www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 1 January 2015, Page No. 9941-9946

Data Cleaning System to Handle Noisy Data

A.F.Elgamal

CS Dep., Mansoura University
amany_elgamal@mans.edu.eg

Abstract: ***Data cleaning techniques are used for identification of record duplicates, missing data, and duplicate elimination. This paper presents a data cleaning system, it goes through six steps: selection of attributes, formation of tokens, clustering algorithm, similarity computation, elimination function, and finally merge step. The system architecture contains three components: users interface, data cleaning, and reports component where they can communicate and cooperate with each other's. It is implemented using SQL Server 2010 and Microsoft visual c# 2010.***

Keywords: Data cleaning, attribute selection, clustering algorithm, similarity computation, elimination function**.**

1. Introduction

The quality of data needs to be improved by using the data cleaning techniques. Data cleaning process includes detection and removing of errors and inconsistencies from data [1]. Existing data cleaning techniques are used to identify record duplicates, missing values, record and field similarities, and duplicate elimination [2]. Errors in data can often be found when multiple sources of data are merged [3]. Data cleaning monitoring is an incessant activity starting from data gathering stage and continues until the ultimate choice of analysis and interpretation of results. The importance of data cleaning and data quality is increasingly clear, as evidenced by software, tools, consulting companies, and seminars addressing data quality issues [4]. The classical application of data cleaning occurs in data warehouses. Data warehouses are generally used to provide analytical results from multidimensional data through effective summarization and processing of segments of source data relevant to the specific analysis. Business data warehouses are basis of Decision Support Systems DSS which provide analytical results to officials so that they analyze a situation and make important decisions. Cleanliness and integration of data contribute to the accuracy and correctness of results and affect the impact of any decision made or conclusion drawn [5]. Moreover, the records processed within different database systems may have different data formats or representations [6]. When such databases are merged, two records referring to the same entity may not match, resulting in duplicate entries or missing data.

The problem of detecting and eliminating duplication in data is one of major problems in broad area of data cleaning and data quality [7]. Duplicate elimination is a hard task, because it is caused by several types of errors. Most existing approaches rely on tuned distance metrics to estimate the similarity of potential duplicates [8]. The goal of record matching (duplicate detection) and deduplication is to identify the matching records, defined records that correspond to the same real-world entity. The output of record matching (duplicate detection) is pairs of matching records while the output of deduplication is clusters of matching records. Current cleaning techniques are very domain-specific and hard to extend, hindering their use in some areas [9].

This paper proposes the implementation and application of a flexible data cleaning system. Section 2 illustrates the steps of the used framework; proposed system components and implementation are illustrated in Section 3; experimental results are presented in section 4, and finally the conclusion is presented in Section 5.

2. Framework Design

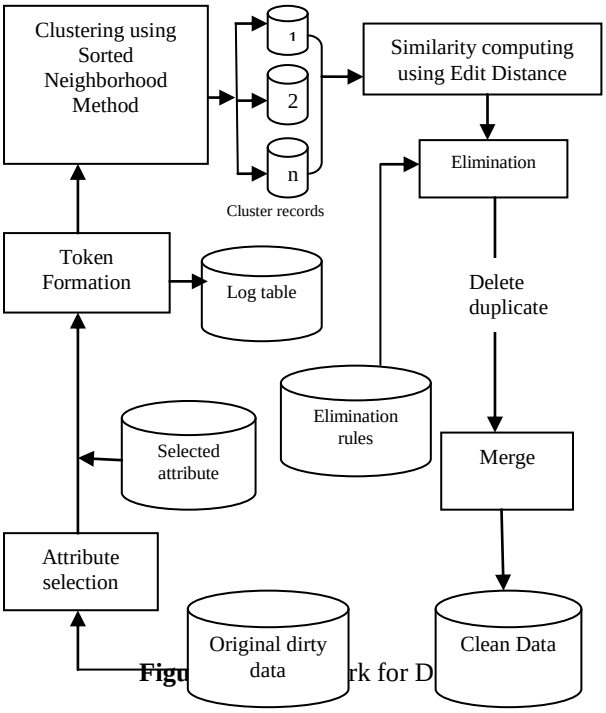
The Data Cleaning Framework is designed with flexibility as the central idea. It offers the user interaction by selecting the suitable algorithm. The framework steps are illustrated in figure1, they are as follows [10]:

A. Selection of attributes
B. Formation of tokens
C. Clustering algorithm
D. Similarity computation for selected attributes
E. Selection of elimination function
F. Merge.

A. **Selection of attributes**

Attribute selection is one of the most important and frequently techniques which are used in data preprocessing. It reduces number of attributes by removing irrelevant attributes, which are insignificant in the classification task [11].

A good attribute variable for clustering should contain a large number of values that are fairly uniformly distributed; and such an attribute must have low probability of reporting errors. For example, an attribute named gender has two values only and can't impart information to determine a match uniquely. Conversely, an attribute named surname imparts much more information, but it may frequently be recorded incorrectly [12].



Figurk for D

Attributes are ranked based on the threshold value. There are three criteria used to identify high threshold value of attributes: Identifying key attributes, classifying attributes with high distinct value and low missing value, and classifying the types of attributes [13].

Key attribute (or a set of attributes) is that uniquely identifies a specific instance of the dataset. The characteristics of key attributes are: non-null value, unique, and not change or become null during the life of each entity instance.

Distinct value is used to retrieve a number of tuples that have unique values for each attribute, as well as to calculate an identification power of the attribute. For example, the address attribute is effective to identify duplicate records because these attributes have many distinct values; while the salary attribute is not effective to identify duplicate records, because it has many similar values. A missing value is expressed and treated as a string of blanks, which means that some variables do not have a measurement. The following equation identifies the attribute's power [14]:

Power of the attribute=

$$\frac{\text{Number of distinct equivalence classes on the total of record}}{\text{Total number of records}}$$

The value of measurement types are considered for the attribute selection. The threshold value is the average of: type value, distinct, and not null. Calculating the data type value depends on the data type. The following equations identify the string and numeric data type:

String type value=

$$\frac{\text{Count of records that have maximum limit of characters}}{\text{Total number of records}}$$

Numerical type value=

$$\frac{\text{Count of records that have value between max limit. for this data type and max limit for smaller data type}}{\text{Total number of records}}$$

The following table illustrates attribute selection for sample of records.

Table 1: Attribute selection

Attribute	*Data type*	*Type value %*	*distinct values %*	*Not null %*	*Thre shold %*	*Rank*
Id	Int16	20	100	100	73.33	5
Name	Str.	92	68	97	85.67	2
Add.	Str.	90	80	100	90.00	1
B. D.	Date	100	38	82	73.33	6
Phone	Str.	89	56	89	78.00	4
Comp	Str.	87	28	97	70.67	7
Postal	Str.	50	100	100	83.33	3
Coun.	Str.	40	20	97	52.33	11
Pho.2	Str.	84	36	84	68.00	8
Email	Str.	50	52	88	63.33	9
Web	Str.	39	36	85	53.33	10

B. **Formation of tokens**

This step attempts to remove typographical errors and abbreviations in data fields. This increases the probability that potentially matching records be brought closer after sorting, which uses keys extracted directly from the data fields. Steps have to be taken for the best token key. These steps are: Removing unimportant tokens (Appendix A contains a Reference table for the Unimportant characters used in this work); Expanding abbreviations using (Reference table which is illustrated in appendix B and contains a sample of abbreviations); Formation of Tokens; and Maintaining a LOG table [15],[13]. Figure 2 illustrates the token algorithm.

```
Input: A Table with dirty data after selected attributes.
Output: LOG token table.
Begin:
For each attribute i
    For each row j
        Remove unimportant characters such as special characters.
        Expand abbreviations using Reference table.
        If  row (j) is numeric then
                - Convert string into number
                - Sort number
                - Put into LOG table
            ElseIf  row (j) is alphabet then
                - Select first character of every word
                - Sort these characters in alphabetic order
                -Combine them to obtain the token
                - Put into LOG table
                    ElseIf row (j) is alphanumeric then
                    - Split alphanumeric to numeric and alphabetic
                    - Combine numeric together and alphabetic together
                     - Sort the components
                     -Put numeric first then put alphabetic to formulate
                     token
                     - Put into LOG table
        End If
    Next
```

Table 2 shows token key for the address attribute.

Table 2: Token key for address

Address	*Address Token*
39 Port Saied Street	39PSS
Marquee Retreat	MR
115 Ahmed Maher St.	115AMS

C. **Selection of clustering algorithm**

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters) which shares some common features [16].

Clustering method is also known as Blocking method which refers to the procedure of subdividing data into a set of blocks. There are several clustering methods such as Standard Blocking, Sorted Neighborhood method, Bigram Indexing, and Canopy Clustering [17].

Sorted Neighborhood Method (SNM) is used in this work. It can be summarized in three phases as follows [18]:

1- Creating keys: A key is computed for each record by extracting relevant attribute or portions of attribute which form an important discriminating attribute. The key selection process is a highly knowledge-intensive and domain-specific process, which should know the characteristics of the data.

2- Sorting Data: Sort the records in the data list to find similar records using the key of step 1.

3- Merge: Move a fixed size window W>1 through the sequential list of records limiting the comparisons for matching records to those records in the window. Table 3 illustrates sample of records after clustering using window size W=5.

Table 3: Clustered records

ID	*Cluster key*
361	27DS
360	28DS
3050	A10BS
3051	A10BS
2620	A1MS
2621	A1MS

D**. Similarity computation for selected attributes**

Data cleaning based on similarities involves identification of close tuples, where closeness is evaluated by variety of similarity functions. A variety of string similarity functions are considered, such as edit distance, jaccard similarity, cosine similarity and generalized edit distance [19].

The edit distance algorithm is used to compute similarity. Given two strings s1[1..m] and s2[1..n] over an alphabet Σ, the edit distance between s1and s2 is the minimum number of edit operations needed to convert s1 to s2 [20]. Most common edit operations are the following [21]:

1. Change: Replace one character of s1 by another single character of s2;
2. Deletion: Delete one character from s1;
3. Insertion: Insert one character into s2.

Let D (i, j), $0 \le i \le m$ and $0 \le j \le n$, be the edit distance between s1[1..i] and s2[1..j]. Initially, D (i, 0) = i for $0 \le i \le m$ and D (0, j) = j for $0 \le j \le n$. An entry D (i, j), $1 \le i \le m$ and $1 \le j \le n$, of the D-table is determined by the three entries D(i−1,j −1), D(i −1, j), and D (i, j −1). The recurrence for the D-table is as follows: for all $1 \le i \le m$ and $1 \le j \le n$ [22].

$$D(i,j) = \min \begin{cases} D(i-1,j)+1 \\ D(i,j-1)+1 \\ D(i-1,j-1)+(1 \text{ if } s1(i) \neq s2(j)) \end{cases}$$

The edit similarity ES (s1,s2) is considered as [19]:

$$ES(s1,s2) = 1 - \frac{ED(s1,s2)}{Max(s1,s2)}$$

As: ED(s1,s2) is last value for D(i,j).

Tale 4 illustrates the ES values for the sample of records.

Table 4: ES Values

Id	*Cluster key*	*ES*
361	27DS	0.75,0.2,0.2,0.25
360	28DS	0.2,0.2,0.25,0.25
3050	A10BS	1,0.6,0.6,0.4
3051	A10BS	0.6,0.6,0.4,0.4
2620	A1MS	1,0.5,0.5,0.33
2621	A1MS	0.5,0.5,0.33,0.33

E. **Selection of elimination function**

The elimination process is important to produce cleaned data. In the duplicate elimination step, one copy of the duplicated records is retained and the rest records are eliminated. Several rule-based approaches are proposed for the duplicate elimination process. The commonly available rule-based approaches are the 'Bayes decision rule' for minimum error, Fuzzy rule, Decision with a Reject Region 'Equational theory' and so on [10].

Given two records, r1 and r2, the rule of duplicate elimination can be presented as:

High similarity (r1.cluster key, r2.cluster key) ∧ ((r1.id) ≠ (r2.id)) —> duplicate

Similarity threshold value for the rule based technique as 0.8 to get best results. Because, higher threshold value (0.9) can effect on number of declared duplicates caught by the rule based and this will increase the false positive. And lower threshold will decrease the caught declared duplicates and that will effect on the true positive and thus precision and recall. Threshold value of 0.78 is used in other related work to get good results for both recall and precision [23].

F. **Merge**

In the merge step, records are merged as a cluster, and then appended to the above cluster to form a clean data table. There are different merging strategies used in collecting records as a single cluster. This step is useful for the incremental data cleaning. Incremental data cleaning deals with checking new data with the already created file when entering a new data into the data warehouse. This helps to reduce the time needed for data cleaning [13]. The user must maintain the merged record and the primary representative as a separate file in the data warehouse. This information helps the user to make further changes to the duplicate elimination process. Merged records are loaded into the data warehouse for the decision support process.

3. System Implementation

The proposed data cleaning system contains three components: users interface, data cleaning, and reports component. They can

communicate and cooperate with each other as shown in figure 3.

.

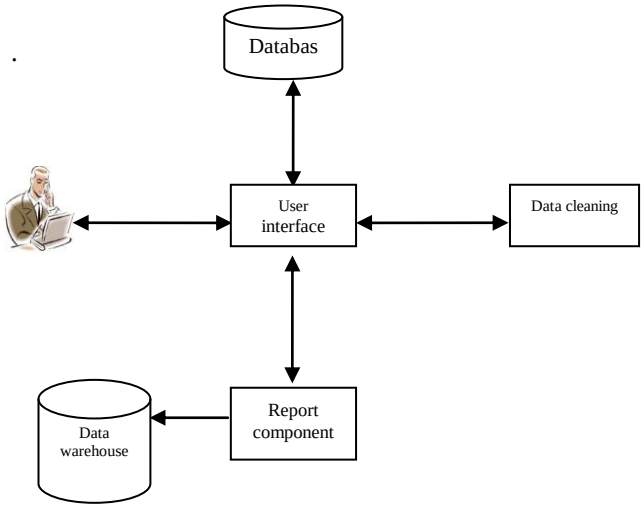


Figure 3: System Components

The system is carried out on a PC with the following specifications: an Intel Core i5 of 2.35 GHz and 4GB RAM, MS Windows 7 (32-bit) operating system, and using the technologies of SQL Server 2010 and Microsoft visual c# 2010. Figure 4 illustrates sample of the proposed system procedures, where the pseudocode for cluster table procedure is presented.

```
Assign dt to Datatable
columnsBeforeToken = dt.Columns.Count -1
for i = 0 to dt.ColumnNames.Count -1 do
        Add(ColumnNames[i]) to dt.Columns
end for
ClusterKeyDT = dt
Assign getStartColId = dt.Columns[ColumnNames[0]].Ordinal
for   i = 0 to dt.Rows.Count -1  do
      for  colId = 1 to ColumnNames.Count do
          if (colId == 1) then
             - getStartColId = dt.Columns[ColumnNames[colId –
                                                    1]].Ordinal
            - if (getStartColId <= columnsBeforeToken) then
            - getStartColId = columnsBeforeToken + 1
              else
                getStartColId = columnsBeforeToken + getStartColId
             end if
             else
                   getStartColId ++
            end if
           Assign s to dt.Rows[i][ColumnNames[colId-1]]
            if(ColumnNames.Count >1) then
                  dt.Rows[i][ getStartColId ] = s
          end for
          dt.Rows[i][getStartColId] = s
  end for
```

Figure 4: Cluster procedure pseudocode

Following figures represent sample of the proposed system screens, where figure 5 illustrates sample of the loaded data screen, figure 6 shows cluster key window, and figure 7 represents the reported cleaned data as an Excel file.

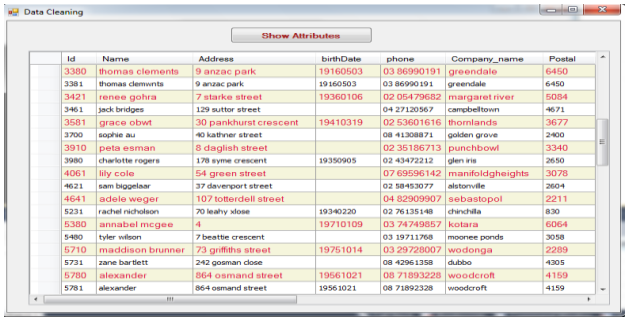


Figure 5: Loaded data

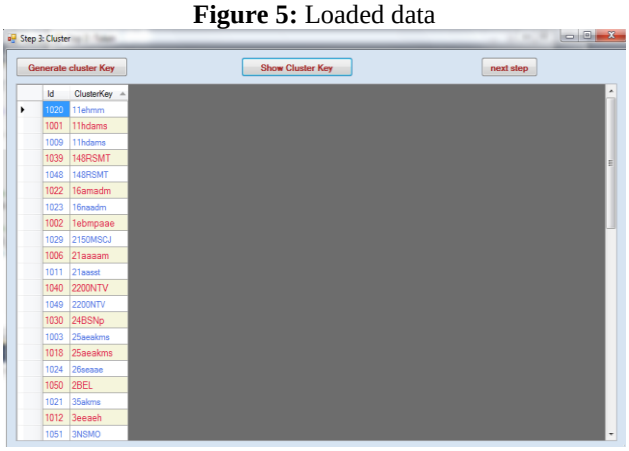


Figure 6: Cluster key window

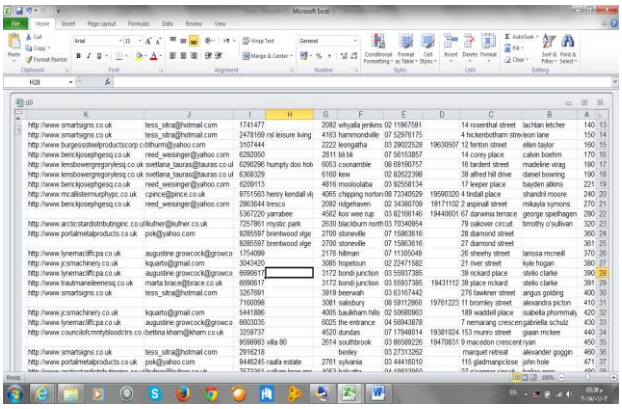


Figure 7: Cleaned data

4. Experimental work

In this section, the proposed system is evaluated and its performance is checked. A dataset is used for the demonstration, FEBRL (Freely Extensible Biomedical Record Linkage), which contains patients' data including 11 attributes: ID, Name, Address, Birth Date, Phone, Company_name, Postal Code, Phone2, Email and Web. Figure 8 illustrates the threshold values for each attribute.

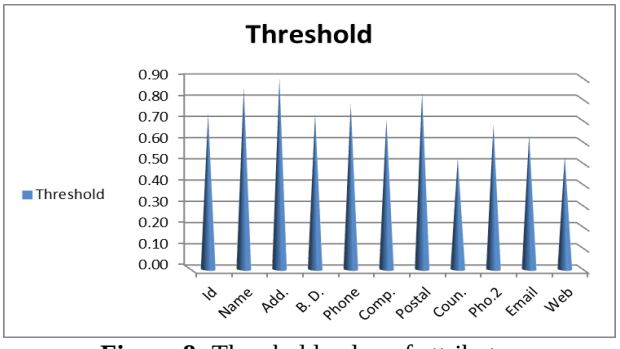


Figure 8: Threshold value of attributes

Precision and recall are the basic measures used in evaluating duplicate detection [23]. They have been used to evaluate the performance of the proposed system. Recall measures the ratio of correctly identified duplicates compared to all true duplicates. It is estimated using the formula:

$$\text{Recall} = \frac{\text{Number of TP}}{\text{Number of TP+Number of FN}}$$

Where: TP (True Positive) means the pairs correctly declared to be duplicate, FN (False Negative) means the candidate pairs not declared to be duplicates while they are actually duplicate. Precision is also called a positive predictor value that measures the ratio of correctly identified duplicates compared to all declared duplicates. It is estimated using the formula:

$$\text{Precision} = \frac{\text{Number of TP}}{\text{Number of TP} + \text{Number of FP}}$$

Where FP (False Positive) means the candidate pairs that are declared to be duplicate may not be duplicates.

Figure 9 presents the effect of various window sizes from 5 records per window to 50 records per window on both recall and precision.

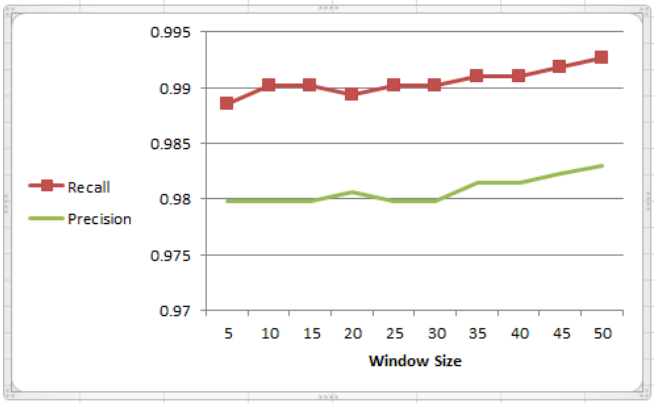


Figure 9: Relation between window size with recall and precision

Figure 10 illustrates the accuracy for duplicate detection techniques according to the dataset size.

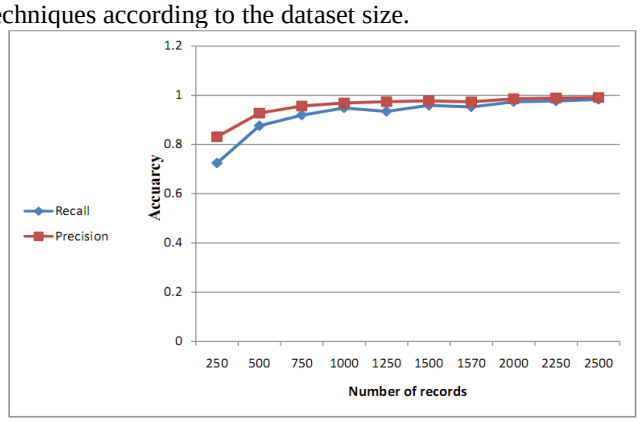


Figure 10: Relation between dataset size with recall and precision

With the dataset contains 2500 records, 1286 of which are unique and 1214 are duplicates. The pervious experiments prove the accuracy of the proposed data cleaning system.

5. Conclusion

The nature of increasing data from time to time, required to clean data to guarantee their quality and facilitate using in the decision support process. A proposed data cleaning system is implemented to meet the needs of users, its framework is consists of six steps working in a sequential order. First, an attribute selection is used to select the best and most suitable attributes depending on the attribute selection criteria. Second, formation of tokens through a token algorithm. In the next step, the SNM clustering algorithm is used to group the records based on a key, then similarity computing is calculated based on the similarity functions. Then, duplicate elimination is done by using the rule-based approach to detect and eliminate low quality duplicates. Finally, cleaned data is merged as a cluster. Experimental results prove the accuracy of the proposed data cleaning system. The system has several advantages such as easy use through interactive interface for the users, it can be is used in different information systems, and it is flexible for all kinds of data.

References

[1] Erhard Rahm and Hong Hai Do,"Data cleaning: problems and current approaches", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2000.

[2] Israr Ahmed and Abdul Aziz," Dynamic Approach for Data Scrubbing Process", International Journal on Computer Science and Engineering Vol. 02, No. 02, pp 416-423, 2010.

[3] Jason D. Van Hulse, TaghiM. Khoshgoftaar, Haiying Huang, "The pairwise attribute noise detection algorithm", Knowl Inf Syst, Volume 11, Issue 2, pp 171-190, 2007.

[4] Enrico Fagiuoli, Sara Omerino and Fabio Stella, "Mathematical Methods for Knowledge Discovery and Data Mining ", Chapter XII, IGI Global, 2008. (URL:http://www.igi-global.com/chapter/bayesian-belief-networks-data-cle-aning/26141)

[5] Judice, Lie Yongkoh," Correlation-Based Methods for Biological Data Cleaning", DOCTOR OF PHILOSOPHY, National University of Singapore, 2007.

[6] Galhardas H., Florescu D., Shasha D., and E. Simon, "An extensible framework for data Cleaning", In Proceedings of 18th international conference on data engineering, IEEE Computer Society, San Jose, 2000.

[7] Rohit Anantha krishna, Surajit Chaudhuri and Venkatesh Ganti, "Eliminating Fuzzy Duplicates in Data Warehouses", Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002.

[8] Mikhail Bilenko and Raymond J. Mooney, "Adaptive Duplicate Detection Using Learnable String Similarity Measures", Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington DC,pp.39-48,August 2003.

[9] Rand Siran Gu," Data Cleaning Framework: an Extensible Approach to Data Cleaning ", degree of Master of Science in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 2010.

[10] J. Jebamalar Tamilselvi and V. Saravanan," A Unified Framework and Sequential Data Cleaning Approach for a Data Warehouse", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.5, May 2008.

[11] D.Lavanya and K.Usha Rani, "Analysis of Feature Selection with Classification: Breast Cancer Datasets", Indian Journal of Computer Science and Engineering, Vol. 2 No. 5, Oct-Nov 2011.

[12] Lifang Gu, Rohan Baxter, Deanne Vickers and Chris Rainsford,"Record Linkage: Current Practice and Future Directions", CSIRO Mathematical and information science, CMIS Technical report, Australia. [Online].

Available: http://dc-pubs.dbs.uni-eipzig.de/files/Gu2003 Record link ageCurrentpracticeandfuturedirections.pdf
[13] j. Jebamalar Tamilselvi, "Detection and Elimination of Duplicate Data Using Token-Based Method for a Data Warehouse: A Clustering Based Approach", PHD thesis, Karunya University, 2009.
[14] Jebamalar Tamilselvi J. and Saravanan V., "Token-based method of blocking records for large data warehouse", Advances in Information Mining, ISSN: 0975–3265, Volume 2, pp-05-10, 2010.
[15] T.E. Ohanekwu, C.I. Ezeife, "A token-based data cleaning technique for data warehouse systems", IEEE Workshop on Data Quality in Cooperative Information Systems, Siena, Italy, January 2003.
[16] K. M. Bataineh, M. Naji, M. Saqer, " A Comparison Study between Various Fuzzy Clustering Algorithms", Jordan Journal of Mechanical and Industrial Engineering, Volume 5, Number 4, Aug. 2011.
[17] Rohan Baxter, Peter Christen and Tim Churches, "A Comparison of Fast Blocking Methods for Record Linkage", CMIS Technical Report,2003.
[18] Wai Lup Low, Mong Li Lee and Tok Wang Ling, " A knowledge-based approach for duplicate elimination in data cleaning ", Information Systems 26, pages 585–606, 2001.
[19] S. Chaudhuri, V. Ganti, and R. Kauskik," A Primitive Operator for Similarity Joins in Data Cleaning", ICDE '06 Proceedings of the 22nd International Conference on Data Engineering, 2006.
[20] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, "Robust and efficient fuzzy match for online data cleaning. In Proceedings of the ACM SIGMOD, June 2003.
[21] Muniba Shaikh, Humaira Dar, Asadullah Shaikh and Asadullah Shah, "Adjusted Edit Distance Algorithm for Alias Detection", International Conference on Information and Knowledge Management (ICIKM), IPCSIT vol.45, IACSIT Press, Singapore, 2012.
[22] Sung-Ryul kim, Kunsoo Park, "A dynamic edit distance table", Journal of descrete algorithms, Vol.2 Issue 2, June 2004, pp.303-312.
[23] Osama Helme Akel, "A comparative study of duplicate record detection techniques", Master Thesis, Middle East University, Amman, Jordan, 2012.

Appendix (a)

Common unimportant tokens

a. **Special characters** are
` , ‘ ” < > - % + _ ( ) . * - $ # ³ º » ¡ ¢ ¤ ¦ © ª « ®! [ ] ^ \ @ : ; ♦ ¯ ← ↑ → ↓ ™ = ? | { } **% & ‘ + - ~ = ? @ µ ¶ Æ Ç È É @** and so on.
b. **Title or Salutation** tokens are
Herr,Monsieur,Hr,Frau,Admiraal,Admiral,Baron,Brig,Brother,Canon,Capt,Captain,Cardinal,Cdr,Cik,Col,Colonel,Count,Mr,Mrs, Ms,Miss,Dr,Chief,Dean,Doctor,Dra,Drs,Father,General,Jonkheer, Judge,Justice,Kolonel,Lady,Lic,Madame,Major,Master,Miss, me, Prof,Prof Dr, Professor, The Hon Dr,The Hon Justice,The Hon Miss,The Hon Mr,The Hon Mrs,The Hon Ms,The Hon Sir, Sir,Sister,Sqn Ldr,Sr,Sr D and so on.
c. **Ordinal forms** are
st,nd,rd,th,ad,ado,and,an,a,din,dor,id,idol,in,ion,dial,do,lion,lir,lord,loan,no,land,nod,road,rand,radio,rin,old,ran,al,in,or and so on
d. **Common abbreviations** are ‘Pvt’, ‘Ltd’, ‘Co’, ‘Rd’, ‘St’, ‘Ave’, ‘Blk’, ‘Apt’, ‘Univ’, ‘Sch’, ‘Corp’ and etc
e. **Common words** are
by,she,or,as,what,go,their,can,who,get,if,would,her,all,my,make,about,know,will,as,up,one,time,there,the,be,and,of,a,in,to,have,to,it,that,for,you,he,with,on,do,say,this,they,at,but,we,his and etc

Appendix (b)

Sample of abbreviations

Shortcut	**Full form**
Bbrev	Abbreviation (of)
Argt.	Argument
Arith	Arithmetic
Arrangem.	Arrangement
art.	Article
Bk.	Book
BNC	British National Corpus
Bord.	Border
cent.	Century
.	.
.	.
Cent.	Central
Chr.	Christian
Dict.	Dictionary