Handling Duplicate Data in Data Warehouse for Data Mining

Dr. J. Jebamalar Tamilselvi
Assistant Professor
Dept of Computer Applications
Jaya Eng College, Chennai

C. Brilly Gifta
Mphil (Research Scholar)
Dept of computer Applications
Bharathiyar University

ABSTRACT

The problem of detecting and eliminating duplicated data is one of the major problems in the broad area of data cleaning and data quality in data warehouse. Many times, the same logical real world entity may have multiple representations in the data warehouse. Duplicate elimination is hard because it is caused by several types of errors like typographical errors, and different representations of the same logical value. Also, it is important to detect and clean equivalence errors because an equivalence error may result in several duplicate tuples. Recent research efforts have focused on the issue of duplicate elimination in data warehouses. This entails trying to match inexact duplicate records, which are records that refer to the same real-world entity while not being syntactically equivalent. This paper mainly focuses on efficient detection and elimination of duplicate data. The main objective of this research work is to detect exact and inexact duplicates by using duplicate detection and elimination rules. This approach is used to improve the efficiency of the data.

Keywords

Data Cleaning, Duplicate Data, Data Warehouse, Data Mining

1. INTRODUCTION

Data warehouse contains large amounts of data for data mining to analyze the data for decision making process. Data miners do not simply analyze data, they have to bring the data in a format and state that allows for this analysis. It has been estimated that the actual mining of data only makes up 10% of the time required for the complete knowledge discovery process [3]. According to Jiawei, the precedent time consuming step of preprocessing is of essential important for data mining. It is more than a tedious necessity: The techniques used in the preprocessing step can deeply influence the results of the following step, the actual application of a data mining algorithm [6]. Hans-peter stated as the role of the impact on the link of data preprocessing to data mining will gain steadily more interest over the coming years. Preprocessing is one of the fourth future trend and major issues in data mining over the next years [7].

In data warehouse, data is integrated or collected from multiple sources. While integrating data from multiple sources, the amount of the data increases and as well as data is duplicated. Data warehouse may have terabyte of data for the mining process. The preprocessing of data is the initial and often crucial step of the data mining process. To increase the accuracy of the mining result one has to perform data preprocessing because 80% of mining efforts often spend their time on data quality. So, data cleaning is very much important in data warehouse before the mining process. The result of the data mining process will not be accurate because of the data duplication and poor quality of data. There are many existing methods available for duplicate data detection and elimination. But the speed of the data cleaning process is very slow and the time taken for the cleaning process is high with large amount of data. So, there is a need to reduce time and increase speed of the data cleaning process as well as need to improve the quality of the data.

There are two issues to be considered for duplicate detection: Accuracy and Speed. The measure of accuracy in duplicate detection depends on the number of false negatives (duplicates you did not classify as such) and false positives (non-duplicates which were classified as duplicates) [12].

In this research work, a duplicate detection and elimination rule is developed to handle any duplicate data in a data warehouse. Duplicate elimination is very important to identify which duplicate to retain and duplicate is to be removed. The main objective of this research work is to reduce the number of false positives, to speed up the data cleaning process reduce the complexity and to improve the quality of data. A high quality, scalable duplicate elimination algorithm is used and evaluated it on real datasets from an operational data warehouse to achieve objective.

2. DUPLICATE DETECTION AND ELIMINATION

In the duplicate elimination step, only one copy of exact duplicated records are retained and eliminated other duplicate records [4]. The elimination process is very important to produce a cleaned data. Before the elimination process, the similarity threshold values are calculated for all the records which are available in the data set. The similarity threshold values are important for the elimination process. In the elimination process, select all possible pairs from each cluster and compare records within the cluster using the selected attributes. Most of the elimination processes compare records within the cluster only. Sometimes other clusters may have duplicate records, same value as of other clusters. This approach can substantially reduce the probability of false mismatches, with a relatively small increase in the running time.

The following procedures are used to identify or detect duplicates and eliminate duplicates.

i. Get threshold value from LOG table
ii. Calculate certainty factor
iii. Calculate data quality factor for each record
iv. Detect or identify duplicates using certainty factor, threshold value and data quality factor
v. Eliminate duplicate record based on data quality, threshold value, number of missing value and range of each field value
vi. Retain only one duplicate record which is having high data quality, high threshold value and high certainty factor

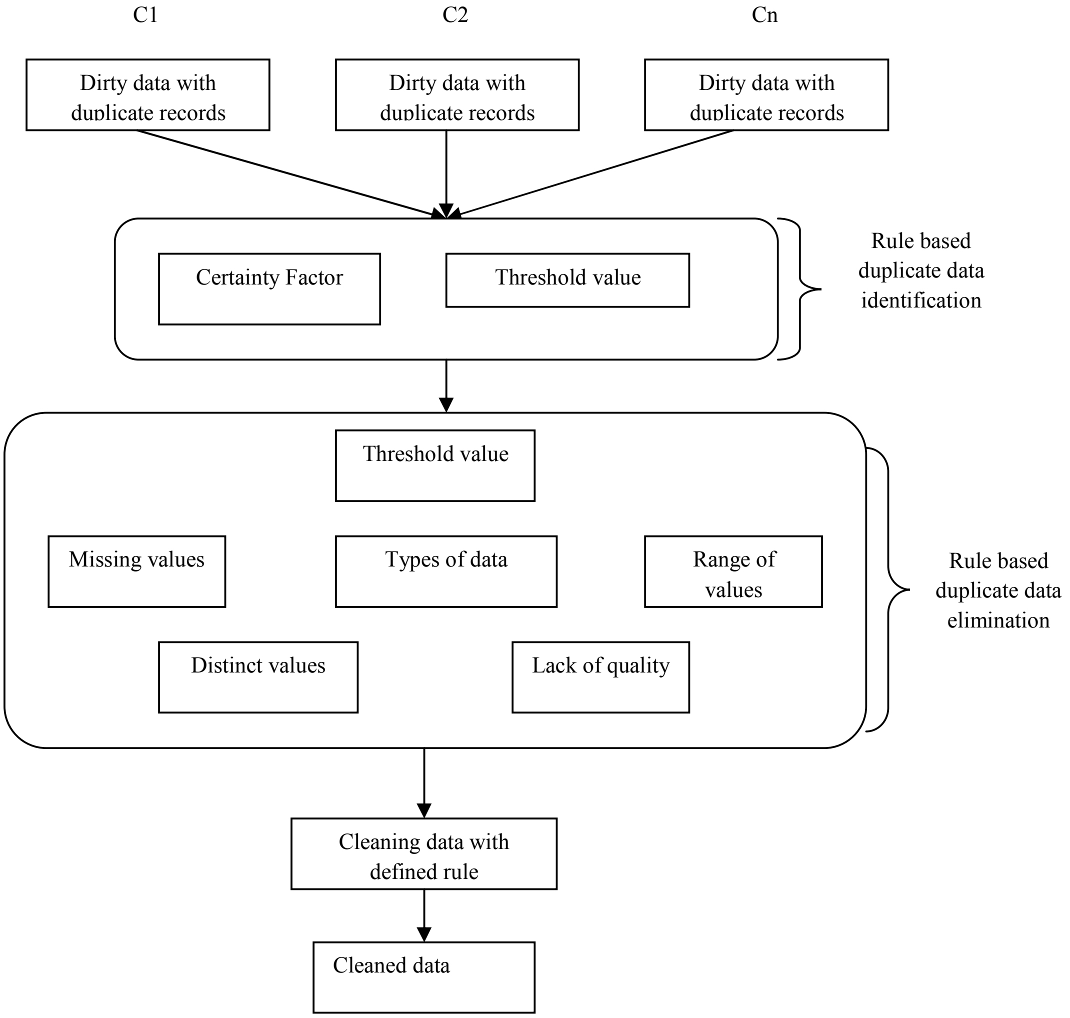
C1
C2
Cn
Dirty data with duplicate records
Dirty data with duplicate records
Dirty data with duplicate records
Certainty Factor
Threshold value
Rule based duplicate data identification
Threshold value
Missing values
Types of data
Range of values
Distinct values
Lack of quality
Rule based duplicate data elimination
Cleaning data with defined rule
Cleaned data

Figure 1. Framework for duplicate identification and elimination

Figure 1 shows the framework for duplicate data detection and duplicate elimination. There are two kinds of rules used in this framework. i) duplicate data identification rule ii) duplicate data elimination rule. Duplicate data identification rule is used to identify or detect duplicate using certainty factor and threshold value. Duplicate data elimination rule is used to eliminate duplicate data using certain parameters and used to retain only one exact duplicate data.

3. DUPLICATE DATA IDENTIFICATION / DETECTION RULE

Duplicate record detection is the process of identifying different or multiple records that refer to one unique real world entity or object if their similarity exceeds a certain cutoff value. However, the records consist of multiple fields, making the duplicate detection problem much more complicated [13]. A rule-based approach is proposed for the duplicate detection problem. This rule is developed with the extra restriction to obtain good result of the rules. These rules specify the conditions and criteria for two records to be classified as duplicates. A general if then else rule is used in this research work for the duplicate data identification and duplicate data elimination. A rule will generally be of the form:

if <condition >
then <action >

The action part of the rule is activated or fired when the conditions are satisfied. The complex predicates and external function references may be contained in both the condition and action parts of the rule [10]. In existing duplicate detection and elimination method, the rules are defined for the specific subject data set only. These rules are not applicable for another subject data set. Anyone with subject matter expertise can be able to understand the business logic of the data and can develop the appropriate conditions and actions, which will then form the rule set. In this research work, the rules are formed automatically based on the specific criteria and formed rules are applicable for any subject dataset. In duplicate data detection rule, threshold values of record pairs and certainty factors are very important.

3.1 Threshold value

Threshold value is calculated by identifying similarity between records and field values; that is, similarity value is used for calculating threshold value. In the similarity computation step, the threshold value is calculated and stored in the LOG table. In data elimination and identification step, the threshold value is extracted from the LOG table to identify and eliminate duplicate records. The threshold value is calculated for each field as well as for each record in each cluster. The cutoff threshold values are assigned for each attribute based on the types and importance of the attribute in the data warehouse. This cutoff threshold is used for identifying whether record is duplicated or not by using calculated threshold values of each record pairs.

3.2 Certainty factor

In the existing method of duplicate data elimination [10], certainty factor (CF) is calculated by classifying attributes with distinct and missing value, type and size of the attribute. These attributes are identified manually based on the type of the data and the most important of data in that data warehouse. For example, if name, telephone and fax field are used for matching then high value is assigned for certainty factor. In this research work, best attributes are identified in the early stages of data cleaning. The attributes are selected based on the specific criteria and quality of the data. Attribute threshold value is calculated based on the measurement type and size of the data. These selected attributes are well suited for the data cleaning process. Certainty factor is assigned based on the attribute types.

Table 1: Classification of attribute types

S. No	Key Attribute	Distinct values	Missing values	Size of data	Types of data
1	√			√	√
2	√			√	
3	√				√
4		√	√	√	√
5		√	√	√	
6		√	√		√
7		√		√	√
8		√	√		
9		√			√
10		√		√	

Table1 produces variety of attribute types. For example, an attribute may be a key attribute or a normal attribute. If normal attribute, attributes are classified with high distinct values, low missing values, high data type and high range of values. Each attribute can have any one of the type which is presented in Table 6a. Certainty factor is calculated based on the types of the attributes and calculation of certainty factor is listed below.

Rule 1: certainty factor 0.95 (No. 1 and No. 4)

- Matching key field with high type and high size
- and matching field with high distinct value, low missing value, high value data type and matching field with high range value

Rule 2: certainty factor 0.9 (No. 2 and No. 4)

- Matching key field with high range value
- and matching field with high distinct value, low missing value, and matching field with high range value

Rule 3: certainty factor 0.9 (No. 3 and No. 4)

- Matching key field with high type
- and matching field with high distinct value, low missing value, high value data type and matching field with high range value

Rule 4: certainty factor 0.85 (No. 1 and No. 5)

- Matching key field with high type and high size
- and matching field with high distinct value, low missing value and high range value

Rule 5: certainty factor 0.85 (No. 1 and No. 5)

- Matching key field and high size
- and matching field with high distinct value, low missing value and high range value

Rule 6: certainty factor 0.85 (No. 2 and No. 5)

- Matching key field with high type
- and matching field with high distinct value, low missing value and high range value

Rule 7: certainty factor 0.85 (No. 1 and No. 6)

- Matching key field with high size and high type
- and matching field with high distinct value, low missing value and high value data type

Rule 8: certainty factor 0.85(No. 2 and No. 6)

- Matching key field with high size
- and matching field with high distinct value, low missing value and high value data type

Rule 9: certainty factor 0.85 (No. 3 and No. 6)

- Matching key field with high type
- and matching field with high distinct value, low missing value and high value data type

Rule 10: certainty factor 0.8 (No. 1 and No. 7)

- Matching key field with high type and high size
- and matching field with high distinct value, high value data type and high range value

Rule 11: certainty factor 0.8 (No. 2 and No. 7)

- Matching key field with high size
- and matching field with high distinct value, high value data type and high range value

Rule 12: certainty factor 0.8 (No. 3 and No. 7)

- Matching key field with high type
- and matching field with high distinct value, high value data type and high range value

Rule 13: certainty factor 0.75 (No. 1 and No. 8)

- Matching key field with high type and high size
- and matching field with high distinct value and low missing value

Rule 14: certainty factor 0.75 (No. 2 and No. 8)

- Matching key field with high size
- and matching field with high distinct value and low missing value

Rule 15: certainty factor 0.75 (No. 3 and No. 8)

- Matching key field with high type
- and matching field with high distinct value and low missing value

Rule 16: certainty factor 0.7 (No. 1 and No. 9)

- Matching key field with high type and high size
- and matching field with high distinct value and high value data type

Rule 17: certainty factor 0.7 (No. 2 and No. 9)

- Matching key field with high size
- and matching field with high distinct value and high value data type

Rule 18: certainty factor 0.7 (No. 3 and No. 9)

- Matching key field with high type
- and matching field with high distinct value and high value data type

Rule 19: certainty factor 0.7 (No. 1 and No. 10)

- Matching key field with high type and high size
- and matching field with high distinct value and high range value

Rule 20: certainty factor 0.7 (No. 2 and No. 10)

- Matching key field with high size
- and matching field with high distinct value and high range value

Rule 21: certainty factor 0.7 (No. 3 and No. 10)

- Matching key field with high type
- and matching field with high distinct value and high range value

Duplicate data is identified based on the certainty factor, threshold value and quality data. Quality data is combined in the certainty factor. So, certainty factor and threshold values are very important in data elimination. Certainty factor and threshold value are calculated based on the defined rule listed in Table 2. In rule 1, if the attribute has high distinct values (D), low missing values (M), high measurement type (DT) and high data range, then the highest certainty factor (0.95) is assigned for attributes and less threshold value (0.75) is enough for comparing records. Because comparing attributes should have high quality of values for duplicate detection.

Certainty factor is calculated based on the quality of data. If an attribute has high certainty factor value, then less threshold value is assigned in record comparison. If then else condition is used to identify duplicate value. For example,

```
If CF=0.95 and TH>0.75 then
        Records are duplicates.
End
```

Table 2: Values of Certainty factor and Threshold value

S.No	Rules	Certainty Factor (CF)	Threshold value (TH)
1	{TS}, {D, M, DT, DS}	0.95	0.75
2	{T, S}, {D, M, DT, DS}	0.9	0.80
3	{TS, T, S}, {D, M, DT}, {D, M, DS}	0.85	0.85
4	{TS, T, S}, {D, DT, DS}	0.8	0.9
5	{TS, T, S}, {D, M}	0.75	0.95
6	{TS, T, S}, {D, DT}, {D, DS}	0.7	0.95

TS – Type and Size of key attribute

T – Type of key attribute

S – Size of key attributes

D – Distinct value of attributes

M – Missing value of attributes

DT – Data type of attributes

DS – Data size of attributes

Duplicate records are identified in each cluster to identify exact and inexact duplicate records. The duplicate records can be categorized as match, may be match and no-match. Match and may be match duplicate records are used in the duplicate data elimination rule. Duplicate data elimination rule will identify the quality of the each duplicate record to eliminate poor quality duplicate records.

3.3 Duplicate data elimination rule

Typically duplicate data elimination is performed as the last step and this step has to take place while integrating two sources or performed on an already integrated source. The combination of attributes can be used to identify duplicate records. In the duplicate elimination, only one best copy of duplicate record has to be retained and remaining duplicate records should be eliminated. Correct duplicate records are identified using certainty factor and threshold value. Duplicate data is eliminated based on the number of missing value, range of each field value, data quality of each field value and representation of data. Each duplicate data or overall similarities of two records are determined from the similarities of selected record fields. An example of duplicate data elimination is the rule that two records with (i) identical field value (ie) high threshold value (ii) are of the same length, and (iii) belong to the same type of data, are duplicates. The rule can be represented as: high threshold value ^ same length of field value ^ same representation of data with slight changes → not duplicate

The following factors are used in the duplicate data elimination rule.

i. Number of missing values
ii. Range of values
iii. Data representation
iv. Lack of quality
v. Threshold value

Duplicate records are identified by using specific and high discrimination power attributes. In general, duplicate records can have so many missing fields. Hence, records can be eliminated based on the number of missing values in each duplicate record. Duplicate record is eliminated if the duplicate record is has more missing values than other duplicate records. The size and range of each field value is calculated and compared with other duplicate data field to eliminate poor quality duplicate data. For example, sometimes duplicate data can have shortcut form or abbreviation. So, the range of each field value is calculated to remove duplicate data which have low range than other duplicate records. Most of the time record is duplicated because of the different format used for data representation. For example, 'M' and 'F' are used for male and female but '1' and '0' are used for gender representation. So, there is a need to identify exact format for each field representation. Wrong format of duplicate data can be eliminated to retain best and high quality duplicate data. Data warehouse may contain data with poor quality. Data with poor quality must be removed or to be changed while eliminating duplicate data. Mostly, duplicate records are identified and eliminated based on the threshold value of each duplicate record. Highest threshold value duplicate record is retained and lowest threshold value duplicate records are eliminated.

In this research work, rules are developed to eliminate poor quality duplicate data based on the missing values, length, format, quality of the data and threshold value. For example, two records are identified as duplicates. In these two records, one record has higher missing values than the other record. Hence, duplicate record with high number of missing value is eliminated and other duplicate record is retained. If both duplicate records have the same number of missing values, then it checks length, format and quality of the data for duplicate data elimination.

Rule 1	Missing(high)=1.0
Rule 2	Size or length(low)=1.0
Rule 3	Format(low)=1.0
Rule 4	Quality(low)=1.0
Rule 5	Threshold(low)=1.0
Rule 6	Missing(high)=1.0 ^ Size or length(low)=1.0
Rule 7	Missing(high)=1.0 ^ Format(low)=1.0
Rule 8	Missing(high)=1.0 ^ Quality(low)=1.0
Rule 9	Missing(high)=1.0 ^ Threshold(low)=1.0
Rule 10	Size or length(low)=1.0 ^ Format(low)=1.0
Rule 11	Size or length(low)=1.0 ^ Quality(low)=1.0
Rule 12	Size or length(low)=1.0 ^ Threshold(low)=1.0
Rule 13	Format(low)=1.0 ^ Quality(low)=1.0
Rule 14	Format(low)=1.0 ^ Threshold(low)=1.0
Rule 15	Threshold(low)=1.0 ^ Quality(low)=1.0
Rule 16	Missing(high)=1.0 ^ Length(low)=1.0 ^ Format(low)=1.0
Rule 17	Missing(high)=1.0 ^ Length(low)=1.0 ^ Quality(low)=1.0
Rule 18	Missing(high)=1.0 ^ Length(low)=1.0 ^ Threshold(low)=1.0
Rule 19	Missing(high)=1.0 ^ Length(low)=1.0 ^ Size or length(low)=1.0
Rule 20	Missing(high)=1.0 ^ Format(low)=1.0 ^ Quality(low)=1.0
Rule 21	Missing(high)=1.0 ^ Format(low)=1.0 ^ Length(low)=1.0
Rule 22	Missing(high)=1.0 ^ Format(low)=1.0 ^ Threshold(low)=1.0
Rule 23	Missing(high)=1.0 ^ Threshold(low)=1.0 ^ Quality(low)=1.0
Rule 24	Missing(high)=1.0 ^ Threshold(low)=1.0 ^ Length(low)=1.0
Rule 25	Missing(high)=1.0 ^ Quality(low)=1.0 ^ Length(low)=1.0
Rule 26	Missing(high)=1.0 ^ Length(low)=1.0 ^ Format(low)=1.0 ^ Length(low)=1.0
Rule 27	Missing(high)=1.0^Length(low)=1.0^Format(low)=1.0 ^ Threshold(low)=1.0
Rule 28	Missing(high)=1.0 ^ Length(low)=1.0 ^ Format(low)=1.0 ^ Quality(low)=1.0
Rule 29	Missing(high)=1.0^Format(low)=1.0^Quality(low)=1.0 ^ Threshold(low)=1.0

Rule 30	Missing(high)=1.0 ^ Format(low)=1.0 ^ Quality(low)=1.0 ^ Length(low)=1.0
Rule 31	Missing(high)=1.0^Threshold(low)=1.0^Quality (low)=1.0 ^ Length(low)=1.0
Rule 32	Missing(high)=1.0^ Length(low)=1.0 ^ Threshold(low)=1.0 Quality(low)=1.0 ^ Format(low)=1.0

Table 3: Rules for duplicate data elimination

Table 3 produces rules for duplicate data elimination. Different criteria are assigned for each rule. Each rule has its highest and lowest priority than other rules. Based on the priority of the rule, the duplicate records are eliminated. The rules are used to delete duplicate records and to retain only one copy of exact duplicate record based on the defined factors. The above factors are used in a set of restricted values on each of the matching criteria, which can be calculated for data elimination. After data elimination, the quality data are merged in the next stage.

4. DATASET ANALYSIS

Two datasets are used namely Customer dataset and Student dataset in this research work for result analysis. The customer data set is a synthetic data set containing 720 customer-like Records and 17 attributes with 287 duplicates. The attributes are CUSTOMER CREDIT ID, CUSTOMER NAME, CONTACT FIRST NAME, CONTACT LAST NAME, CONTACT TITLE, CONTACT POSITION, LAST YEAR'S SALES, ADDRESS1, ADDRESS2, CITY, REGION, COUNTRY, POSTAL CODE, E-MAIL, WEB SITE, PHONE and FAX.

Also, for this research work a real world 10, 00,000 student's records from college data set are collected from multiple departments having same schema definition for executing duplicate detection process. The data set uses numeric, categorical and date data type. Address table consist of total 22 attributes. The attributes are ID, ADD_MNAME, ADD_LNAME, ADD_DISTRICT,ADD_ADDRESS3, ADD_ADDRTYPE, ADD_CUSER,ADD_MDATE, ADD_MUSER, ADD_ADDRESS4, ADD_EMAIL, ADD_FAX, ADD_MOBILE, ADD_ADDRESS1, ADD_PHONE2, ADD_NAME, ADD_ADDRESS2, ADD_PHONE1, ADD_CDATE, ADD_DEL, ADD_PARENTTYPE and ADD_PINCODE. Out of that 6 important attributes ADD_FNAME, ADD_ADDRESS1, ADD_ADDRESS2, ADD_PINCODE, ADD_PHONE, and ADD_CDATE are used for duplicate detection process. Efficiency of duplicate detection and elimination is largely depends on the selection of attributes.

5. EXPERIMENTAL RESULTS

To evaluate the performance of this research work, the errors introduced in duplicate records range from small typographical changes to large changes of some fields. Generally, the database duplicate ratio is the ratio of the number of duplicate records to the number of records of the database. To analyze the efficiency of this research work, proposed approach is applied on a selected data warehouse with variant window size, database duplicate ratios and database sizes. In all test, the time taken for duplicate data detection and elimination process is analyzed to evaluate the efficiency of time saved in this research work.

5.1 Duplicates Vs Threshold Values – Student Dataset

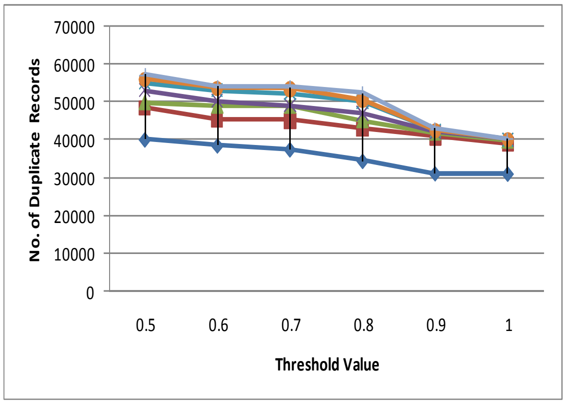

Figure 2: Duplicate detected Vs varying threshold value

ATTRIBUTES	0.5	0.6	0.7	0.8	0.9	1
ADD_ADDRESS1	40153	38511	37462	34423	31048	30988
ADD_ADDRESS1 ADD_NAME	48523	45263	45139	42874	40743	38944
ADD_ADDRESS1 ADD_NAME ADD_PHONE1	49898	48990	48987	45214	41985	39860
ADD_ADDRESS1 ADD_NAME ADD_PHONE1 ADD_CDATE	52984	**50234**	49136	46987	42035	40128
ADD_ADDRESS1 ADD_NAME ADD_PHONE1 ADD_CDATE ADD_DEL	55133	53134	52456	50156	42542	40156
ADD_ADDRESS1 ADD_NAME ADD_PHONE1 ADD_CDATE ADD_DEL ADD_ADDRESS2	55982	53634	53452	**50598**	42590	40160
ADD_ADDRESS1 ADD_NAME ADD_PHONE1 ADD_CDATE ADD_DEL ADD_ADDRESS2 ADD_PINCODE	57213	54136	53874	52345	42898	40172

Table 4: Attributes and threshold value

Figure 2 shows the performance of duplicate detection with false mismatches by varying the threshold value. Table 4 contains number of duplicates detected by each combination of attributes for particular threshold value. The data values shown in bold letters represent the total number of duplicate records detected at optimum threshold value. The optimum threshold values are 0.6, 0.7 and 0.8. The results are not accurate if the threshold value is greater than or less than the optimum threshold value. The number of mismatches and false mismatches are increased and it is not possible to detect exact and inexact duplicates. Hence, the accuracy of the duplicate detection is not exact. The threshold value is an important parameter for duplicate detection process.

5.2 Duplicates Vs Threshold Values – Customer Dataset

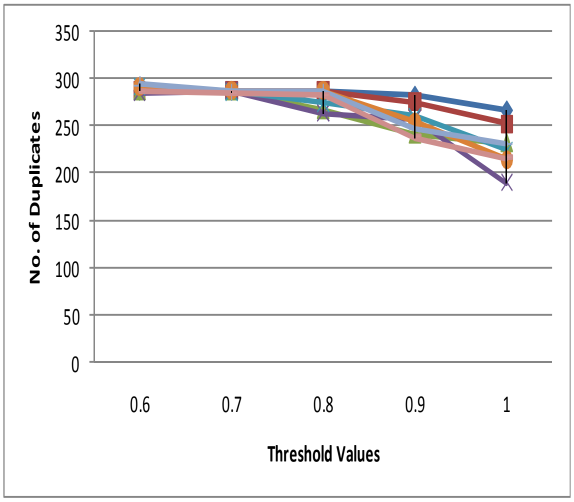

Figure 3: Duplicate detected Vs varying threshold value

ATTRIBUTES	0.6	0.7	0.8	0.9	1
PHONE	287	287	286	282	267
PHONE FAX	287	287	287	275	252
PHONE FAX POSTAL_CODE	287	287	267	241	232
PHONE FAX POSTAL_CODE CUSTOMER_NAME	286	287	263	257	190
PHONE FAX POSTAL_CODE CUSTOMER_NAME E_MAIL	292	287	274	261	225
PHONE FAX POSTAL_CODE CUSTOMER_NAME E_MAIL CONTACT_LAST_NAME	291	287	287	254	215
PHONE FAX POSTAL_CODE CUSTOMER_NAME E_MAIL CONTACT_LAST_NAME CITY	294	286	286	247	231
PHONE FAX POSTAL_CODE CUSTOMER_NAME E_MAIL CONTACT_LAST_NAME CITY ADDRESS1	287	285	283	238	216

Table 5: Attributes and threshold value

Figure 3 shows the performance of exact duplicate detection by varying the threshold value. Table 5 contains number of duplicates detected by each combination of attributes for particular threshold value. The exact duplicate value is 287 in selected customer dataset. Figure 6c shows that threshold values 0.6, 0.7 and 0.8 are optimum threshold values for duplicate detection. The results are not accurate if the threshold value is greater than or lesser than the optimum threshold value. Generally, data warehouse contains inexact duplicates with slight changes because of the typographical errors. Hence, the threshold value is an important parameter for duplicate detection process. If the threshold value is 1, then the detection of duplicates is very low because it takes only exact duplicate values. If the threshold value is less than 0.5, then the detection of duplicates is very high but the ratio of false mismatches is increased.

5.3 Percent of incorrectly detected duplicated pairs Vs Attributes

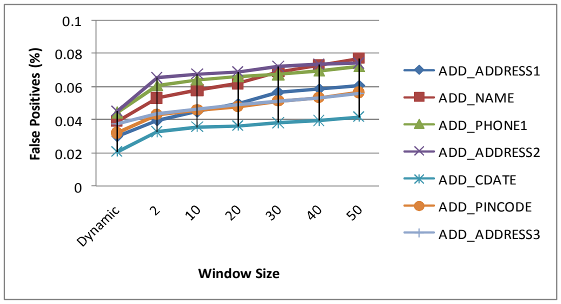

Figure 4: False Positives Vs Window size and Attributes

Figure 4 shows that false positive ratio is increased as increases the window sizes. Identification of duplicates depends on keys selected and size of the window. This figure shows the percent of those records incorrectly marked as duplicates as a function of the window size. The percent of false positive is almost insignificant for each independent run and grows slowly as the window size increases. The percentage of false positives is very slow if the window size is dynamic. This suggests that the dynamic window size is more accurate than fixed size window in duplicate detection.

5.4 Time taken on databases with different dataset size

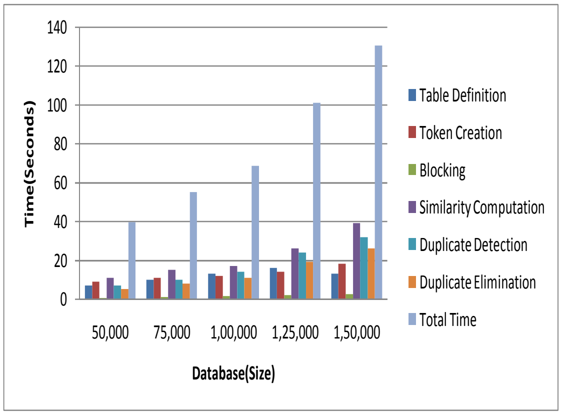

Figure 5: Time taken Vs Database size

Figure 5 shows variations of time taken in different database sizes. The result on time taken by each step of this research work is shown in figure 5. The time taken by proposed research work increases as database size increases: the time increases when the duplicate ratio increases in dataset. The time taken for duplicate data detection and elimination is mainly dependent upon size of the dataset and duplicate ratio in dataset. The efficiency of time saved is much larger than existing work because token based approach is implemented to reduce the time taken for cleaning process and improves the quality of the data.

5.5 Time performance Vs Different size databases and Percentage of created duplicates

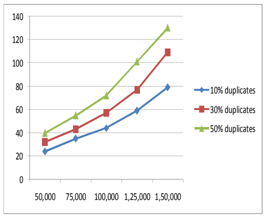

Figure 6: Time Vs Database size and % of Duplicates

10%, 30% and 50% duplicates are created in the selected datasets for result analysis. The results are shown in Figure 6. The time taken for duplicate detection and elimination is varied based on the size of the data and percentage of duplicates available in the dataset. For these relatively large size databases, the time seems to increase linearly as the size of the databases increase is independent of the duplicate factor.

6. CONCLUSION

Deduplication and data linkage are important tasks in the pre-processing step for many data mining projects. It is important to improve data quality before data is loaded into data warehouse. Locating approximate duplicates in large data warehouse is an important part of data management and plays a critical role in the data cleaning process. In this research wok, a framework is designed to clean duplicate data for improving data quality and also to support any subject oriented data.

In this research work, efficient duplicate detection and duplicate elimination approach is developed to obtain good result of duplicate detection and elimination by reducing false positives. Performance of this research work shows that the time saved significantly and improved duplicate results than existing approach.

The framework is mainly developed to increase the speed of the duplicate data detection and elimination process and to increase the quality of the data by identifying true duplicates and strict enough to keep out false-positives. The accuracy and efficiency of duplicate elimination strategies are improved by introducing the concept of a certainty factor for a rule. Data cleansing is a complex and challenging problem. This rule-based strategy helps to manage the complexity, but does not remove that complexity. This approach can be applied to any subject oriented data warehouse in any domain.

7. REFERENCES

[1] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, Vassilios S. Verykios (January 2007), Duplicate Record Detection: A Survey, IEEE Transactions on Knowledge and Data Engineering, Volume 19, NO. 1.

[2] 2. Bilenko, M., Mooney, R.J (August 2003), Adaptive Duplicate Detection Using Learnable String Similarity Measures, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), Washington, DC.

[3] 3. Dorian Pyle (1999), Data Preparation for Data Mining, Published by Morgan Kaufmann, ISBN 1558605290, 9781558605299, 540 pages.

[4] 4. A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios (2007), Duplicate Record Detection: A Survey, IEEE TKDE, 19(1):1-16.

[5] 5. Feekin A. and Z. Chen (2000), Duplicate detection using K-way sorting method, Proc. ACM SAC Conference, pages 323-327.

[6] 6. Hui Xiong and Gaurav Pandey and Michael Steinbach and Vipin Kumar (2006), Enhancing Data Analysis with Noise Removal, IEEE Transactions on Knowledge and Data Engineering, IEEE Computer Society, volume 18, page no 304-319.

[7] Hans-peter Keriegel, Karsten M. Borgwardt, Peer Kroger, Alexey Pryakhin, Matthias Schubert, Arthur Zimek (2007), Future trends in data mining, Data Mining and Knowledge Discovery, Volume 15 , Issue 1, Pages: 87 – 97, ISSN:1384-5810.

[8] 8. Jiawei Han, Micheline Kamber (March 2006), Data Mining: Concepts and Techniques, Publisher: Elsevier Science & Technology Books, ISBN-13: 9781558609013.

[9] Judice L.Y.Koh, Mong Li Lee, Asif M. Khan, Paul T.J. Tan and Vladimir (September 24,2004), Duplicate Detection in Biological Data using Association Rule Mining, 2nd European Workshop on Data Mining and Text Mining for Bioinformatics, Pisa, Italy.

[10] 10. Lup Low W.; Li Lee M.; Wang Ling T.(December 2001), A knowledge-based approach for duplicate elimination in data cleaning, Information Systems, Volume 26, Issue 8, pp. 585-606(22), Publisher: ELSEVIER, ISSN:0306-4379.

[11] Partrick Lehti(2006), Unsupervised Duplicate Detection Using Sample Non-duplicates, Lecture Notes in Computer Science, NUMB 4244, pages 136-164.

[12] 12. Robert Leland (August 2007), Duplicate Detection with PMC – A Parallel Approach to Pattern Matching Department of Computer and Information Science, Norwegian University of Science and Technology, Ph.D. Thesis.

[13] 13. Shen H, Zhang Y.(Nov. 2008), Improved approximate detection of duplicates for data streams over sliding windows, Journal of Computer Science and Technology, Volume 23(6), pp. 973-987.