DRTX: A Duplicate Resolution Tool for XML Repositories

Randa Mohamed Abd El-ghfar, Ali EL-Bastawissy and Mostafa Abdel Azeim Mostafa

Arab Academy for Science, Technology & Maritime Transport, Cairo,Egypt

Faculty of Computer science, Cairo University, Cairo, Egypt

Arab Academy for Science, Technology &Maritime Transport, Cairo, Egypt

Summary
Detecting duplicates in XML is not trivial due to structural diversity and object dependency. This paper suggests a duplicate detection and resolution tool (DRTX) which is an efficient XML duplicates detector and resolution that applies two famous techniques of duplicates detection, normal edit distance (NED) and token based damerau-levenshtein distance algorithm (TBED) then compare the results and suggests the better similarity for each of them. DRTX is not only a duplicate detection and resolution system but it also provides two extra services: - first the XML file merger which is used to merge XML documents thus solves the structure heterogeneity problem, second dirty XML generator which is used to insert known duplicate problems on clean XML file to apply the mentioned algorithms on that file therefore explore how much the system can detect accurately these problems.To minimize the number of pair-wise element duplicates comparison, a set of filters were used to increase the efficiency of DRTX while its effectiveness is adjustable. Experimental results show that there is no algorithm better than the other but each of them has its own use ie.NED is better to use at lower threshold similarity values while TBED is better at higher ones.

Keywords
Duplicates detection, XML, similarity, Data cleaning, efficiency and effectiveness of detection Algorithms.

1.Introduction

The process of duplicates detection and resolution is not easy and not trivial because duplicate data represents the most dangerous problem affecting data. It is essential because these duplicates are due to the integration of data coming from different sources and that data suffering from many types of errors as, Typographical errors, Spilling errors, Inconsistent formatting & abbreviations, Different representations of the same logical value and missing values [1].

These errors may come due to many reasons as Poor data-entry procedures, Lack of a standard representation of data, Poor data validation, changing of object data, Poor data integration. As a consequence, duplicates detection and resolution can’t be performed by just checking the equality of attributes but more complex algorithms are required as the use of complex similarity measure to compare objects pair-wisely [2].

In this paper we suggest Duplicates Resolution Tool (DRTX) for XML repositories. In DRTX we only handle artificial duplicates which have high percentage of similarity that will be detected using only the basic entity attributes. Problem statement and related work will be briefly discussed in section2 and section3 consequently then DRTX will be discussed in section.4. Extended tasks of DRTX will be described in Section 5. Comparison and analysis results used to evaluate the solution will be described in section6 while final conclusion will be mentioned in section7 and finally section8 will be suggested some future work.

2.Problem Statement

XML Duplicates detection and resolution is an important task of data cleaning process which we aim at identifying multiple representation of the same real world entity.

Detecting duplicates in the world of XML is not easy and suffers from many problems as object definition which refers to the problem of defining which data values actually describe an object (i.e. which values to consider when comparing two objects). Structural diversity which discusses the problem of multiple structures of the same XML elements unlike relational tubles that makes detecting duplicates in XML more complex than any other data store, and element dependency which means that XML elements relate to their ancestors and descendants. These relationships can be considered to improve duplicate detection. For instance, two <city> elements with text Los Angeles are nested under different <country> elements with text USA and Chile. Although the city names are identical, we can detect that they are not the same city in the real-world, because they are in different countries [3].

3. Related Work.

Duplicates detection and resolution is an important task in data cleansing which is applied in data integration, customer relationship management, data mining and data warehouse as in [4,5,1,6]. Indeed Most of the recent work

Manuscript received July 5, 2012
Manuscript revised July 20, 2012

focuses on duplicates detection and resolution on relational database as in [7,8,9,10].

Detecting duplicates in the field of XML in not easy because we face many problems as structural diversity and identifying Objects Description (OD) which will be used to compare objects. Many works try to resolve such problems as in [11] where many heuristics were identifying to choose OD as r-distant which consider as OD all elements whose depths in XML schema doesn't differ more than radius r from an XML node e depth .Also they proposed second way to choose OD named k-closest which considers the next K elements following node e in breadth first order. In addition [3] contributes in automatically select OD by making use of Statistics on XML element structure. As well [12] states that to choose two XML elements as OD, they must have the same name, equal/similar parent and similar children structure. In addition to [13] which design OD by a set of queries which will be then evaluated and union in one tuple.On the other hand to solve the problem of structure heterogeneity researches in [11,13,14,15] suggest mapping the two sources to a common schema while others in [4] assume that when 2 elements have different names also have semantics i.e. the data they contain can't represent the same real world entity. Thus two elements with equal element's name may be nested under other elements, their ancestors. If 2 elements have different ancestors we assume that they can't be duplicate thus solve the structure problem.

Detecting the duplicates between XML entities involves detecting similarity between entities, which is measured using their edit distance as in [3,12,11,13,16]. But researches in [17] develop an efficient algorithm for detecting duplicates in complex XML using MD5 algorithm. As well [15] presents a novel method for XML duplicate detection, called XMLDup which uses a Bayesian network to determine the probability of two XML elements being duplicates, considering not only the information within the elements, but also the way that information is structured. Authors in [18] Evaluate and compares several unconstrained clustering algorithms for duplicate detection by extensive experiments over various sets of string data with different characteristics. As well [19, 20] provide survey of duplicates detection methods and identify their strength and shortcoming.

To minimize the number of pair-wise element comparisons, an appropriate filter function is used, there are three traversal strategies used to prune expensive computations first is Top-down strategy to limit pair-wise comparisons to XML elements that have the same or similar ancestors. Second is Bottom-up strategy which first compares all leaf nodes and then only compares ancestors that have at least one child in common. Finally relationship aware strategy which determines an order of comparisons, based on the influence that elements have on each other [3]. While [12] used 3 filters: first length filter, triangle inequality and Bag distance. While [21] introduce two comparison strategies apply to all kind of parent/child relationship not only 1: n. First one of them uses the order to reduce the number of classification, second uses the order to reduce the number of missed comparison and hence rotationally missed duplicates. As soon as [17] introduce many filters as F (first letter),F(equal) and F(order) in addition to [15] which presents network pruning strategy that is applied in two in two ways, A lossless approach, with no impact on the accuracy of the final result, and a loosely approach, which slightly reduces recall.

4.Proposed Duplicates Resolution Tool for XML (DRTX).

In this paper we introduce DRTX which is the initials of Duplicates Detection and Resolution Tool for XML repository. The Architecture of (DRTX) is illustrated as shown in Figure(1) in which a dirty XML Document is introduced to DRTX to be verified to generate tokens using tokenization module then classify these tokens as duplicates or none duplicates using duplicate classifier module. After detecting duplicate records, they are handled using many scenarios as discussed in duplicate handling and cleaning module.

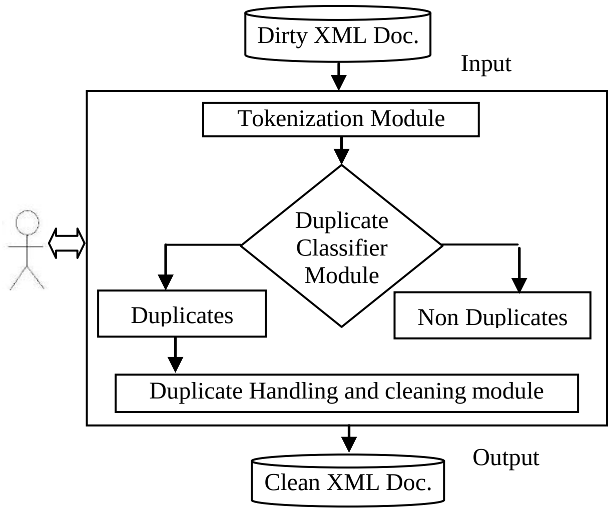

Figure (1) DRTX duplicate detection and resolution system (DRTX)

4.1 Tokenization Module:

This module aims to select the best objects that describe XML -Object Description phase- and implements data scrubbing and standardization -pre processing phase- on those objects to formulate the tokens (tokens formulation phase) as figure (2) illustrates. Token based Algorithms are used in relational database as in [1].

4.1.1 Object Description (OD) Selection Phase:

It defines which information describes an XML element that is considered as an object, so it will be used in duplicate detection to classify objects as duplicates or non-duplicates. Ideally, an OD includes information that characterizes a particular object, such as a book's ISBN [21].

Candidates or object description were designed by [17] as a set of queries $q_1, q_2, \ldots, q_k$, and the effect of the candidate selection(CS) operator is to evaluate all queries and union their results into a flow of 1-tuples. Formally: CS $(q_1,q_2,\ldots,q_k)= q_1 \cup q_2 \cup \ldots \cup q_k$. Also [3] suggests heuristics that automatically come up with object descriptions of their own.

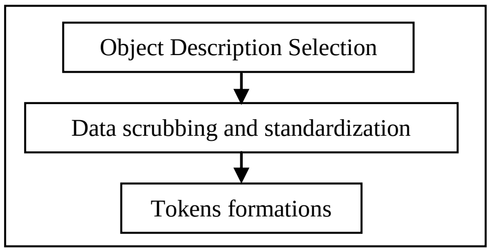
Object Description Selection
Data scrubbing and standardization
Tokens formations

Figure (2) Tokenization Module

Table (1) illustrates XML records before data standardization and scrubbing. The XML records are in the form of table just for simplicity.

Table (1) XML records before standardization and scrubbing

author	*title*	*genre*	*Publish date*
Gambardella, Matthew	XML Developer's Guide	Computer	2000-10-01
Gambardella, Matthew	XML Developer's Guide	Computer	2000-10-01
Ralls, mr.Kim	Midnight Rain	Fantasy	16-12-2000
Corets, mrs.Eva	Oberon's Legacy	Fantasy	10-mar-2001
Corats, Eva	Oberon's Legcy	Fantasy	2001-3-10
Randall, Cynthia	Lover Birds	Romance	2/9/2000
Randall, Cynthia	Lover Birds	Romance	2/9/2000

In the above example (OD) may be author, title and publish date as chosen by the user.

4.1.2 Data scrubbing and standardization Phase

Data scrubbing and standardization is used to store the data in a uniform formatting in order to detect duplicates accurate and correctly and not to miss any potential duplicates. Data scrubbing is the process of removing data that is incorrect, incomplete, improperly formatted. (e.g. Remove Mr., Mrs., the, in, on, in, at, dr,…..) While data standardization is the process of making all data of the same type or class conform to an established convention or procedure to ensure consistency and comparability across all XML records. (e.g. apply uniform formatting on date and time).

Data scrubbing and standardization is applied to Table(1) in which ", Mr., Mrs. " is removed from author and a date standardization is done to the publish date in the format D/M/YYYY.

4.1.3 Token formation Phase.

Tokens are formed for the selected XML objects description which will be used in the duplicate classifier module to detect duplicates. In the previous example tokens are formed for the objects author, title and publish date.

4.2 Duplicate classifier module.

Duplicate classifier takes token objects and examine them to determine whether they are duplicate or not according to similarity function which return true (i.e. Candidates are duplicate) or false (i.e. Candidates are not duplicate).

We adopt XML classifiers that use a similarity measure based on normal edit distance and enhanced token based damerau-levenshtein string edit distance. The Levenshtein distance between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character which will cost 2 units, While Damerau edit distance is the operations needed to transform one string into the other. These operations are inset, delete, and substitution of a single character or transposition of two adjacent char each will cost 1 unit.

The similarity between two strings s1 and s2 is calculated using equation (1).

$$\text{Similarity}(S_1, S_2) = 1 - \frac{Dist(S_1, S_2)}{MaxLen(S_1, S_2)} \quad (1)$$

Where Dist(S1,S2) is the edit distance between S1 and S2 and MaxLen(S1,S2) is the max string length of S1 and S2.

To get the final conclusion about whether the two XML records are duplicate or not, the similarity percentage will be compared by a similarity threshold defined by the expert user which will result in:

Duplicate if the similarity >= user threshold.

Not Duplicate if the similarity < user threshold.

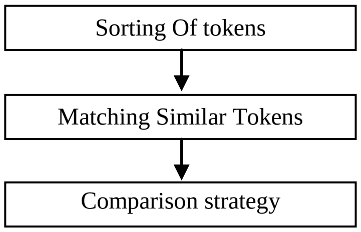

Figure (3) Duplicate Classifier module

Duplicate classifier module has three phases which are sorting of tokens, matching similar tokens and finally comparison strategy as illustrates in figure (3).

4.2.1 Sorting of tokens phase.

Token records obtained from the token table are sorted by any of the chosen OD. The purpose of sorting the tokens is that the potentially records will be in a close neighborhood, so records with exact match will be detected as duplicate.

4.2.2 Matching similar tokens phase.

The input of this phase is the token table resulting from the sorting phase. Table (2) shows which records are compared to detect the similarity between them.

Table (2) sorted Tokens table after deleting exact match records.

Rec. No.	***Author***	***title***	***Publish date***
1	Corats Eva	Oberon's Legcy	10/3/2001
2	Corets Eva	Oberon's Legacy	10/3/2001
3	Gambardella Matthew	XML Developer's Guide	1/10/2000
4	Ralls Kim	Midnight Rain	16/12/2000
5	Randall Cynthia	Lover Birds	2/9/2000

In odder to detect the similarity between XML's records all chosen OD are union to compose one compound token for example in record No.1 the compound token is Corats Eva Oberon's Legcy 10/3/2001 then all compound token are compared to measure their similarity using equation(1) where Dist is the NED. Then to compute the similarity using TBED, we partition each compound tokens into a list of tokens and repeat the pair-wise comparisons between these tokens of compound tokens using TBED and take the max (MaxLen-Dist) of every ingredient.

Let T1 corresponds to compound token of the first XML record, and T2 corresponds to compound token of the second XML record.

1. Partition compound token T_1 to a set of tokens (t_i, t_{i+1},………………….., t_n) and T_2 to a set of tokens (t_j, t_{j+1},………………….., t_m) where n and m are total the number of tokens in T_1 and T_2.
2. Make a pair-wise comparison between t_i and the set of tokens (tj, t_{j+1},………………….., t_m) and compute (MaxLen-Dist) for every pair of tokens.
3. Aggregate Max (MaxLen-Dist) of every result between t_i and (t_j, t_{j+1},………………….., t_m).
4. compute the final similarity between the two token sets use equation(2)

$$\frac{\sum(MaxLen - Dist(S_1, S_2))}{MaxLen(S_1, S_2)} \qquad (2)$$

Partition each string in the token table into list of tokens and compute the similarity between each two words in the token lists will help to get a better similarity between token lists. For example using equation(1) to calculate the NED to compute the similarity between" computer science faculty" and" faculty of computer science" will result in a very low similarity percentage which is 1-(19/24) = 0.208 which is not actually true but using TBED will result in a similarity equal to $\frac{8+7+7}{24}$ = 0.91.

Table(3) illustrates the final similarity between each two XML records, if we consider two XML records to be duplicate when their similarity exceed 0.75 , So we conclude that records one and two are duplicates and hence they will belong to the same cluster.

Table (3) Final similarity between XML records.

Rec. NO	***Author***	***title***	***Publish date***	***Final similarity***
1	Corats Eva	Oberon's Legcy	10/3/2001	.94
2	Corets Eva	Oberon's Legacy	10/3/2001	
1	Corats Eva	Oberon's Legcy	10/3/2001	.24
3	Gambardella Matthew	XML Developer's Guide	1/10/2000	
……………………………………………………………………………..……………				

4.2.2.1 Comparison strategy phase.

A full duplicate detection runs on XML data requires the detection of duplicates at every level of the hierarchy. The researchers present two traversal strategies that developed with efficiency in mind to prune expensive computations (mainly pair-wise comparisons) to reduce the overall duration of duplicate detection these traversal strategies are decreasing index filter and length filter:

4.2.2.2 Decreasing index (Removing Duplicate filter) (RDF)

Duplicate Detection using decreasing index filter requires comparing the first XML ODs with all XML ODs of the rest of XML records and once a duplicate record detected, remove the duplicate entity from comparison process.But before removing that entity, use the transitive relation to check if there are duplicate records with it to add them to the same cluster.

4.2.2.3 Length filter (LF filter)

Text with nearly the same length are likely to be duplicate than others ,so comparison is done first by sorting XML object description attribute text according to length ,then compare objects only within that length(i.e. prune all the comparisons that exceed certain threshold because they are not likely to be duplicate).

4.3 Duplicate Handling and cleaning module.

Candidate duplicates are produces in clusters of duplicate XML records and handled in many scenarios which are deleting the record, merging (unifying) more than one record into one record, updating record values, and Ignoring which means mark the two records as false positive.

5. Extended tasks of DRTX

DRTX is not only a duplicate detection and resolution system, but it has two more functionality first the ability to merge more than XML file, second Duplicate generator which is used to generate duplicate XML records according to certain criteria to be used as an input to DRTX tool to test its effectiveness more easily and accurate because duplicate records numbers and places is known in advance.

5.1 Merging XML Document.

DRIX may be used to merge more than XML file coming from different sources with different structure where a merging process is occurred to match the tags from the first file with its equivalent in the second file taking into consideration data standardization. Without standardization, many duplicates entries could be erroneously designated as no duplicates based on the fact that common identifying information cannot be compared.

5.2 Duplicates Generation Module.

Any XML duplicates detection and resolution system should have two types of dirty data, first duplicate data that are already exist in the real XML data set, second duplicate data that are artificially inserted into the data set:

Existing Duplicates: A dirty XML data coming from a real data set, the difficulty of implementing XML duplicates detection using this kind of data is to find and mark the duplicates in the source XML doc, which is a very hard and time consuming. Such a pre-knowledge of duplicates is very important in order to evaluate duplicates detection and resolution system using precision and recall.

Artificial duplicates: Duplicates are artificially injected into the end of the XML data source. The new generated duplicate record and the original records assigned a common identifier <OriginalID>.Duplicates are generated according to criteria's Exact Duplicate, Missing character, Swap character, Swap word and Insert character.

Authors in [22] consider many types of data contaminations as exact Duplicates, contradictory text, missing data, wrong reference constraints and different structure while authors in [11,12] use dirty xml generator available in the internet available by sven puhlmann.

Actually the tasks of duplicates generation module are not only to generate duplicate records but it is extended to be an evaluator thus it will have the following extended tasks:

1. Threshold determination.

XML duplicates generator module will determine the optimal threshold for similarity measure performance which will detect all duplicate records.

2. Help the user to choose object description (OD).

Statistics are collected on either the structure of XML records or on the content of XML elements. To help the user choose (OD) we use statistics on the produced Dirty XML file. The best object description is the objects have the greatest occurrence and exclude objects that have the null occurrence.

6. Comparison and analysis results

The dataset of our experiment was generated using our XML generator using many parameters which are dataset size, number of exact duplicate records, number of deleting characters, number of adding characters, number of swapping characters, number of swapping words and object descriptions.

The dataset size was set to 200 records , exact duplicate records was 40 records, the object descriptions was selected to be(author, title and publish date) and the rest of the parameters was increased from 1 to 5 thus we have 5 dirty XML dataset will be used by the detector as shown in table (4).

Table (4) Dataset setup

Data size	***Exact duplicate***	***Del char ,Add char ,Swap char ,Swap word***	***Object description***
200	40 record	1	author ,title, publish date
		2	author ,title, publish date
		3	author ,title, publish date
		4	author ,title, publish date
		5	author ,title, publish date

After generating the contaminated XML file we run XML duplicates detection module many times by changing the object description used in detection tool from 3OD (author, title and publish_date) to 2OD (author, title) and finally to 1OD (publish_date) to test the effect of the chosen object description(OD) on the efficiency and effectiveness.

We conduct experiments on dirty XML file using two approaches normal edit distance (NED) and enhanced token based damerau edit distance (TBED) and compare the effectiveness and efficiency on each approach.

Approaches of removing duplicates entities aims to improve effectiveness and efficiency so we use different metrics to test efficiency and effectiveness of the proposed algorithm:

6.1 Improving efficiency:

Efficiency is measured using the number of pair-wise comparisons (computational complexity) and the observed runtime of the algorithm.

6.1.1 Complexity matrix.

Computational complexity is measured using the number of pairwise comparisons. Full duplicates detection requires(n-1)!, but using the above mentioned filters the complexity(n-1)! is enhanced.

6.1.2 Runtime matrix.

Run time includes all the phases of the algorithm i.e. Reading data, performing duplicates detection and returning the results.

We measured the efficiency by duplicates detection time .when measuring efficiency we had three conclusions:

A. Length filter has the min. duplicates detection time in both NED and TBED as shown in figure (4) which signifies Duplicates detection time using all filters in TBED.

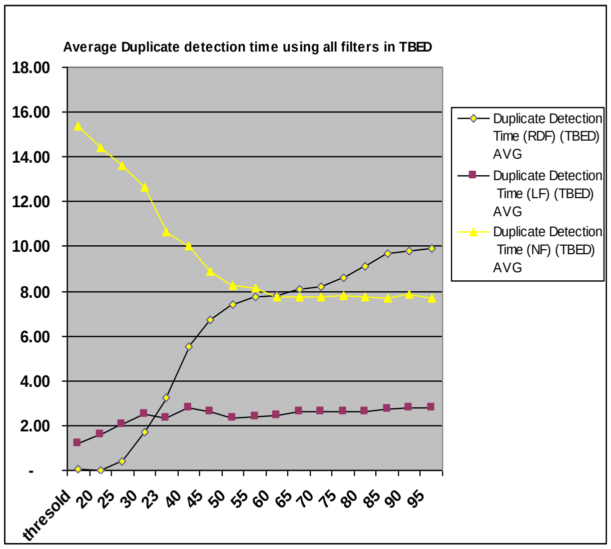

Figure (4) Average Duplicates detection time using all filters in TBED

B. TBED has min. duplicates detection time than NED using the two filters (RDF and LF) as illustrated in figures (5) which denotes Average duplicates detection time using removing duplicates filter (RDF).

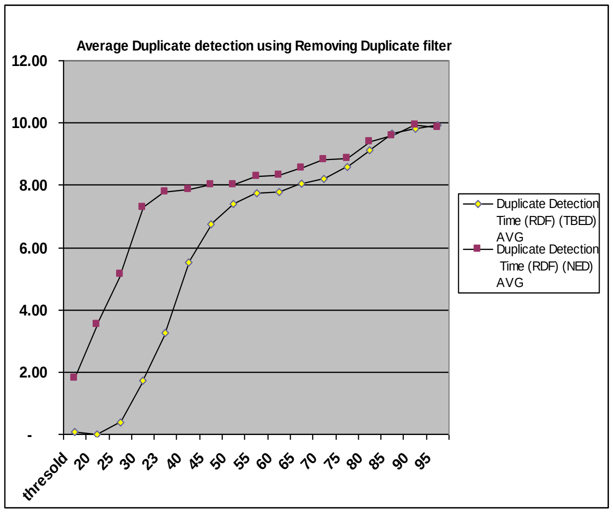

Figure (5) Average duplicates detection time using removing duplicates filter (RDF).

c) Using only one object description (OD) in both NED and TBED has the min. performance time when applying all filters.

6.2 Improving effectiveness:

In order to evaluate the quality of the duplicates clusters found by the clustering algorithms, we use several accuracy measures from the clustering literature and also measures that are suitable for the final goal of duplicates detection and resolution system (DRTX).Suppose that we have a set of k ground truth clusters of the base relation R. Let C= {c1, . . . , ck} denote the set of k′ output clusters of a clustering algorithm. We define a mapping f from the elements of g to the elements of C, such that each cluster g_i is mapped to a cluster $c_j = f(g_i)$ that has the highest percentage of common elements with g_i. Precision and recall for a cluster g_i, $1 \le i \le k$ is defined as follows [18]:

$$Pr_i = \frac{|f(g_i) \cap g_i|}{|f(g_i)|} \qquad (3)$$

and $$Re_i = \frac{|f(g_i) \cap g_i|}{|g_i|} \qquad (4)$$

Intuitively, the value of Pr_i is a measure of the accuracy with which cluster $f(g_i)$ reproduces cluster g_i, while the value of Re_i is a measure of the completeness with which $f(g_i)$ reproduces class g_i. Precision, Pr, and recall, Re, of the clustering are defined as the weighted averages of the precision and recall values over all ground truth clusters. More precisely:

$$Pr = \sum \frac{|g_i|}{|R|} pr_i \qquad (5)$$

$$, Re = \sum \frac{|g_i|}{|R|} Re_i \qquad (6)$$

and F1-measure is defined as the harmonic mean of precision and recall i.e. $F1 = \frac{2 \times Pr \times Re}{Pr + Re}$ (7)

To measure the effects of duplicates detection filters We run our duplicates detection and resolution system (DRTX) on each generated XML file many times once by 1OD another by 2OD and finally by 3OD and on each one of them we apply the 2 filters (decreasing index filter(RDF), length filter(LF)) and compare the results with no filter (NF) in which all pair-wise comparison are occurred so for each dirty XML file we have 9 results for each filter using NED and other 9 results for TBED for each of the chosen ODs (3 for each filter).

To get the final results we calculate average precision, recall and f-measure for each filter applied with each duplicates detection method (NED and TBED).Results show that length filter (LF) has the better precision up to 65 threshold similarity value either by using NED or TBED. But using values of similarity greater than 65 all filters are almost equal except the last three values of precision using NED in which also the length filter is superior.

As well , results show that Removing duplicates filter (RDF) has the better average recall up to 60 and 70 threshold values using NED and TBED respectively.

Experimental results show the superior of effectiveness of TBED algorithm using high threshold values with all filters. For example using length filter, TBED achieves better precision, recall and f-measures results after 65 similarity threshold as figure (6) illustrates Average F-measure using TBED &NED using Length filter .

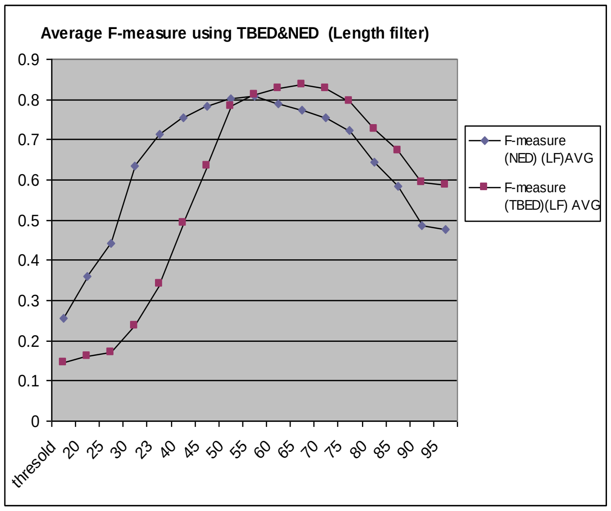

Figure (6) Average F-measure using TBED &NED using Length filter

6.3 Adding contributions

This work tried to solve the mentioned XML problems by adding the following contributions:

1. Mixed duplicate detection algorithm: which uses both NED and TBED algorithms. Experiments show that no algorithm is better than the other but both of them have its own use.

2. Generic Text similarity tool: This has the ability to add any text similarity methods not only NED and TBED.

3. General Duplicate detection and resolution tool (DRTX): We present a general duplicate detection tool used to detect and resolve duplicates in all types of data not only XML because all data types can easley converted to XML.

4. XML's documents integrator: which overcomes the problem of XML structural diversity by mapping XML elements to a user defined one.

5. Domain Independent: object description is chosen either manually or by produce statistics on XML file where the objects have the max. occurrence is more likely to be chosen as object description because the describe objects much better.

6. Dirty XML Generator: The input of the detector was artificially generated by generator, this makes the detection more reliable and easily to testing.

Conclusion

In this paper we present two different duplicates detection algorithms NED and TBED and try to min. the number of pair-wise element duplicates comparison by applying two filters RDF and LF .From experimental results we can conclude that there is no algorithm better than the other but each of them has it's own use ie.NED is better to use at lower threshold similarity values while TBED is better at higher ones so on that basis we have a mixed approach of duplicates detection and resolution uses NED in lower thresholds and TBED in higher thresholds that can be used with any kind of data that will be transformed to XML. Also the DRTX is very flexible that it can work with any algorithm of text similarity. Using the two pruning filters contribute in making a great enhancement in performance time where using LF with TBED contributes in decreasing performance time about 73% in average and using RDF with TBED contributes in decreasing performance time about 25% although there was not a greet decreasing in effectiveness.

Future work

Future work will include implementing data analyzer to choose which approach is more effective and accurate to the input XML data . Also we may use Metadata to handle synonyms to detect XML data that have the same meaning and manage different alternatives of attribute values.

Reference

[1] J. Jebamalar Tamilselvi and V. Saravanan,"Detection and Elimination of Duplicate Data Using Token-Based Method for a Data Warehouse: A Clustering Based Approach", Department of Computer Application, Karunya University, 2009.

[2] M.Weis and C.Markschies,"Duplicate Detection in XML Data",computing.dcu.ie ,Germany , 2007.

[3] Melanie Weis,"Fuzzy Duplicate Detection on XML Data" , Humboldt-University at zu Berlin Unter den Linden 6, Berlin, Germany,2005.

[4] Mong li lee ,Hongjun lu ,tok wang ling and yee teng ko, "cleansing data for mining and warehousing ", school of computing ,national university of Singapore,1999.

[5] Rohit Ananthakrishna , Surajit Chaudhuri and Venkatesh Ganti, "Eliminating Fuzzy Duplicates in Data Warehouses" , Cornell University, Microsoft Research ,2002.

[6] Jonathan D. Becher, Pavel Berkhin, Edmund Freeman, fermont, Automating Exploratory Data Analysis for Efficient Data Mining,2000.

[7] Vijayshankar Raman and Joseph M. Hellerstein. Potter'sWheel: An Interactive Data Cleaning System , University of California at Berkeley,2001

[8] Mikhail Bilenko and Raymond J. Mooney, "On Evaluation and Training Set Construction for Duplicate Detection" , Department of Computer Sciences University of Texas at Austin, 2003.

[9] L.Gravano,PG Ipeirotis,N.Koudas and D.Srivastava ,"Text Joins for Data Cleansing and Integration in an RDBMS", Columbia University , AT&T Labs– Research,2003 .

[10] Erhard Rahm and, Hong Hai Do, "Data Cleaning: Problems and Current Approaches ", University of Leipzig, Germany, 2000.

[11] Melanie Weis and Felix Naumann ,"DogmatiX Tracks down Duplicates in XML", Humboldt-Universität zu Berlin Unter den Linden 610099 Berlin , Germany, 2005.

[12] Melanie Weis and Felix Naumann,"Detecting duplicate objects in XML documents", Humboldt-Universität zu Berlin, germany,Workshop on Information Quality in Information Systems, 2004

[13] Melanie Weis1 and Ioana Manolescu. "Declarative XML Data Cleaning with Xclean", 2006.

[14] Melanie Weis and Ioana Manolescu , "Xclean in action", proceeding of: CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007.

[15] Luıś Leitão, Pável Calado and Melanie Herschel, "Efficient and Effective Duplicate Detection in Hierarchical Data," IEEE Transactions on Knowledge and Data Engineering, 09 Mar. 2012.

[16] Qiangfeng Peter Lau , Wynne Hsu , Judice L. Y. Koh and Mong Li Lee " DeepDetect An Extensible System for detecting attribute outliers and duplicates in XML", 2008

[17] Lwin, T , "An efficient Duplicate detection system for XML document " Computer Engineering and Applications (ICCEA), Second International Conference,2010.

[18] Oktie Hassanzadeh, Fei Chiang, Hyun Chul Lee and Ren´ee J. Miller, University of Toronto,"Framework for Evaluating Clustering Algorithms in Duplicate Detection",2009.

[19] Norbert Baumgartner, Wolfgang Gottesheim2, Stefan Mitsch2, Werner Retschitzegger3, and Wieland Schwinger Same, "same ,Same but Different:A Survey on Duplicate Detection Methods for Situation Awareness", team Communication Technology Mgt. Ltd., Goethegasse 3, 1010 Vienna, Austria Johannes Kepler University Linz, Altenbergerstr. 69, 4040 Linz, Austria 3University of Vienna, Dr.-Karl-Lueger-Ring 1, 1010 Vienna, Austria, 2009.

[20] Ahmed K. Elmagarmid , Panagiotis G. Ipeirotis and Vassilios S. Verykios, New York . "Duplicate Record Detection: A Survey", 2006.

[21] Melanie Weis and Felix Naumann, "Detecting Duplicates in Complex XML Data" , HumboldtUniversität zu Berlin, Germany,2006.

[22] Melanie Weis ,Felix Naumann ,and Franziska Brosy. "A Duplicate Detection Benchmark for XML and Relational Data", HumboldtUniversität zu Berlin, Germany, 2006.

Randa Mohamed received a computer science and information system diploma in 2006, During 2007-2009, she finished the pre-master in computer science and information system at Arab Academy for Science Technology & Maritime Transport. From 2006 to 2010 she worked as an oracle developer at ministry of finance(MOF). She is currently working as Asset management Project manager at MOF.

Prof. Dr. Dr.Ali H. El-Bastawissy is the director of the Center of Research and Development of Computers and Information Systems. Ali El-Bastawissy is a professor of information systems at Cairo University. He has written and published more than 50 articles and studies on Data Modeling and storage, Data Integration, Business Intelligence, Information Strategy, and domains. He has over 30 years experience designing and implementing business intelligence and enterprise integration solutions for dozens of Middle East Ministries and Governmental Associations that help them accelerate decision-making and improve corporate performance.

Prof. Dr. Mostafa Abdel Azeim Mostafa is a full professor and Dean of college of Computing and Information Technology, Arab academy of science & IT . He received his B. Sc with honor degree in computer engineering from MTC University, Egypt 1980, M. Sc degree in computer engineering from Faculty of Engineering Cairo University, Egypt 1984, and Ph. D degree from Cranfield University, U.K 1992. He is the author and reviewer of several books and papers in the field of Computing and Information Technology. Also, he is a member of scientific committee of International Conferences and Journals (IEEE, IARIA, IJCIS). His Current research interested is computing systems, software engineering and information technology applications.