Proof of Duplication Detection in Data by Applying Similarity Strategies

Varsha Wandhekar
Student
DYPSOET, Lohegaon
Pune, India

Arti Mohanpurkar
HOD, Department of Computer Engg.
DYPSOET, Lohegaon
Pune, India

ABSTRACT
Deduplication is the process of determining all categories of information within a data set that signify the same real life / world entity. The data gathered from various resources may have data high quality issues in it. The concept to identify duplicates by using windowing and blocking strategy. The objective is to achieve better precision, good efficiency and also to reduce the false positive rate all are in accordance with the estimated similarities of records. Various Similarity metrics are commonly used to recognize the similar field entries. So the main focus of this paper is to applying appropriate similarity measure on appropriate data to properly identifying the duplicates. De-duplication is a property which provides additional information of similarities between the two entities. Thus, in today's data centric environment there are huge numbers of defects in similarity measure. As a result to identify the duplicates is always been a challenging task. In this paper the primary focus is given on exact identification of duplicates in the database by applying concept of windowing & blocking. The objective is to achieve better precision, good efficiency and also to reduce the false positive rate all are in accordance with the estimated similarities of records.

Keywords
Deduplication; Similarity Measure; Sorted Neighborhood Method(SNM); Windowing; Blocking.

1. INTRODUCTION
Data cleaning may be the important dependence on the particular businesses whoever data will be growing rapidly in addition to they want to retain pace using the growing technology in order to meet the particular emerging requirements of their users. Offered data have to be clear making sure that good choices might be considered for the data in addition to customer satisfaction can also become increased. Data cleaning is essential with respect to organization viewpoint. When data aren't appropriate, complete, as well as reliable subsequently decisions considered on such basis as data can be bad or might be inaccurate. As soon as standalone resources tend to be integrated, the data quality difficulties are by mistake increased. The main difficulty with dirty data may be the existence of duplicates. The removal of duplicates is definitely a critical cleaning issue which is the particular concentrate point in this particular research study. [1]

A lot of industries along with techniques are determined by the particular accuracy of databases when planning on taking their own operations, proper along with competitive initiatives. The quality of the data located within the particular databases, can offer major expense significances to a system that depends on information to function along with perform organization. Minimal data quality leads to improper reporting, inability to create a comprehensive view of the consumers from several sectors and results in inadequate customer satisfaction along with costs, quantities of dollars for you to businesses within postage, printing, along with personnel overhead. That's why; data quality enhancement is definitely an ongoing exercising and essential stage before starting data warehouse. [2]

Duplicates are generally a numerous of representations in the same real-world entity as well as item. De-duplication can be a significant method within data integration as well as data cleansing. IT detects records of which signify the same entities as well as merges them in to a single record. De-duplication will become any non-trivial task happens because duplicates usually are not exactly similar, frequently on account of ambiguity within the data. Thus, utilization of feasible complex matching strategy to evaluate all object representation, to determine as well as discover if they are same real life entity or not. Rather, we can't discover the duplicates by simply common comparison algorithm. Because of its highly realistic importance within data integration as well as data cleansing scenarios, de-duplication has been studied extensively for relational data located in the table. However, the detection of duplicates commonly performed by comparing pairs of tuples by simply computing similarity score depending on their particular attribute values. In case similarity of two tuples are previously mentioned by predefined threshold then that two tuples are categorized as duplicates [6].

This typographical fluctuation of string data is probably the most frequent source of mismatch in database. Several "Similarity Strategy's" have been described to estimate the similarity of a pair of data entities. Therefore, de-duplication generally depends on string analysis techniques to handle typographical fluctuations [7].

A number of data mining tasks entail computing similarity in between two pairs of records. The total number of pairwise similarity computations increases steadily along with the size of the input dataset, scaling to huge datasets is challenging process. For small datasets, estimation of complete similarity matrix might be complicated. Essentially the most illustration pairs are very distinct therefore in several process task similarity computation are unneeded [8]. There are various solutions to determining the de-duplication, but this papers primarily focuses on the two techniques. The first is Windowing along with another one Blocking [9].

The structure of the paper is raised as takes after. Section II provide concept of Windowing and Sorted Neighborhood Method along with issues. Section IV is concerned with the diverse Similarity Strategy. The proposed framework working

is clarifying in Section V. Section VI depicts Mathematical Modeling. An experimental result gives in Section VII. Section VIII is the conclusion.

2. BLOCKING

One particular method for detecting similar records in the database would be to traverse the particular table and also analyze the value of a hash function for every record. The value of the hash function describes the particular "blocks" in order to which often that record has allotted. By means of definition, two records that are very same will be assigned for the same bucket. Therefore, to find out duplicates, it is adequate to equate simply the records that fall under the same blocks for matches. The hashing strategy is not used specifically for approximate duplicates since there is no assurance that the hash value of two same records is definitely the same. On the other hand, there's an interesting comparative of this approach, known as blocking [4].

Blocking techniques dividing the record tuples set in to disjoint dividers as well as blocks. After that compare all pairs associated with record tuples merely within specific block. So the entire number of comparisons is getting decreased. During the past years variety of blocking algorithms are already proposed by researchers [10], [11], [12], [13], [14]. These techniques usually form blocks or groups of observations by using sorting as well as indexing. For subsequent similarity computations this allows useful selection of instance pairs of each and every block. A number of blocking techniques depend on the similarity metric.

3. SORTED NEIGHBORHOOD METHODE AND WINDOWING

The key representative intended for windowing is usually Sorted Neighborhood Method (SNM). They have three phases:

1) Key selection: Sorting key is usually allocate to each and every record. The key is actually made by concatenating 2 or more values of attributes.

2) Sorting: All records are generally sorted based on key.

3) Windowing: Window get slides over sorted data. Within certain window all records pairs are compared and duplicates are generally marked [4].

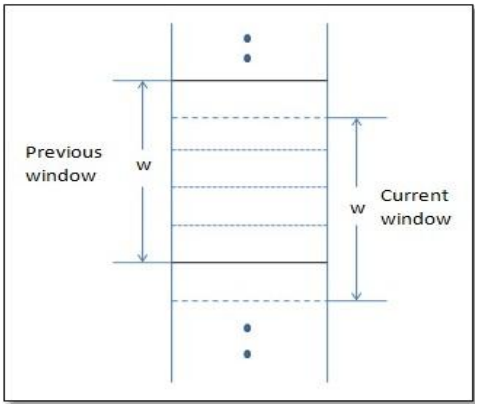

Fig 1: The representation of Window in Sorted Neighborhood Method

Fig 1. indicates the representation of window in SNM. The single search within of SNM over complete 'n' number of records with 'w' number of records per window makes n − w + 1 blocks. Considering that each and every block acquires w − 1 quantity of record comparisons. Because of the relatively faster running time when compared to the model approach as well as much easier implementation, the particular SNM technique has became a regular selection of duplication detection algorithm in several data applications [5].

A drowback of the sorted neighborhood method is the fixed window size. A number of duplicates might be skipped when choosed window size is too small. In contrast, unwanted comparison performed when window size too large. To accomplish effectiveness adaptive window size is used [3], [5], [9], [11].

To make de-duplication solution sutaible, look at that adaptivety performs essential role. Therefore paper concentrate on adaptively and dynamically changing parameters of de-duplication during execution. To maintain efficiency as well as effectiveness, we compare the Adaptive SNM (ASNM) and SNM.

4. SIMILARITY MEASURES

One of the most common assets of mismatches within information source records is the typographical alterations of string information. Consequently, duplicate recognition usually depends on string analysis strategies to deal with typographical alterations. Various strategies have been intended for these techniques, and each and every approach useful for specific kinds of problems. While mistakes might appear in number areas as well, the related exploration is still in its early stages. In this region, we describe strategies that have been applied for appropriate regions along with string information in the duplicate record identification perspective [4].

There are two types of record matching; one is lexical heterogeneity and another is structural heterogeneity. The databases having similar structure but distinct representation of data are lexical heterogeneity, such as 'Varsha Wandhekar', 'V. Wandhekar' and 'Varsha W'. The issue of matching two databases with distinct area structures is structural heterogeneity. For e.g. a customer education stored in the attribute 'references' in one database but represented in attributes 'author', 'year', and 'publication' in another database [8] . Three types of similarity strategies as follows: Character-based similarity, Token based similarity and Phonetic similarity. But proposed system mainly focuses on the character-based similarity.

4.1 Character-Based Similarity Measure

The issue associated with incorrect matches in databases is because of the typographical dissimilarities regarding entered data. The procedure detection depends upon approximate string matching strategies to manage these kinds of issues. Character-based similarity strategy handles typographical mistakes for strings..

4.1.1 Edit Distance Measure:

The edit distance between two strings 1 and string 2 is the minimal number of edit operations of single characters likely to change the string 1 into string 2.

There are three edit operations:

- insert a character into the string,
- remove a character from the string, and
- replace one character with a different character.

In the easiest type, each one modify operation has cost 1. The edit distance measurements work well for capturing typographical mistakes, but they are generally worthless for other kinds of mismatches [4]. The Levenshtein distance is new version of edit distance. So similarity calculated by using:

$$Simedit = 1-(LevenshteinDist/Max(s1, s2))$$

4.1.2 Jaro Distance Measure:

Jaro was mainly string comparison algorithm introduced for comparing the first and last names. For comparing the two strings 1 and string 2 some basic algorithmic steps required to calculate:

- Calculate the lengths of string 1 and 2;
- Find the "common characters" c in the two strings;
- Find the number of transpositions t;

$$SimJaro = (1/2)((m/s1) + (m/s2) + (m-t/m))$$

The number of transpositions is calculated as follows:

We compare the i^{th} common character in string1 with the i^{th} common character in string2. Each non-similar character is a transposition. [16]

4.1.3 Jaro-Wrinkler Distance Measure:

The Jaro-Wrinkler is extention of the Jaro distance metric. Let p be the length of the common prefix of string 1 and 2.

$$Sim_{Wrinkler} = Sim_{Jaro} + (1- Sim_{Jaro})((c-p+1)/(s_1+s_2-p(p-1)))$$

Table 1. Comparison of String Comparators

Two Strings		**String Comparator Values**		
		Edit Distance	***Jaro***	***Wrinkler***
Neeta	Nitaa	0.199	0.783	0.826
Kiran	Karan	0.6	0.919	0.832
Madhura	Madhuri	0.714	0.904	0.961
Ajay	Vijay	0.4	0.672	0.672
Snehal	Sneha	0.833	0.944	0.972
Ram	Shyam	0.199	0	0
Swaraj63	Svarajya3	0.555	0.805	0.729
Div004	Deep04	0.333	0.666	0.572
Varsha	Varsha	1	1	1

Table 1. compares the values of the Edit-Distance, Jaro, and Winkler values for some first names and last names. Edit Distance are normalized to be between 0 and 1. All string comparators take value 1 when the strings similar as character-by-character.

5. PROPOSED SYSTEM

De-duplication is depending on the similarity strategy as well as windowing and blocking algorithm. For maintain efficiency the proposed system uses the adaptive windowing technique.

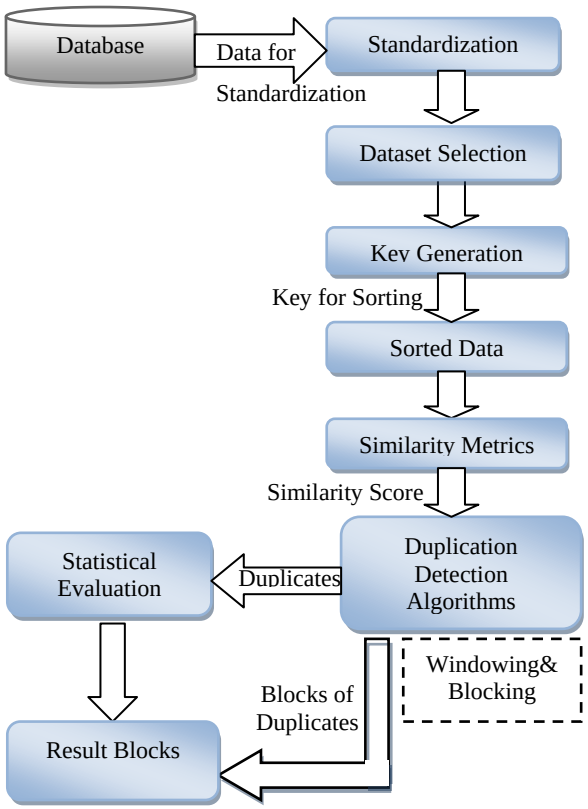

Fig 2: The Flow diagram of Proposed System

Fig. 2 indicates the flow of de-duplication system. This system works on various datasets which are storing in the databases. System divides in different steps as follows:

5.1 Standardization

De-duplication system, standardization converts the data in specific or unique standardize format as names or addresses into components that can be easily differentiate. In proposed system standardization perform on the first name attribute. In this which name contains 'oo' that will converted into 'u', etc. eg: 'Pooja' gets converted to 'puja'. 'Neeta' gets converted to 'Nita'. [18]

5.2 Key Generation

Key Generation has most important and necessary state in de-duplication. Key selection has done as per categories of dataset [17].

Table 2. Key Generation

First Name	**Last Name**	**Address**	**Phone No.**
Arti	Mohanpurkar	Pune	9421234567

So key made by concatenation of some attributes:

- 3 letters from First_Name,
- 3 letters from Last_Name,
- 3 letters from PhoneNo.

eg: KEY: artmoh567

Duplication detection algorithms in this stage algorithms are compared and this are based on windowing and blocking techniques. After that result of the each algorithm compares.

- *Proposed System Algorithm :*
- *Input:* Record Dataset, Key, Threshold(Φ)
- *Steps:*
    1. Sort the data using key
    2. Initialize Window size(w)
    3. Comparison is on Window
        a. Similarity Measure(dist)
        b. Comparing With Threshold(Φ)
        c. Enlargement or Retrenchment
    4. Block of duplicates(b)
- *Output:* Blocks of Duplicates, Values of F-score, Precision, Recall.

6. MATHEMATICAL MODEL

A. Set Theory

Consider S is the set of the system

- $S = \{K,D, A, M, R|\rightarrow s_\Phi\}$

1. K is set of Number of Keys.

$$K = \{k_0\ k_1, k_2,\ldots\ldots\ldots\ k_n\ |\rightarrow k_\Phi\ \}$$

2. D is the set of Number of Dataset for source data.

$$D = \{d_0, d_1, d_2\ldots\ldots\ldots\ d_n\ |\rightarrow d_\Phi\}$$

3. A is the Set of Duplication Detection algorithm.

$$A = \{a_0, a_1, a_2,\ldots\ldots\ldots\ldots\ a_n\ |\rightarrow a_\Phi\}$$

4. M is set of Similarity Measures.

$$M = \{m_0, m_1, m_2\ldots\ldots\ldots\ m_n\ |\rightarrow m_\Phi\}$$

5. R is Result Set

$$R = \{r_0, r_1, r_2\ldots\ldots\ldots\ldots\ r_n\ |\rightarrow r_\Phi\}$$

B. Relevant Mathematics

The proposed system mainly based on the blocking and windowing. So considering the following equations:

6.1.1 Windowing:

$$Ws = \Phi * Wc / dist(W1,Wn) \qquad (1)$$

Where:

Ws = Final Window Size

Φ = Distance Threshold

Wc = Current Window Size

W1 = First record in Window

Wn = Last record in Window

dist() = Distance according to Similarity Measure

6.1.2 Duplicate Blocks:

$$b = N/Ws \qquad (2)$$

Where:

b = Number of Duplicate Blocks

N = Total Number of Tuples in Dataset

7. RESULT ANALYSIS

This system uses two real data sets and one artificial data set, all of which have been used in the existing work on blocking, to test the performance of adaptive sorted neighborhood methods. Controlled experiments have also been done to evaluate the adaptive methods from different aspects. Table 3 shows the summary of datasets.

Table 3. Smmary Of Datasets

Database Name	Size	Property	Field	Content
Cora	1295	Real	12	Citation
Restaurant	894	Real	4	Restaurant address
Mytable	1000	Artificial	14	Name and Address

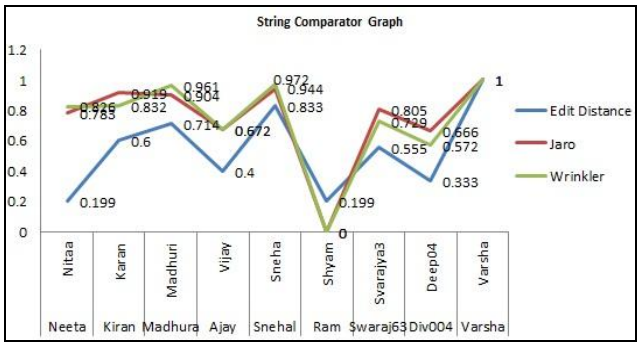

Fig 3: String Comparator Graph

According to table I. string comparator graph has drawn. Fig. 3 shows the string comparator graph. In this graph Jaro provide better similarity than the Edit distance and Wrinkler when string is the combination of character and number. Otherwise, Wrinkle provides the better similarity than the both Jaro and Edit distance.

The de-duplication system is mainly concerned to the threshold value of similarity measure. In the proposed system, we designed the Dedup algorithm [5],[18],[19]. The graph shows comparison between SNM overlapping, SNM and Dedup algorithm.

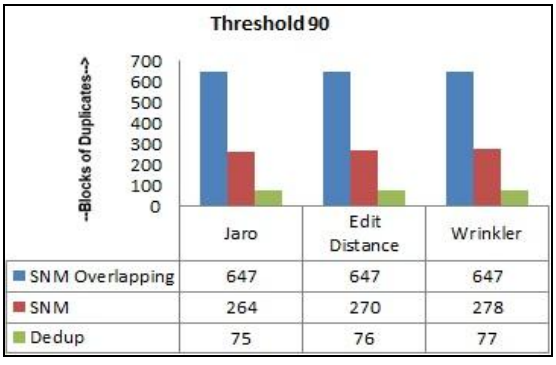

Fig.4: Comparison of blocks of duplicates vs. threshold 90 for Cora Dataset

Fig. 4 shows the graph of cora dataset. This graph shows comparison of three algorithms done by using three similarity strategy and threshold 90. So the Dedup algorithm has fewer blocks of duplicates than SNM and SNM Overlapping.

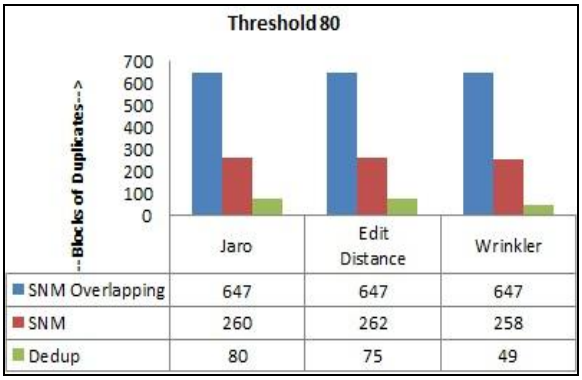

Fig.5: Comparison of blocks of duplicates vs. threshold 80 for Cora Dataset

Fig. 5 shows the graph of cora dataset. This graph shows comparison of three algorithms done by using three similarity strategy and threshold 80. In this also Dedup algorithm has fewer blocks of duplicates than both. If we consider similarity strategy, then Wrinkler has fewer blocks than Jaro and Edit distance.

Fig.6: Comparison of blocks of duplicates vs. threshold 70 for Cora Dataset

Fig. 6 shows the graph of cora dataset. This graph shows comparison of three algorithms done by using three similarity strategy and threshold 70. In this also Dedup algorithm has fewer blocks of duplicates than both and Wrinkler has fewer blocks than Jaro and Edit distance.

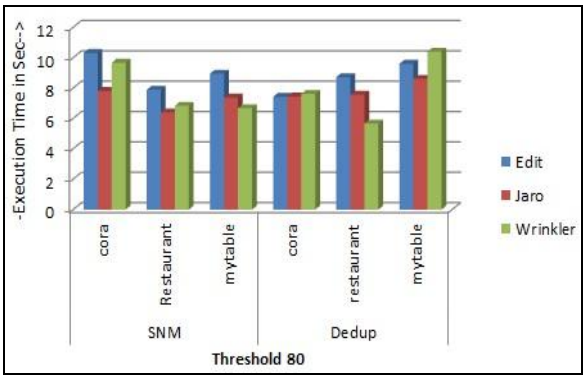

Fig.7 Execution time of system when threshold 80

Fig 7: Shows the execution time of system when threshold 80. For cora dataset Dedup algorithm requires less execution time than the SNM for all similarity measures, when threshold value is 80. According to Restaurant Dataset wrinkler required less execution time in SNM and Dedup. But according to Mytable dataset Dedup algorithm required more execution time than the SNM in all Similarity Strategy.

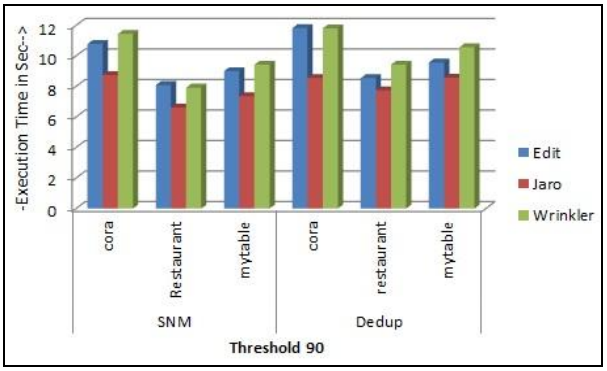

Fig.8 Execution time of system when threshold 90

The fig 8: shows the time of execution when threshold value is 90. In both cases the system using jaro is slightly faster than using edit distance and wrinkler.

8. CONCLUSION

This paper describes the comparison of existing algorithm with proposed 'Dedup' algorithm by applying three similarity strategies. By experimentally dedup algorithm provides fewer blocks of duplicates than the existing algorithm. Because of the adaptive nature of algorithm, It provide efficiency. It removes the drawback of SNM.

The selection of appropriate similarity strategy for appropriate dataset is important for finding duplicates. This paper also discusses & justify that Jaro gives efficient results than Edit distance and Wrinkler.

9. ACKNOWLEDGMENTS

The authors would like to thank Shri Chairman Groups and Management and the Director/Principal Dr.Uttam Kalawane, Colleague of the Department of Computer Engineering and Colleagues of the varies Department the D.Y.Patil School of Engineering and Technology, Pune Dist. Pune Maharashtra, India, for their support, suggestions and encouragement.

10. REFERENCES

[1] M.Rehman, V.Esichaikul, "Duplicate Record Detection For Database Cleansing", Second International Conference on Machine Vision, 2009.

[2] E. Rahm and H. Hai Do, "Data Cleaning: Problems and Current Approaches", IEEE Computer Society Technical Committee on Data Engineering, 2000, pp:3-13.

[3] L. Gu and R. Baxter, "Adaptive filtering for efficient record linkage," in Proceedings of the SIAM International Conference on Data Mining, 2004, pp. 477–481

[4] A. Elmagarmid, P. Ipeirotis, and V. Verykios, "Duplicate record detection: A survey", IEEE Transactions on Knowledge and Data Engineering (TKDE), 2007, pp:1-16.

[5] S. Yan, D. Lee, M. Kan, C. Lee Giles, "Adaptive Sorted Neighborhood Methods for Efficient Record Linkage", ACM,JCDL, June 2007, pp:17-22.

[6] L. Leitao, P. Calado, and M. Herschel, "Efficient and Effective Duplicate Detection in Hierarchical Data", IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 5, May 2013

[7] N. Koudas, S. Sarawagi, D. Srivastava,"Record Linkage: Similarity Measures and Algorithms", ACM, SIGMOD 2006, pp:802-804.

[8] V. Wandhekar, A. Mohanpurkar, "A Review on Efficient and Effective Duplicate Detection in Data", International Journal for Research in Applied Science and Engineering Technology (IJRASET), ISSN: 2321-9653, Volume 2 Issue XI, November 2014,pp: 103-107.

[9] U. Draisbach, F. Naumann, "A Generalization of Blocking and Windowing Algorithms for Duplicate Detection", IEEE, 2011, pp: 18-24.

[10] M.Bilenko, B.Kamath, R.Mooney, "Adaptive Blocking: Learning to Scale Up Record Linkage", In Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM-06), Hong Kong, December 2006, pp. 87-96.

[11] U. Draisbach, F. Naumann, S. Szott, and O. Wonneberg, "Adaptive windows for duplicate detection," ACM SIGKDD interntional conference on Knowledge discover and data mining, NY, USA, 2011

[12] K.Prasad, S. Chaturvedi, T. Faruquie, L. Subramaniam, "Automated Selection of Blocking Columns for Record Linkage", IEEE,2012.

[13] J. Nin, V. Mulero, N.Bazan, Josep-L. L.Pey, "On the Use of Semantic Blocking Techniques for Data Cleansing and Integration",11th International Database Engineering and Applications Symposium, 2007.

[14] U. Draisbach and F. Naumann, "A comparison and generalization of blocking and windowing algorithms for duplicate detection," in Proceedings of the International Workshop on Quality in Databases (QDB),2009.

[15] R. Baxter and P Christen. , "A comparison of fast blocking methods for record linkage," In In ACM SIGKDD workshop on Data Cleansing, Record Linkage and Object Consolidation, pages 25-27, Washington DC, 2003.

[16] D.Bharambe, S.Jain, A.Jain, "A Survey: Detection of Duplicate Record", International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 11, November 2012.

[17] W. Winkler, "Overview of Record Linkage and Current Research Directions", Statistical Research Report, February 8, 2006

[18] V. Raisinghani, S. Sarawagi, " Cleaning Methods in Data Warehouse", School of Information Technology, IIT Bombay, 1999.