicates in relational data, but only a few solutions focus on identifying duplicates which has complex hierarchical structure, as XML data. In this paper, we present a new technique for identifying XML duplicates, so-called XML duplication using Xpath. XML duplication using Xpath technique uses a Bayesian network to conclude the possibility that two xml elements are duplicates, based on the information within the elements and other information organized in the XML. In addition, to increase the proficiency of the web usage, a new pruning strategy was created. This pruning strategy will help to gain maximum benefits over non-computing algorithm. This technique can be used to increase the proficiency of identifying duplicates and remove it, so no duplicate record will be there. Through many experiments, our algorithm is able to achieve high accuracy and retrieve count in several XML dataset. XML duplication using Xpath technique is able to outclass another technique for identifying duplicates, both in proficiency and potency.*

***Keywords:** Identifying duplicates, XML, Bayesian network, object cleaning, hierarchical structure, Xpath.*

I. Introduction

Electronic data plays an important role in today's world for business processes, applications and making quick decisions. As we are focusing on how the data can be essential and we have to compromise on different types of errors which come in different representations [1]. In this paper we are focusing on different types of errors that can be occurred in Data. We will mainly focus on fuzzy duplicates or duplicate records. Duplicate records are multiple representation of same real world object that are differently represented. These records are somewhat different from each other. These records attributes differ in some way from each other in XML document.

Duplicate detection means finding out these different representations of same real world object. Duplicate detection is a tough task to find duplicate records. The common comparisons algorithm to find the duplicates cannot be used, so finddifferent possibly matching strategyto compare, so that they are referring to the same object or not

In this paper, the focusis on which possibly matching strategy can find out to detect duplicate records. The Focus should be able to match the different representation of information at a conceptual level. Take xml dataset for comparing different possibilities. An XML dataset or document includes a set of nodes in the document. It consists of root node and child nodes. It starts with an opening tag (Ex. <A>) and a closing tag(Ex. </A>).

```
<Customer>
        <Country>Australia</country>
</Customer>
<Customer>
        <Country>AUS</country>
</Customer>
```

Figure 1: Attribute Scope

In Fig.1two records are shown for two different XML records. But both records are representing the same country so there can be possibility that in XML dataset or document it can be present. So find different such possibilities that represent the same object. Duplicate records are exactly same by textual information. But if they are slightly changed; the information are not exactly duplicates.

Another problem is that XML can be presented in different structures so the possibility of finding the duplicates becomes high. An XML document contains one root element and number of child element, but child element can also have different child elements and so on. In this paper, a novel technique is presented which can be used to detect XML duplication of same real world object.

Rest of the paper is organized as follows: Section II the related work is discussed. Section III describes the proposed work. Section IV describes the mathematical model. Section Vpresents the performance analysis. Section VI shows graphical model. Section VII concludes the paper.

II. Related Work

Data cleaning [2] or*data cleansing* deals with identifying and removing errors, irregularities from data are represented in such a way that it can degrade the quality of data.Dataquality problems are there in single data files such as databases or XML File ex. Due to misspelling during entry of data, invalid data or some missing information. When integrating multiple file data to integrate into one file then it should be able to get a single file that is free from duplicate records.

2.1 Eliminating Fuzzy Duplicates in Data Warehouses

In *data warehouses* large databases are integrated ex. global web-based information system, merged database systems so there can be a possibility of duplicate records. The need for data cleaning process becomes an important factor for getting the accurate records with no duplication. The duplicate records or redundant records are represented in different representations. In order to get the data accurate and consistent, joining different types of data and merging into one data and eliminate the duplicate data becomes a necessary step for faster processing of data. Firstly find out the different possibilities of data that can be represented in different structure so that multiple documents when combine together will form a single document which is free from duplicate records.

For Example in Table 1 there are records that consist of*f_n* (*firstname*), *country* and *email*. Record *r1* and *r2* are exact duplicates because their *f_n, country and email* values are same so we can say they are exact (100 %) duplicates, so we can easily say they are duplicates and can be easily removed. Record *r1* and *r3* are not exact duplicates because their*f_n* and *email* values are same, but not *country* values are same. But in record *r3* the country values is denoted in different format, but it refers to same record. Therefore, find such duplicates. These duplicates are called as *fuzzy duplicates [3]*.

Table 1: Exact Duplicates as well as Fuzzy Duplicates

Record	f_n	country	Email
r1	John	Australia	john@gmail.com
r2	John	Australia	john@gmail.com
r3	John	AUS	john@gmail.com

2.2 DogmatiX Tracks down Duplicates in XML

Inthis,dogmatix defines a general framework for identifying the duplicates. In this, records are checked whether they are duplicates or not based on their values. In real world, records are represented in multiple patterns for same object. The *dogmatix framework*is flexible to work on different algorithms and new methods can be added to improve this framework. An overview of the framework is given in [5].Theframework consists of three types,

- **Candidate definition:** Defines which document should be compare.
- **Duplicate definition:** Defines when two objects are duplicates.
- **Duplicate detection:** Defines How Duplicates are searched.

III. Proposed Work

3.1 XML Duplication Using XPath

Every tuple in a relational table has exactly one value for every attribute. Most duplicate detection approaches designed for a single relation iteratively compare pairs of tuples as follows: They first compare attribute values pair wisely by computing a value similarity, and then combine these similarities to a total tuple similarity. If the similarity is above a specified threshold, tuple pairs represent duplicates, otherwise they represent non-duplicates. This comparison approach is called a threshold similarity measure approach.

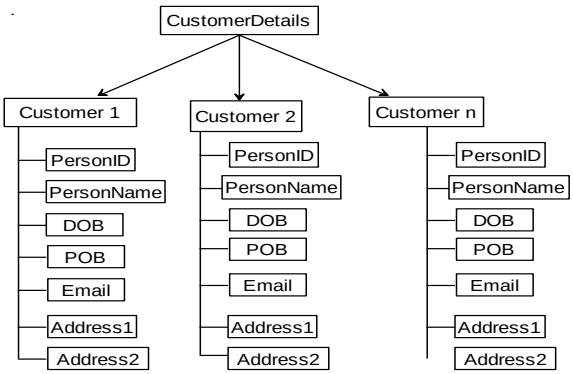

Figure 2: CustomerDetails XML

DTD are a strict representation or set of rules for XML.XML can be represented by a tree structure as shown in Fig. 2.In this *customerdetails,* is the root element and *customer* are the child elements. The child elements have 7 attributes as *personid, personname, dob, pob, Email, Address1, and Address2*.

Today, XML is used in many web applications. The popularity of xml has increased because of its platform independent, less space and easy to use ability. XML is mostly used for data storage and fast transfer of data.

3.2 Different Types of Parsers

On web there are various types of parsers are available for parsing the XML document. Some of them are good in some features and some of them are not. In Table 2 different types of parser are used for *parsing*, but some parser are design for read only access. But in this paper, a novel approach is proposed for parsing the XML file and finding the exact duplicates or fuzzy duplicates and removing the duplicates, so that pure XML document must be formed.*STAX* parser, *SAX* parser, *DOM* parser etc. are used for parsing. In Table 2 how these parsers differ from each other.The notation for the following Table 2 can be used as *Xpath Capability = XC, CPU & Memory = C&M, Forward only = FO, Read xml = RXML,Write xml = WXML, create,read,update,delete = crud.*

Table 2: Features Table

Feature	STAX	SAX	DOM
API Type	Pull,Streaming	Push, Streaming	In memory tree
XC	No	No	yes
C&M	Good	Good	Varies
FO	Yes	Yes	No
RXML	Yes	Yes	Yes
WXML	Yes	Yes	Yes
CRUD	No	No	Yes

3.3 XMLDOM Parser

The structure of Dom parser can be identified using the following diagram.

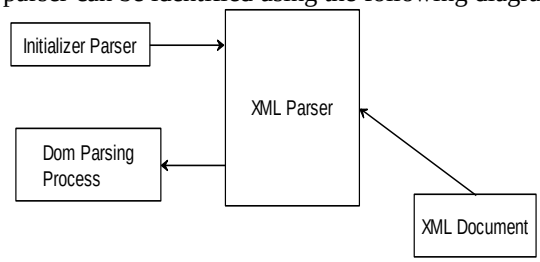

Figure 3: XML Dom Parser Structure

DOM, also known as *document object model*. It is mostly used for XML operation today. The responsibility of a DOM parser is to read the XML document specified and convert that into a tree structure suitable for traversal. Internally, a DOM parser takes help from SAX parser to read the file and the compares the XML against the *DTD* or the *schema*, so that relationships between parent-child tags can be set up and the tag tree is built into the memory. *DOM* first copies XML into memory before parsing It, So It is a good advice to have large heap size to avoid exceptions.

3.4 XML Duplication Using Xpath Algorithm

In this, we describe how the algorithm works and how it is able to detect duplicates in XML dataset and remove the duplicates from the XML.

Steps in xml duplication using XPath algorithm,
Begin;
1) Read the xml file and get the ordered list of parent nodes (**L**).
2) Read the child nodes of xml file.
3) Current score = 0
4)For each node n in L do
5)If (attributes of child nodes= threshold value) then
Completely duplicate
Else

No duplicate
6) Remove all duplicate nodes and save the new xml file.

Table 3: Results for Customer, Suplier and Order Dataset

Dataset	Customer	Suplier	Order
Records	1000	1000	1000
Time Depth Search For Duplicate	2973 Milliseconds	3000 Milliseconds	2933 Milliseconds
Sorted Set	2873 Milliseconds	2876 Milliseconds	2872 Milliseconds
Deduplication For The Duplicate Record	91 Milliseconds	130 Milliseconds	59 Milliseconds
Total Time	1532 Milliseconds	1565 Milliseconds	1496 Milliseconds

IV. Mathematical Model

1. Identify XML records R

$$R = \{r1, r2, r3, r4, r4, r5, \dots \dots .\}$$

Where *R* is main set of records

2. Then Identify Nodes of each record N

$$N = \{n1, n2, n3, n4, r5, \dots \dots .\}$$

Where N is main set of nodes for a record

3. Probability that a node is duplicate

$$P = \{p1, p2, p3, p4, p5, \dots \dots .\}$$

$$P(t_{ij} | Vt_{ij}, Ct_{ij}) = \begin{cases} 1 \; if Vt_{ij} = Ct_{ij} = 1 \\ 0 \qquad Otherwise \end{cases}$$

Where P is main set of probability
If $V_{tij} = C_{tij} = ThresholdValue$ Then It Is Duplicate

4. Identify Duplicate nodes DN

$$DN = \{dn1, dn2, dn3, dn4, dn5, \dots \dots .\}$$

Where DN is main set of the duplicate nodes

5. Identify Duplicate records for DR

$$DR = \{dr1, dr2, dr3, dr4, dr5, \dots \dots .\}$$

Where DR is main set of the duplicate records

6. Calculating Total Time for Duplication

$$TotalTime = \left(\frac{TimeDepthSearch + Deduplication}{2}\right)$$

V. Performance Analysis

In this paper, the experiments are performed on *customer*, *suplier and order* datasets. The *customer* dataset consist of attributes as *personid*,*personname*, *dob*, *pob*, *email*, *address1*, *address2*. The *suplier*dataset consist of attributes*suppkey*, *name*, *address*, *nationkey*, *phone*, *acctbal*, *comment*. The *Order*datasetconsists of *orderkey*, *custkey*, *orderstatus*, *totalprice*, *orderdate*, *orderpriority*, *and clerk*.

The datasets are downloaded from *www.cs.washington.edu/research/xmldatasets/www.repository.html*

5.1 Test Case I

In *customerdetails*file there are seven attributes they are *personid*, *personname*, *DOB*, *POB*, *email*, *address1 and address2*. When parsed the *customerdetails*XML file then which elements are exact duplicates then that records are duplicates.

Record 1

```
<Customer>
<personid>1</personid>
<personname>Order#0000001</personname>
<dob>01/01/2001</dob>
```

<pob>Australia</pob>
<email>Order1@gmail.com</email>
<Address1>Jalna</Address1>
<Address2>Beed</Address2>
</Customer>

Record 2
<Customer>
<personid>1</personid>
<personname>Order#0000001</personname>
<dob>01/01/2001</dob>
<pob>Australia</pob>
<email>Order1@gmail.com</email>
<Address1>Jalna</Address1>
<Address2>Beed</Address2>
</Customer>

Above both records are exactly same then they are exact duplicates. But if attributes places of *Adddress1 and Address2* are changed then readings as

Record 3
<Customer>
<personid>1</personid>
<personname>Order#0000001</personname>
<dob>01/01/2001</dob>
<pob>Australia</pob>
<email>Order1@gmail.com</email>
<Address1>Jalna</Address1>
<Address2>Beed</Address2>
</Customer>

Record 4
<Customer>
<personid>1</personid>
<personname>Order#0000001</personname>
<dob>01/01/2001</dob>
<pob>Australia</pob>
<email>Order1@gmail.com</email>
<Address1>Beed</Address1>
<Address2>Jalna</Address2>
</Customer>

It Shows, they are not (100 %) exactly Duplicates, because of their address places are change, but they are duplicates. They are represented in different manner so they represent to same object.

5.2 Test Case 2

In *CustomerDetails*, alternate names for *countries* are given,soitcan be easily identifiedwhetherit is duplicate records or not.

Table 4: Alternate Names for Countries

Countries	Other name	Lowercase name
Australia	AU	au
Canada	CA	ca
China	CN	cn
India	IN	in
United Kingdom	UK	uk
Sri Lanka	LK	lk
France	FR	fr
Iceland	IS	is
Mexico	MX	mx
New Zealand	NZ	nz

Record 1
<Customer>
<personid>1</personid>
<personname>Order#0000001</personname>
<dob>01/01/2001</dob>
<pob>Australia</pob>
<email>Order1@gmail.com</email>
<Address1>Jalna</Address1>
<Address2>Beed</Address2>
</Customer>

Record 2
<Customer>
<personid>1</personid>
<personname>Order#0000001</personname>
<dob>01/01/2001</dob>
<pob>AU</pob>
<email>Order1@gmail.com</email>
<Address1>Beed</Address1>
<Address2>Jalna</Address2>
</Customer>

Record 3
<Customer>
<personid>1</personid>
<personname>Order#0000001</personname>
<dob>01/01/2001</dob>
<pob>au</pob>
<email>Order1@gmail.com</email>
<Address1>Beed</Address1>
<Address2>Jalna</Address2>
</Customer>

These records are not (100 %)Duplicates,becausethe countries names are different, but all they refer to one country,so above 3 records are duplicates. Table 3 shows the performance results.

VI. Graphical Model

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

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

Table 5 shows the calculated results for Dogmatix and Xml Duplication Using Xpath.Also the graphical comparison of Dogmatix and Xml Duplication using Xpath for customer dataset, suplier dataset and order dataset are given below.

Table 5: Comparison Results In terms Of Precision and Recall

Dataset	Dogmatix		XML Duplication Using Xpath	
	Precision	Recall	Precision	Recall
Customer	0.4522	0.45	0.502	0.5
Suplier	0.1206	0.12	0.417	0.41
order	0.417	0.41	0.48	0.48

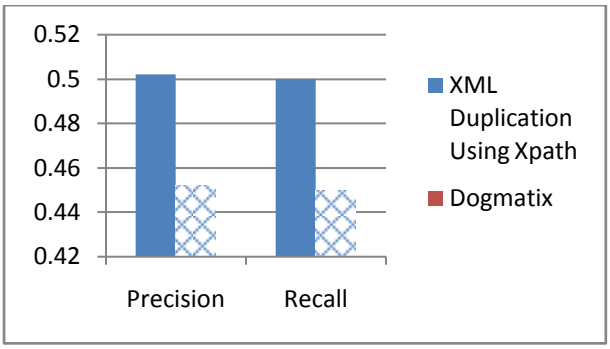

Figure 4: For Customer Dataset

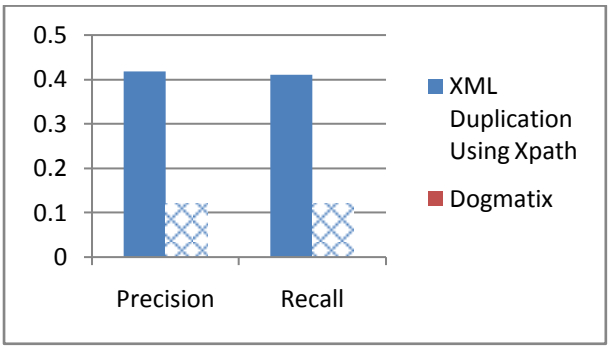

Figure 5: For Suplier Dataset

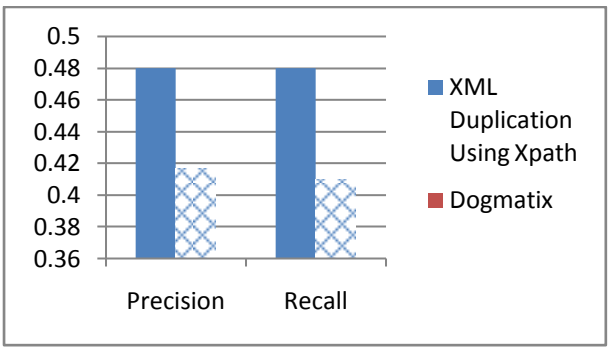

Figure 6: For Order Dataset

VII. Conclusion

In this paper, we present the algorithm to determine whether two recordsare duplicates or not based on a given threshold. For finding the duplicates the Algorithm uses a Bayesian network. Xml duplication using Xpath requires little user interaction, since user only needs to provide the Xml dataset file and based on that file the user has to give the threshold value. However this technique is very flexible for duplicate detection in XML data.

These techniques will able to solve the problem of duplicate data. Nowadays more and more data is generated, because of various devices so thereis lots of data to be maintained in the database so there can be duplicate records for a single record. To avoid the problem of efficiency of network, this technique can be useful to reduce the network load. This technique is performed under various experiments to find duplicate values. These experiments are performed on both artificial and real world dataset and showed that XML duplication using Xpath is very good technique. The process of duplication and Deduplication of records using Xpath technique is able to outclass other parser such as (Ex. DOM, SAX, and STAX Etc.).

When calculated Recall and Precision for Xpath is 93 %, when compared with Recall and Precision for Dogmatix is 66 %. The success demonstrated in the experimental results will show that there is still something more we can do for the future work.

References

[1] E. Rahm and H.H. Do, "Data Cleaning: Problems and Current Approaches," IEEE Data Eng. Bull., vol. 23, no. 4, pp. 3-13, Dec. 2000.

[2] Melaine Weis and Felix Naumann, "Detecting Duplicate Objects In XML,"ACM SIGMOD Special Interest Group on Mgmt of Data IQIS 2004, pp. 10-19, 2004
[3] R. Ananthakrishna, S. Chaudhuri, and V. Ganti, "Eliminating Fuzzy Duplicates in Data Warehouses," Proc. Conf. Very Large Databases (VLDB), pp. 586-597, 2002.
[4] D.V. Kalashnikov and S. Mehrotra, "Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph."ACM Trans. Database Systems, vol. 31, no. 2, pp. 716-767, 2006.
[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. Leita˜o, P. Calado, and M. Weis, "Structure-Based Inference of XML Similarity for Fuzzy Duplicate Detection," Proc. 16th ACM Int'l Conf. Information and Knowledge Management, pp. 293-302, 2007.
[7] A.M. Kade and C.A. Heuser, "Matching XML Documents in Highly Dynamic Applications," Proc. ACM Symp. Document Eng. (DocEng), pp. 191-198, 2008.
[8] D. Milano, M. Scannapieco, and T. Catarci, "Structure Aware XML Object Identification," Proc. VLDB Workshop Clean Databases (CleanDB), 2006.
[9] Luis Leitao, PavelCalado, and Melaine Herschel, "Efficient and Effective Duplicate Detection In Hierarchical Data"Vol. 25, No. 5, pp. 1028-1041 2013.
[10] 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.
[11] S. Puhlmann, M. Weis, and F. Naumann, "XML Duplicate Detection Using Sorted Neighborhoods," Proc. Conf. Extending Database Technology (EDBT), pp. 773-791, 2006.