INTERNATIONAL JOURNAL FOR DEVELOPMENT IN COMPUTER SCIENCE & TECHNOLOGY ISSN-2320-7884 (ONLINE)
VOLUME-1, ISSUE-IIIIS NOW AVAILABLE AT: www.ijdcst.com ISSN-2321-0257 (PRINT)

An Improvised Technique for Record Deduplication

Manasa Veena Dokku #1, A .Srinivasa Reddy#2

#1 Student, QIS College Of Engineering Technology, Ongole, Prakasam(dt)

#2 Assoc. professor, QIS College Of Engineering Technology, Ongole, Prakasam(dt)

Abstract: Deduplication is the key operation in data integration from multiple data sources. Duplicate record detection is important for data preprocessing and cleaning. Record linkage is the process of matching records from several databases that refer to the same entities. When applied on a single database, this process is known as deduplication. To achieve higher quality information and more simplified data representation, data preprocessing is required. Data cleaning is one among the data preprocessing steps. Removing duplicate records is crucial step in data cleaning process. Now-a-days in current databases, removing the duplicate records is more complex. This paper presents an analysis of record deduplication techniques and algorithms that detect and remove the duplicate records. In this paper, we proposed a new methodology which is divided two phases. They are: training phase and duplicate detection phase. These are again divided into four steps: (1) Similarity computation for all pair of records, (2) Computing feature vectors, (3) New similarity formulae generation and (4) Duplicate detection using the new similarity formulae. Our proposed system is very effective and efficient, when compared with remaining duplicate detection techniques.

I INTRODUCTION

Data mining is the extraction of hidden predictive information from large databases. It is a new powerful technology with great potential to help companies focus on the most important information in their data warehouses[1]. In real world, data mining technique is applicable to many areas like banking systems, educational systems, airline reservation systems etc. With the increase in size of the database the problem intensifies taking into account the huge amount of computational resource required for examination and removal of duplicate records.

Initially, record linkage techniques are used to link together records which relate to the same entity in one or more data sets where a unique identifier is not available. Each record in one data set has to be compared to all records in a second data set, the number of record pair comparisons grows quadratically with the number of records to be matched. This approach is computationally infeasible for large data sets. Most previous work is based on predefined matching rules hand-coded by domain experts or matching rules learned offline by some learning method from a set of training examples. To reduce the number of possible record pair comparisons, traditional record linkage techniques work in a blocking fashion, i.e. they use a record attribute (or sub-set of attributes) to split the data sets into blocks.

Data deduplication is important task in data cleaning and data integrity. Due to the duplicate records and dirty data, many problems will occur like performance degradation, quality loss and increasing operational costs. Data cleaning is the process of detecting and correcting in accurate records from a record set, table or data base. After cleaning, a data set will be consistent with other similar data sets in the systems. The inconsistencies detected or removed are originally caused by user entry errors or corruption in transmission. Record deduplication is the process of identifying and removing duplicate entries in a repository.

In this paper, we proposed a new methodology which is divided two phases. They are: training phase and duplicate detection phase. These are again divided into four steps: (1) Similarity computation for all pair of records, (2) Computing feature vectors, (3) New similarity formulae generation and (4) Duplicate detection using the new similarity formulae. Our proposed system is very effective and efficient, when compared with remaining duplicate detection techniques.

II UNSUPERVISED DUPLICATE DETECTION (UDD) TECHNIQUE

In this technique, there are two classifiers in UDD for iteratively identify the duplicate records[5]. The first classifier is called the Weighted Component Similarity Summing (WCSS) Classifier where the importance of the fields is determined and duplicates are identified without any training. The idea for this classifier is to calculate the similarity between pair of records by doing a field to filed comparison. This serves as input to the second classifier which is the Support Vector Machine (SVM) Classifier which makes use of the duplicates and non duplicates identified from the WCSS classifiers as training dataset. The SVM classifier then uses the training data and processes each record to identify/classify a record as being a duplicate or otherwise. These two classifiers iteratively working together and identify the duplicate records in efficient manner. The iteration stops when new duplicates cannot be found. This algorithm mainly used in the web databases because UDD does not require human labeled training data from the user. So it solves the online duplicate detection problem where the query results are generated on the-fly.

III DIVIDE AND CONQUER BASED DEDUPLICATION

In this approach[2], they first convert the attributes of data into numeric form. Then, this numeric form is used to create clusters by using K-Means clustering algorithm. The use of clustering reduces the number of comparisons. After that the divide and conquer technique is used in parallel with these clusters for identification and removal of duplicated records. Here, this technique identifies all type of duplicated records like fully duplicated records, erroneous duplicated records and partially duplicated records. This technique is only applicable for single table instead of multiple sorted tables. The performance is measured by using the terms like true positives, false positives, false negatives, precision, recall and F-Score.

IV ACTIVE LEARNING BASED DEDUPLICATION

This technique [3,4] automatically constructs the deduplication function by interactively finding the challenging training pairs. An active learner actively picks the subset of instances. It easing the deduplication task by limiting the manual effort for inputting simple, domain specific attributes similarity functions. It interactively labeling a small number of record pairs. First they took the small subset of pair of records. To improve the accuracy of classifier they selected only n instances from the pool of unlabeled data. Active learning requires some training data but in some real world problems the training data are not available, so active learning technique is not suitable for all the problems.

V PROPOSED METHODOLOGY

The proposed approach has two phases such as training phase and duplicate detection phase. These two phases are explained with the four different steps.

Step 1: Similarity computation for all pair of records:

In this step, the similarity computation is carried out by finding the similarity functions on each record field. Each function compares the similarity of each field with other record fields and assigns a similarity value for each field. Accurate similarity

functions are very important to calculate the distance between the records for better duplicate detection. Levenshtein distance and cosine similarity are the two similarity measures used in our proposed approach. Here, the input records are partitioned into two parts and the two measures are computed for the two parts of record pairs.

(1) *Levenshtein distance:* The chosen name fields of the records are "record 1" and "Record 2". The "Levenshtein distance" is computed by calculating the minimum number of operations that has to be made to transform one string to the other, usually these operations are: replace, insert or deletion of a character. The levenshtein distances between the records are found out by considering the record as a whole.

2) Cosine similarity: The cosine similarity between the two records name field "Record 1" and "Record 2" are calculated as follows: First, the dimension of both strings are obtained by taking the union of two string elements in the "record 1" and "record 2" then the frequency of occurrence vectors of the two elements are calculated.

Step 2: Computing feature vectors:

Feature vectors represent the set of elements that is required for the detection of duplicate elements from the data repository. The vectors can be obtained from the processing of the two similarity measure values. In general, the usual similarity functions may fail to find the similarity correctly, because the computation of similarity between fields can vary significantly depending on the domain and specific field under consideration. Therefore, it is necessary to adapt similarity measures for each field of the database with respect to the particular data domain for attaining accurate similarity computations. Consequently, we combine these similarity values obtained from different similarity measures to compute the distance between any two records. Here, we can represent similarity between any pair of records by a feature vector in which each component has the similarity value between two records of anyone of the similarity measure.

Step 3: New similarity formulae generation:

In this step, we consider the new formulae for the extraction of the feature vectors. An expression derived to calculate the fitness of the corresponding data. In order to find more precise output, i.e., to find the near duplicates better, we process a number of expressions. These expression, that we subject to process are used for the calculation of duplicates. A set of similar expression are supplied as input to find better among the supplied inputs. In this step, we find the best among the input expressions, which is capable of providing better solution for the problem.

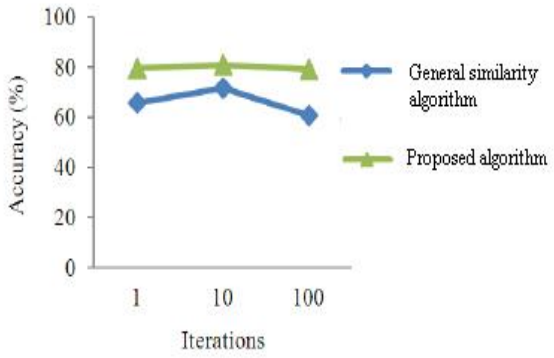
Step 4: Duplicate detection using the new similarity formulae:

Once the optimal similarity formulae are generated from the above step, the generated formulae is used to find the duplicate or nonduplicate records. Here, we fix the threshold, T to find the margin between duplicate and non-duplicate pairs.

VI PERFORMANCE

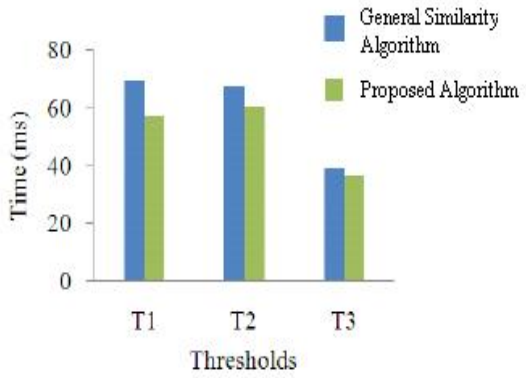
In proposed system, similarity computation is done by the similarity computation factors, listed in the above sections, such as Levenshtein distance method and cosine similarity method. The similarity factors produce feature vectors on regard with the elements in the dataset. The feature vectors produced are represented with variables <a, b, c, d>. The expressions are created from the feature vectors produced by the similarity vectors.

The proposed algorithm has the upper hand over the general similarity algorithm as shown in the below figure.



The above figure shows the accuracy of the two different algorithms on the basis of the number of iterations under specified threshold. It is evident that our proposed algorithms possess more accuracy on compared to the existing algorithm.

The below analysis is based on the time taken for the deduplication proposed by the proposed algorithms and the general similarity algorithm.



The above graph is plotted by varying the number of iterations under three threshold values. The analysis showed that the proposed algorithms, concept similarity based method and cosine similarity method consumes less time for the deduplication purpose than the genetic algorithm.

VII CONCLUSION

With the increase in size of the database the problem intensifies taking into account the huge amount of computational resource required for examination and removal of duplicate records. In this paper, we proposed a new methodology which is divided two phases. They are: training phase and duplicate detection phase. These are again divided into four steps: (1) Similarity computation for all pair of records, (2) Computing feature vectors, (3) New similarity formulae generation and (4) Duplicate detection using the new similarity formulae. Our proposed system is very effective and efficient, when compared with remaining duplicate detection techniques.

VIII REFERENCES

[1] Moises G. de Carvalho, Alberto H.F. Laender, Marcos AndreGoncalves, and Altigran S. da silva, "A Genetic Programming Approach to Record Deduplication", *IEEE Trans. Knowledge and Data Eng.,* vol. 24,no. 3, pp. 399-412, Mar. 2012.

[2] Bilal Khan, Azhar Rauf, Sajid H. Shah and Shah Khusro, "Identification and Removal of Duplicated Records", *World Applied Sciences Journal 13(5): ISSN 1818-4952,* pp.1178-1184, 2011.

[3] S. Sarawagi and A. Bhamidipaty, "Interactive Deduplication Using Active Learning", *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining(KDD'02),* pp.269-278, 2002.

[4] A.K. Elmagarmid, P.G. Ipeirotis, and V.S. Verykios, "Duplicate Record Detection: A Survey", *IEEE Trans. Knowledge and Data Eng.,* vol. 19, no. 1, pp. 1-16, Jan. 2007.

[5] Weifeng su, Jiying Wang, Frederick H. Lochovsky, " Record Matching over Query Results from Multiple Web Databases", *IEEETrans. Knowledge and Data Eng.,* vol. 22, no. 4, pp.578-588, April. 2010.

[6]A.Faritha Banu, C.Chandrasekar,"A Survey on Deduplication Methods", *International Journal of Computer Trends and Technology, ISSN: 2231-2803,* vol. 3, Issue. 3, pp.364-368, 2012

[7] Hamid Haidarian Shahri, Saied Haidarian Shahri, "Eliminating Duplicates in information Integration: An Adaptive, Extensible Framework", *IEEE Computer Society 1541-1672,* pp. 63-71, September/October 2006.