Levenshtein Distance Algorithm for Efficient and Effective XML Duplicate Detection

Mrs.Shital Gaikwad[1]
Department of Computer, KJCOEMR,
Savitribai Phule Pune University
Pune, India
shtlgaikwad7@gmail.com

Prof.Nagaraju Bogiri[2]
Department of Computer, KJCOEMR,
Savitribai Phule Pune University
Pune, India
mail2nagaraju@gmail.com

***Abstract*—Electronic Data Processing used automated methods for processing commercial data. There is big amount of work on discovering duplicates in relational data; merely elite findings concentrate on duplication in additional multifaceted hierarchical structures Electronic information is one of the key factors in several business operations, applications, and determinations, at the same time as an outcome, guarantee its superiority is necessary. Duplicate finding a little assignment because of the actuality that duplicates are not accurately equivalent, frequently because of the errors in the information. Accordingly, many data processing techniques never apply widespread assessment algorithms which identify precise duplicates. As an alternative, evaluate all objective representations, by means of a probably compound identical approach, to identifying that the object is real world or not. Duplicate detection is applicable in data clean-up and data incorporation applications and which considered comprehensively for relational data or XML document. This paper will provide the person who reads with the groundwork for research in Efficient and Effective Duplicate Detection in Hierarchical Data or XML data.**

***Keywords—duplicate detection, electronic data, hierarchical data, XML data, Bayesian Network, NED*.**

I. INTRODUCTION

The concept of Duplicate belongs to the class of problem such as data mining. If two objects which may have different representations, but have similar semantics are said to be duplicate objects.

There is lots of work on discovering duplicates in relational data. Due to the rapid changes in online information, it is necessary to hastily identify changes in XML documents is significant to Internet query organization; various search engines, and permanent query systems. Duplicate finding a little task because of the actuality that duplicates are not exactly equivalent, frequently because of the errors in the information. Accordingly, many data processing techniques never apply widespread assessment algorithms which identify precise duplicates. As an alternative, evaluate all objective representations, by means of a probably compound identical approach, to identifying that the object is real world or not. Duplicate detection is applicable in data clean-up applications and which considered XML is growing in use and popularity to publishing documents on web. It is more challenging than the relational data in the view of object types and attributes types, because XML have different structure of instances of the same object and representation in the form of hierarchical structure which exploit efficiency and effectiveness.

Duplicate detection consists of identifying different representations of objects and each object represented in a data resource. Duplicate detection plays an important role in data clean-up applications. Detecting duplicates records in several types of objects which are associated to each other and then plan techniques modified to semi-structured XML data. Hierarchical structure means associations between different types of objects i.e. tree or a graph structure. XML data which is organized hierarchically is semi-structured in nature; the duplicate detection process in XML data is complicated than well-structured relational data. There are two problems in duplicate detection that are object definition and structural diversity. The description of object refers to the problem of defining which data values actually describe an object, i.e. while comparing two objects which value to consider. Duplicate detection in relational data assumes that each tuple represents an object and object is described by all attribute values. It considers data equivalence, significance, relationship, and made separation of misplaced and ambiguous data. This paper will provide the person who reads with the groundwork for research in Duplicate Detection in Hierarchical Data or XML data.

II. LITERATURE RIVIEW

Yuan Wang David J. DeWitt Jin-Yi Cai [2] introducing an efficient algorithm known as X-Diff which incorporates the XML formation uniqueness for tree to tree modification methodology. A disordered model (only ancestor relationships are significant) is more suitable for most database applications. With the help of an unordered model, any modification identification is difficult task as compare to the ordered model, although the transformation in the result that it generates is more precise. In this paper they describes X-Diff algorithm for identification among the two different versions of the XML data with the help of XML structure information and unordered trees. Luı´sLeita˜ o, Pa´ velCalado, and Melanie Herschel [1] present a new scheme for XML duplicate identification known as XML Dup. The invented method XML Dup is used to find out the probability of the two XML elements which are being duplicates and which can be done with the help of Bayesian network.

Zur Erlangung des akademischen [8] Grades worked on the Duplicate recognition which helping in identifying numerous representations of real world objects, and each object represented in a data resource.

Melanie Weis and Felix Naumann [7] presenting a structure for duplicate identification. This structure is based on the three components which are namely candidate definition, duplicate definition and duplicate detection. The candidate definition decides which two objects are going to compare. The duplicate definition is decide the duplicate candidates are duplicate or not and the duplicate detection identify a way to find out duplicates. Using this structure or framework implemented a technique for XML duplicate identification which is nothing but DogmatiX. This DogmatiX technique is compare XML elements depending on the direct values of data as well as the structure of parent, children etc.

Le Chen, Lei Zhang, Feng Jing, Ke-Feng Deng and Wei-Yeing Ma [9] study a Web Object Ranking problem which plays an important role in building a search engines. In this paper a new method is given known as two score fusion. This technique is related to identifying duplicate photos. The method controls the links which are hidden and identified by the duplicate photo detection algorithm. RohitAnanthakrishna, SurajitChaudhuri, Venkatesh Ganti. They proposed the algorithm for identifying and removing duplicates from the data store in data warehouse which linked with hierarchies. For the data cleaning solution real world entity plays an important role. The duplicate removal problem which used to identifying several tuples. Correctness of the data is very essential because decision support analysis on the data warehouses determining important business decisions.

Duplicate records in XML file mean multiple representation of an entity that store outdated data. Duplicates are not exactly equal; due to errors in the data we cannot use common comparison algorithms that detect exact duplicates.

III. Levenshtein Distance Algorithm FOR Duplicate Detection

A. Existing System.

Most commonly we use relational database to store the data. In this case, the detection strategy typically consists in comparing pairs of tuples (each tuple represent- ing an object) by computing a similarity score based on their attribute values. Then, two tuples are classified as duplicates if their similarity is above a predefined threshold. However, this narrow view often neglect so their available related information as, for instance, the fact that data stored in a relational table relates to data in other tables through foreign key.

B. Objective

The aim of this project is to present a novel algorithm to find duplicate objects in the hierarchical structures, like XML files. The novel algorithm concentrates on a particular type of error, namely fuzzy duplicates, or duplicates for short.

C. Proposed System

Proposed system finds the duplicates in structured (xml) data. To find out duplicate records it compares corresponding leaf node values of both objects. BN structure (Bayesian Network) is constructed for comparing objects and contains leaf node values of both objects. This BN structure is passed to pruning algorithm that finds the similarity between the leafnode values and propagates the similarity score from leaf node to the root as probability of root node being duplicates. Proposed system uses Levenshtein distance algorithm which is best and efficient than the previous NED algorithm.

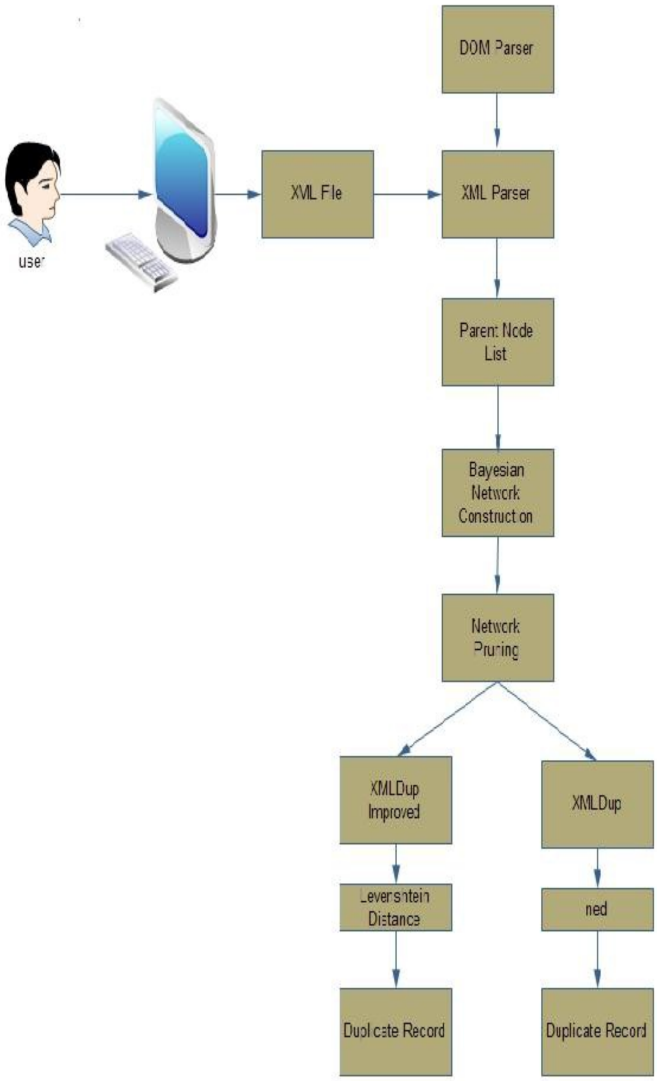

Figure 1.Proposed System Architecture.

D. Parameters to achieve

We achieve the Precision and Recall as a parameter in the project which shows the exact result of project. The precision and recall are explained in detail below.

1. **Precision:** No. of relevant results are retrieved out of total no of retrieved results.

2. **Recall:** No. of relevant results retrieved out of total number of results.

E.g.: Our total number of XML records is 50. i.e. total no. of results

After BN Structure process we get 48 records as aligned. i.e. total no. of retrieved results.

Out of which 15 are relevant that means correctly aligned. i.e. relevant results so,

Precision = 15/48.
Recall = 15/50.

IV.IMPLEMENTATION PROCESS

Process Summary:

a. Take a two files contains few records including the duplicates.

b. Parse the xml files using **XML Parser API.**

c. Construct a BN structure for two objects being duplicates.

d. If the two nodes are similar more than threshold value then those two nodes are duplicates.

During process of duplicate detection an XML files taken as input, which contains some duplicate records .Then files are parsed by parser which gives description like root node & parent node .Then Bayesian network is constructed for two nodes which are being duplicates. Bayesian network is tree like structure in which node represents records and edges represents relationship between records. Then Prior probabilities are calculated with respect to leaf nodes and conditional probabilities are calculated with respect to inner node.

Then average probability is calculated by adding conditional and prior probabilities. This average probability is then compared with predefined threshold, if average probability is greater than threshold then records is duplicate otherwise not.

V. DATA SET AND COMPARISON RESULT

Experimental tests were performed on different datasets. The data sets are CD, Cora, Country which consist of XML objects taken from a real database and artificially polluted by inserting duplicate data and different types of errors. [13]

Experiments were performed to compare the effectiveness of the tested algorithms. To check effectiveness, we applied the commonly used precision, recall, measures [14]. 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 1.
Recall and Precision Score Comparisons using NED and Levenshtein Distance Algorithm.

Input Data Set		**Using NED Algorithm**		**Using Levenshtein Distance Algorithm**	
File1	***File2***	***Precision***	***Recall***	***Precision***	***Recall***
CD1	CD2	0.06	0.06	0.09	0.09
Cora1	Cora2	0.13	0.13	0.42	0.42
Country1	Country2	0.18	0.37	0.24	0.48

Comparison results for Duplicate detection:

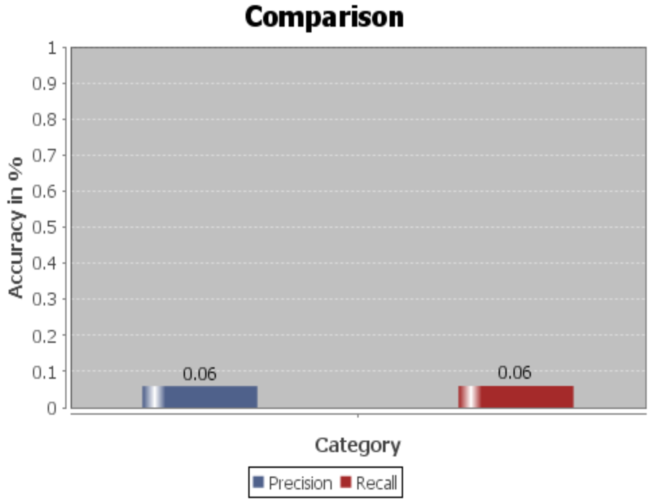

Figure 2. Duplicate Detection Using Normalized Edit Distance algorithm Over CD1 and CD2 dataset.

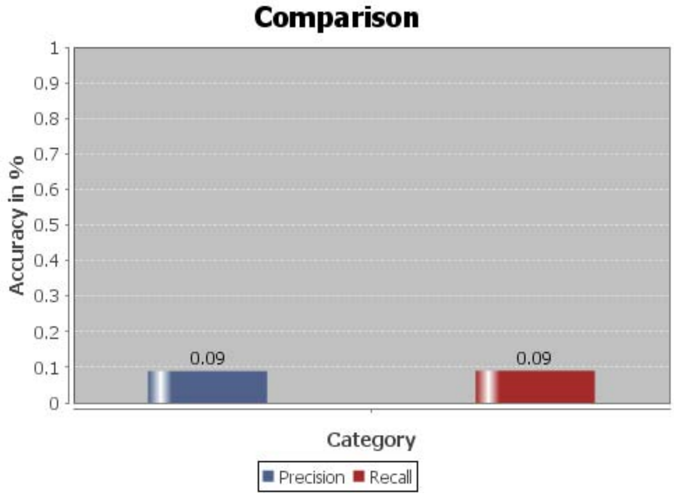

Figure 3. Duplicate Detection Using Levenshtein Distance algorithm Over CD1 and CD2 dataset.

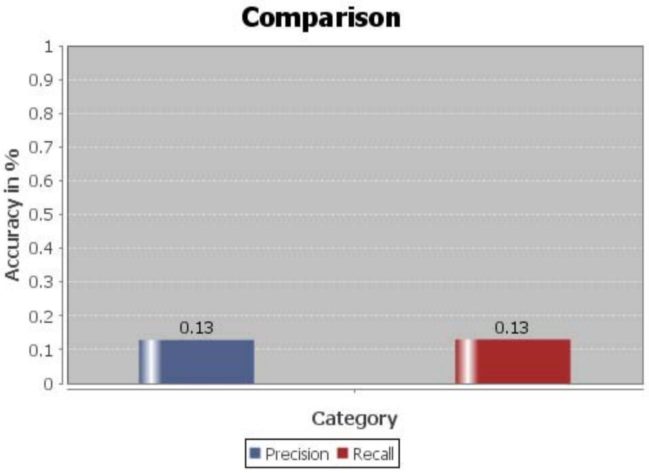

Figure 4. Duplicate Detection Using Normalized Edit Distance algorithm Over CORA1 and CORA2 dataset.

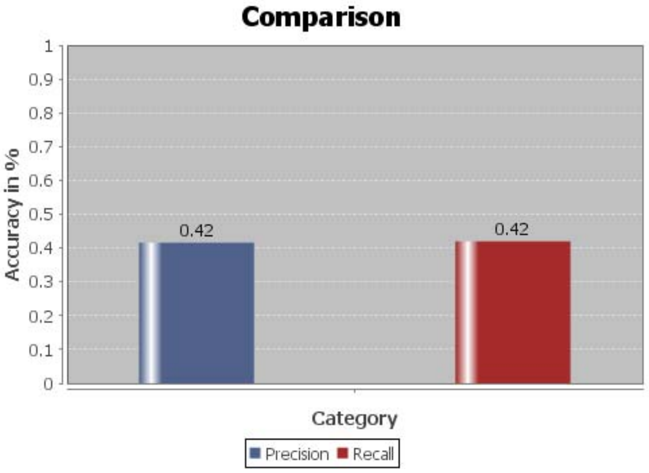

Figure 5. Duplicate Detection Using Levenshtein Distance algorithm Over CORA1 and CORA2 dataset.

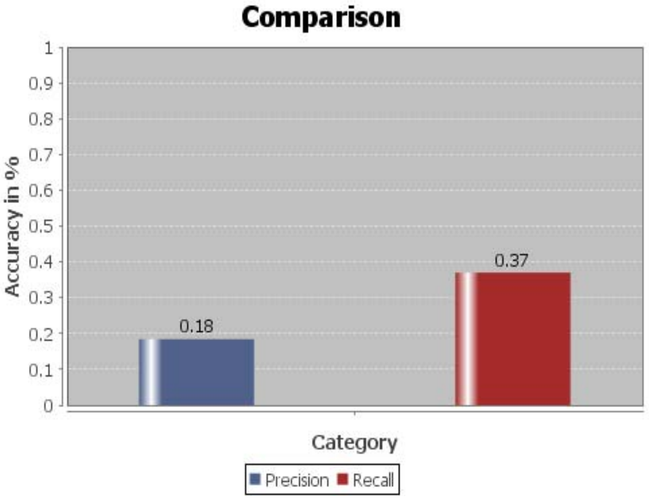

Figure 6. Duplicate Detection Using Normalized Edit Distance algorithm Over COUNTRY1and COUNTRY2 dataset.

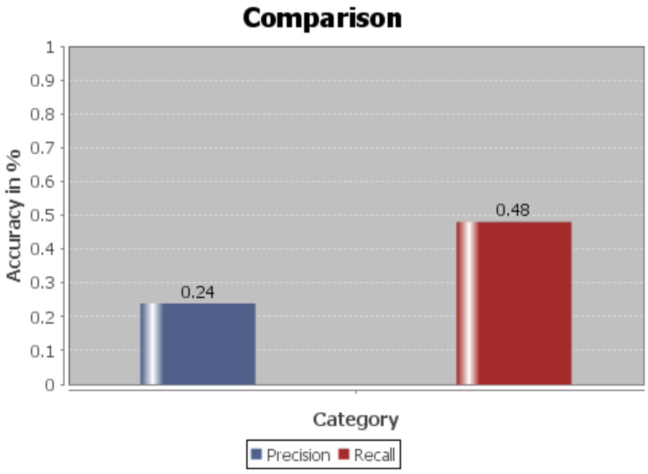

Figure 7. Duplicate Detection Using Levenshtein Distance algorithm Over COUNTRY1 and COUNTRY2 dataset.

Why the Levenshtein distance algorithm is best for our System

In roughly string matching, the main aim is to find matches for small length strings in many longer texts, where a small number of differences are to be expected. The small length strings that could come from a dictionary, for example here, one of the strings are small length, while the other is long string. This has a wide range of applications; for instance, spell checkers, correction systems for optical recognition of characters, are based on translation memory.

NP hard or NP complete:

Our project comes into the NP complete, because in particular time project it gives the results. For the decision problem, so that it give the solution for the problem within polynomial time. The set of all decision problems whose solution can be provided into polynomial time by using the Levenshtein algorithm.

VI.CONCLUSION

Levenshtein distance algorithm uses a Bayesian Network to determine the probability of two XML objects being duplicates. The Bayesian Network model is composed from the structure of the objects being compared, thus probabilities of all objects are computed considering not only the information the objects contain, but also how such information is structured. Levenshtein distance algorithm is very flexible, XMLDup requires little user intervention user only needs to provide the dataset and a similarity threshold.

Using Levenshtein Distance algorithm gives better result than Normalized Edit Distance algorithm. To improve the runtime efficiency of XMLDup process, a network pruning strategy is also presented. The experiments performed on mention data achieve high precision and recall scores.

ACKNOWLEDGMENT

We thank the referees for their valuable suggestions to improvise the content and quality of this paper. The author is grateful to our principal for providing necessary facilities. We acknowledge the diligent efforts of our Head of the Department to guide us towards implementation of this paper.

REFERENCES

[1] Luı´sLeita˜ o, Pa´ velCalado, Melanie Herschel “Efficient and Effective Duplicate Detection in Hierarchical Data”, Knowledge and Data Engineering, IEEE Transactions, Volume: 25, Issue: 5, ISSN:1041-434,May 2013.

[2] Yuan Wang David J. DeWitt Jin-Yi Cai “Local X-Diff: An Effective Change Detection Algorithm for XML Documents” Data Engineering, 2003. Proceedings. 19th International Conference, ISBN: 0-7803-7665, March 2013.

[3] Diego Milano, Monica Scannapieco, Tiziana Catarci, “Structure-aware XML Object Identification”, in ’CleanDB’, 2006.

[4] Adrovane Marques Kade, Carlos Alberto Heuser “Matching XML Documents in Highly Dynamic Applications”, DocEng '08 Proceedings of the eighth ACM symposium on Document engineering’, ISBN: 978-1-60558-081-4, 16 Sep 2008.

[5] Erhard Rahm, Hong Hai Do “Data Cleaning: Problems and Current Approaches”, IEEE Data Eng. Bull.23, no. 4 (2000): 3--13.

[6] RohitAnanthakrishna, SurajitChaudhuri, Venkatesh Ganti ba “Eliminating Fuzzy Duplicates in Data Warehouses”, VLDB '02 Proceedings of the 28th international conference on Very Large Data Bases, August 2002.

[7] Melanie Weis, Felix Naumann, and Franziska Brosy, “A Duplicate Detection Benchmark for XML (and Relational) Data”, Proc. of Workshop on Information Quality for Information Systems (IQIS), 2006.

[8] Zur Erlangung des akademischen “Duplicate Detection in XML Data.

[9] Le Chen, Lei Zhang, Feng Jing, Ke-Feng Deng and Wei-Yeing Ma, “Ranking Web Objects from Multiple Communities”, CIKM '06 Proceedings of the 15th ACM international conference on Information and knowledge management, Pages 377-386, ISBN:1-59593-433-2,

[10] Melanie Weis and Felix Naumann, “DogmatiX Tracks down Duplicates in XML”, SIGMOD '05 Proceedings of the 2005 ACM SIGMOD international conference on Management of data, ISBN: 1-59593-060-4.June 2005.

[11] Dmitri V. Kalashnikov and SharadMehrotra, “Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph”, ACM TODS Journal, Vol. 31(2), June 2006.

[12] ZaiqingNie, Yuanzhi Zhang, JiRon Wen, WeiYing Ma, “Object Level Ranking: Bringing Order to Web Objects”, 05 Proceedings of the 14th international conference on World Wide Web, ISBN: 1-59593-046-9 May 2005.G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[13] 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.

[14] R.A. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., 1999.