Effective and Efficient XML Duplicate Detection Using Levenshtein Distance Algorithm

Shital Gaikwad[1], Nagaraju Bogiri[2]

[1, 2]Department of Computer, KJCOEMR, Savitribai Phule Pune University, Pune, India

Abstract: ***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. Duplicates are several delegacy of the identical real world thing which is dissimilar from each other. 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 it is suggested to use Levenshtein distance algorithm which is best and efficient than the previous Normalized Edit Distance (NED) algorithm. This paper will provide the person who reads with the groundwork for research in Duplicate Detection in XML data or Hierarchical Data.***

Keywords: duplicate detection, , electronic data, hierarchical data, XML data , XML document

1. Introduction

The concept of Duplicate belongs to the class of problem such as data mining and discovery of knowledge. The two objects which may have different representations or structure, but have similar semantics are said to be duplicates.

There is big amount 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 small task 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. Generally XML document is represented in a tree format.

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 multiple representations of real world objects and each object represented in a data resource. Duplicate detection has an important role in data clean-up and data incorporation applications. Detecting duplicates in several kinds of objects associated to each one and plan techniques modified to semi-structured XML data. Associations between several kinds of objects form a hierarchical i.e. tree or a graph structure. XML data which is organized hierarchically is semi-structured in nature; the object detection assignment in XML data is complicated as compared to well-structured relational data. There are two problems which 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. Relational duplicate detection methods assume that each tuple represents an object and all attribute values describe that object. It considers data equivalence, significance, relationship, and made separation of misplaced and ambiguous data. Duplicate identification is valid 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 Duplicate Detection in Hierarchical Data or XML data.

2. Literature Review

Efficient and Effective Duplicate Detection in Hierarchical Data [1]gives a new scheme for XML duplicates identification known as XML Dup.It is used to find out the probability of the two XML elements being duplicates. XML Dup constantly indicates improved results in terms of efficiency and effectiveness.

Local X-Diff [2] In Effective Change Detection Algorithm for XML Documents this paper gives an algorithm known as X-Diff which incorporates the XML formation uniqueness for tree to tree modification methodology. Eliminating Fuzzy duplicates in Data Warehouses paper Proposes an algorithm for identifying and removing duplicates from the data store in data warehouse which linked with hierarchies. Rohit Ananthakrishna1 Surajit Chaudhuri Venkatesh Ganti [6] proposed the algorithm for identifying and removing duplicates from the data store in data warehouse which linked with hierarchies Exploited dimensional hierarchies in data warehouses to extend quality, scalable, and proficient algorithm to identify fuzzy duplicates in Dimensional tables

DogmatiX Tracks down Duplicates [10] in XML Presenting a structure for duplicate identification which is based on the three components which are namely candidate definition, duplicate definition and duplicate detection.

Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph [11] paper authors studying the crissis of reference elucidation. Particularly a condition in which entities are represented with the help of descriptors in the database. The main objective behind this is to recognize the distinctive entity to which every description corresponds. When entities in a database hold references to other entities then the reference disambiguation problem come into the picture.

Object Level Ranking: Bringing Order to Web Objects [12] Zaiqing Nie, Yuanzhi Zhang, JiRon Wen, WeiYing Ma introducing Pop Rank, a domain self-governing entity stage link examination model to grade the objects contained by a particular domain. On the web many search engines considers the complete web page for retrieval purpose but several type of objects are linked with the static web pages or in the databases. Information on web for which are very much related with the particular area and giving order to these objects in conditions of their significance and reputation to respond user queries.

3. Proposed 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.

Proposed system finds the duplicates in structured (xml) data. To find out duplicate records we compare 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 leaf node to the root as probability of root node being duplicates. Also we prove that the proposed algorithm is best and efficient than the previous algorithm.

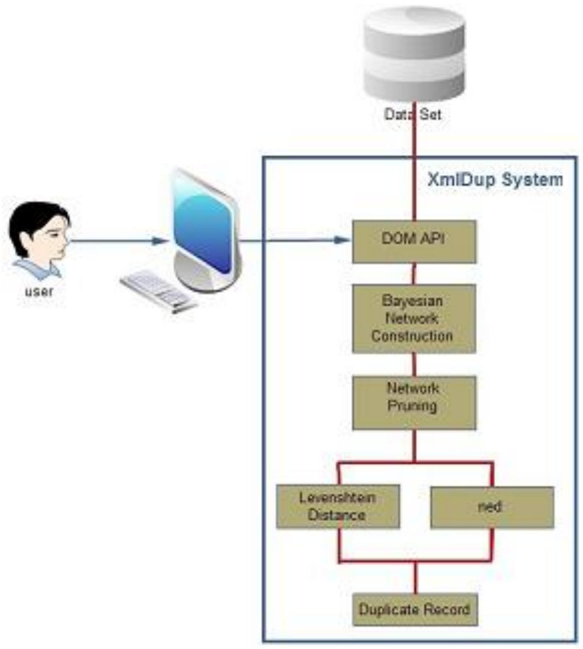


Figure 1: Proposed System Architecture

4. Mathematical Model

Set Theory

1) Identify XML records R

R = {r1, r2, r3, r4….}

Where R is main set of records

2) Identify Nodes of each record N

N = {n1, n2, n3, n4….}

Where N is main set of nodes for a record

3) Probability that a node is duplicate

P = {p1, p2, p3, p4….}

$$P(t_{ij}|V_{t_{ij}}, C_{t_{ij}}) = \begin{cases} 1 & \text{iff } V_{t_{ij}} = C_{t_{ij}} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Where P is main set of probability

4) Identify Duplicate nodes DN

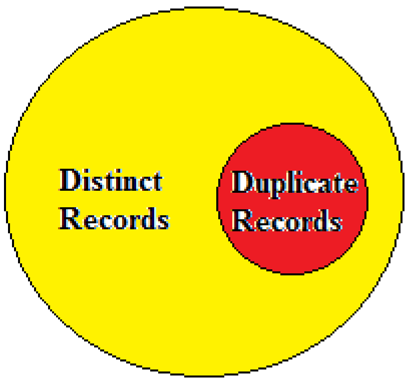
DN = {dn1, dn2, dn3, dn4….}

Where DN is main set of duplicate nodes

5) Identify Duplicate records DR

DR = {dr1, dr2, dr3, dr4….}

Where DR is main set of duplicate records



1) Process:
We first present a probabilistic duplicate detection algorithm for hierarchical data called XML Dup.

Process P = {e1, e2, e3, e4…}
{e1 = XML Parser}
{e2 = Network Pruning}

Implementation Result

Process Summary:

a. Take a file that contains few records including the duplicates.
b. Parse the xml file 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.

Data Set and Comparison Result

Experimental tests were performed on different datasets. The data sets are Cora, Country, Employee 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

Data Set	*Usingned Algorithm*		*Using Levenshtein Distance Algorithm*	
	Precision	Recall	Precision	Recall
Employee (100 records)	0.45226	0.45	0.50251	0.5
Cora (200 records)	0.12060	0.12	0.41708	0.415
Country (200 records)	0.41708	0.41	0.48241	0.48

Comparison results for Duplicate detection:

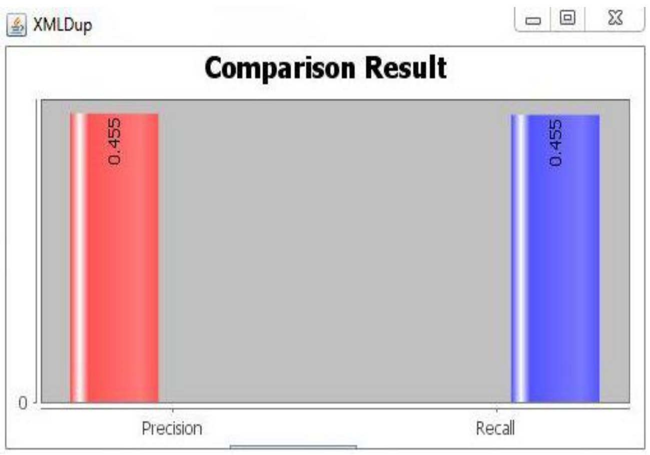


Figure 2: Duplicate Detection Using Normalized Edit Distance Over Employee Dataset.

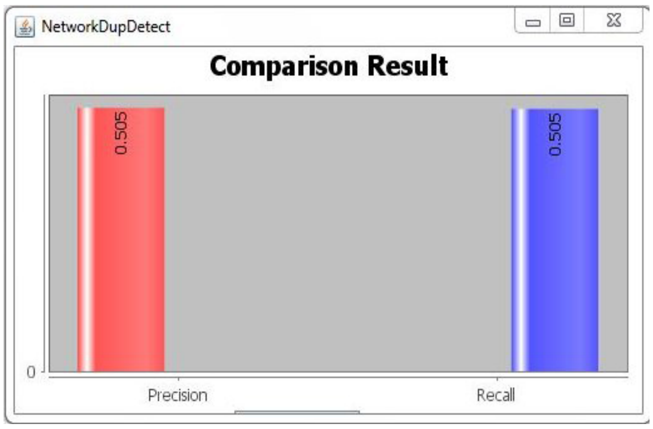


Figure 3: Duplicate Detection Using Levenshtein Distance Algorithm Over Employee Dataset

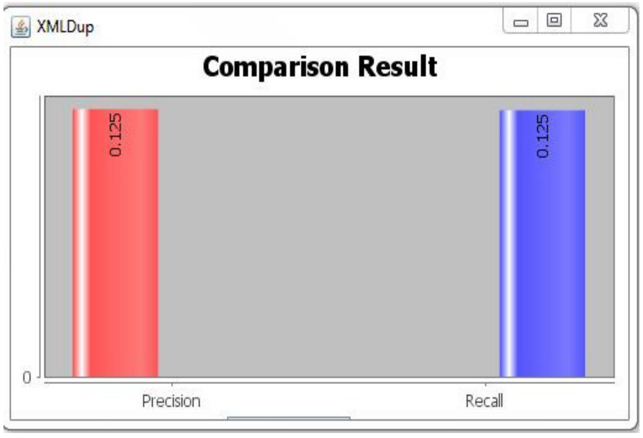


Figure 4: Duplicate Detection Using Normalized Edit Distance Over Cora Dataset.

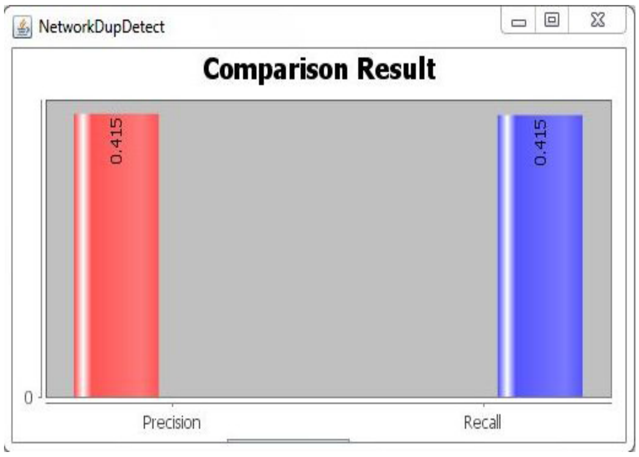


Figure 5: Duplicate Detection Using Levenshtein Distance Algorithm Over Cora Dataset.

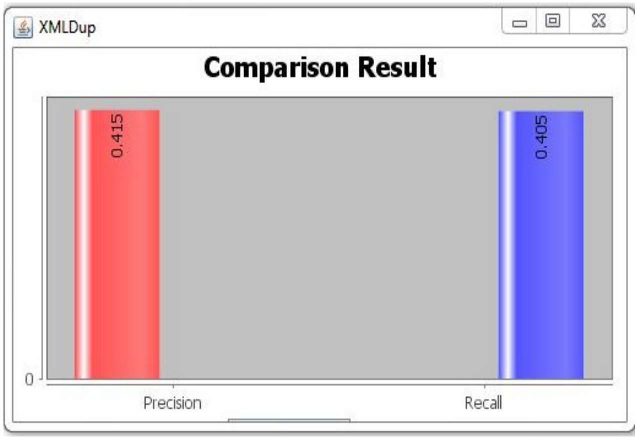


Figure 6: Duplicate Detection Using Normalized Edit Distance over Country Dataset

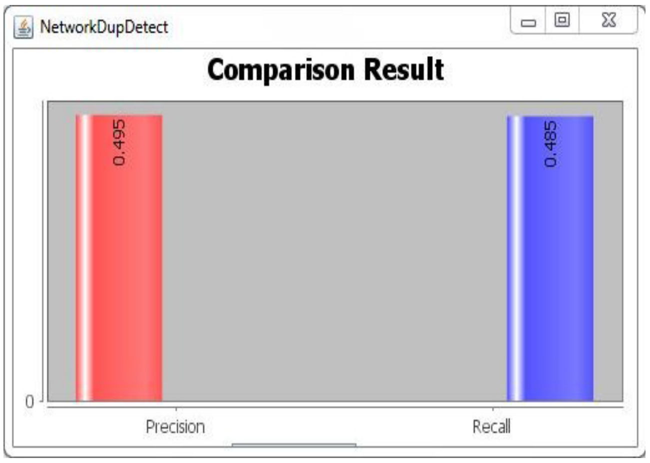


Figure 7: Duplicate Detection Using Levenshtein Distance Algorithm Over Country Dataset.

Why the Levenshtein distance algorithm is best for our System….

In approximate string matching, the main aim is to find matches for short length strings in many longer texts, where a small number of differences are to be expected. The short length strings that could come from a dictionary, for example here, one of the strings are short 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, and software to assist natural language translation based on translation memory.

The Levenshtein distance algorithm can also be computed between two longer length strings. The costs of computing distance, is proportional to the product of the two string lengths, which makes this impractical. Thus, when used to aid in fuzzy string searching in applications such as record linkage,to help improve speeds of comparisons the compared strings are usually short.

5. 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, which allows the use of different similarity measures and different ways of combining probabilities. Using Levenshtein Distance algorithm gives better result Than Normalized Edit Distance algorithm. Experiments performed on both artificial and real-world collected data showed that our algorithm is able to achieve high precision and recall scores in several contexts.

6. Acknowledgment

We thank the mysterious 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 towards carrying out this work. We acknowledge the diligent efforts of our Head of the Department to guide us towards implementation of this review paper.

References

[1] Luı´s Leita˜ o, Pa´ vel Calado, 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] Rohit Ananthakrishna, Surajit Chaudhuri, Venkatesh Ganti ba "Eliminating Fuzzy Duplicates in Data Warehouses", VLDB '02Proceedings 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 '06Proceedings of the 15th ACM international conference on Information and knowledge management, Pages 377-386, ISBN:1-59593-433-2, 2006.
[10] Melanie Weis and Felix Naumann, "DogmatiX Tracks down Duplicates in XML", SIGMOD '05 Proceedings of the 2005 ACM SIGMOD Dinternational conference

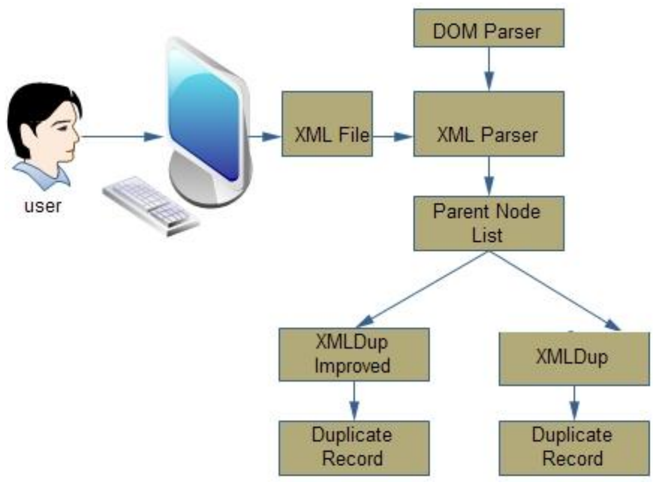
on Management of data, ISBN: 1-59593-060-4.June 2005.

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

[12] Zaiqing Nie, 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.


DOM Parser
user
XML File
XML Parser
Parent Node List
XMLDup Improved
XMLDup
Duplicate Record
Duplicate Record