Validation of Deduplication in Data using Similarity Measure

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.

Keywords

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

1. INTRODUCTION

Databases play an important role in day-to-day IT-based environment. Many industries and systems depend on the databases to carry out operations. Data Warehouse of an enterprise combines the data from multiple sources of the enterprise or organization to support enterprise wide reporting, planning, analyzing and decision making. They depend on the consistency and accuracy of data. Cleansing of data detects and determines and corrects the corrupt, unwanted, faulty, inconsistent data to enhance the data quality. The large volume of data stored in storage suffers from the problems of dirty data. Because of unreliable and inconsistent data from multiple sources enters in data warehouse the quality of enterprise data degrades due to storing of large volume of data in data warehouse [1], [2].

The Deduplication has been known as various names in different disciplines– record linkage [3], duplicate detection [4], entity resolution, citation matching, identity uncertainty, merge-purge, object matching [5].

Duplicates are several representations of the same real-world entity or object. Deduplication is an important process in data integration and data cleaning. It finds records that represent the same entities and merges them into a single record. Deduplication becomes a nontrivial task is because duplicates are not exactly equal, often due to unambiguity in the data. Therefore, use of possible complex matching strategy to compare all object representation, to decide or find if they are same real world entity or not. Instead, we cannot find out the exact duplicates by common comparison algorithm. Due to its highly practical importance in data integration and data cleaning situations, Deduplication has been studied widely for relational data placed in a single table. In this case, the detection of duplicates typically done by comparing pairs of tuples by computing a similarity score based on their attribute values. If similarity of two tuples are above a predefined threshold then that two tuples are classified as duplicates [6].

The typographical variations of string data is one of the most common source of mismatch in database. Various “Similarity Measures” have been defined to calculate the closeness of a pair of data entities. Therefore, deduplication typically relies on string assessment techniques to deal with typographical variations [7].

A number of data mining tasks involve computing similarity between pairs of records. The total number of pairwise similarity computations grows gradually with the size of the input dataset, scaling to large datasets is problematic task. For small datasets, estimation of the full similarity matrix can be difficult. The most instance pairs are highly dissimilar so in many task majority of similarity computation are unnecessary [8]. There are various methods to finding out the deduplication, but this paper mainly focuses on the two methods. One is Windowing and another one Blocking [9].

The structure of paper is erected as follows. Section II explains the concept of Blocking. Windowing and Sorted Neighborhood Method describe in Section III along with problems. Section IV is concerned with the different Similarity Measures. The proposed system working is explain in Section V. Section VI describes the Relevant Mathematics. Analysis of results provides in Section VII. Section VIII is the concluding section.

2. BLOCKING

One method for detecting identical records in a database table is to traverse the table and calculate the value of a hash function for each record. The value of the hash function defines the “blocks” to which this record is allotted. By definition, two records that are same will be assigned to the same bucket. Therefore, in order to discover duplicates, it is sufficient to equate only the records that fall into the same block for matches. The hashing technique cannot be used directly for approximate duplicates since there is no guarantee that the hash value of two similar records will be the same. However, there is an interesting equivalent of this method, named blocking [4].

Blocking methods partitioning the record tuples set into disjoint partitions or blocks. Then compare all pairs of record tuples only within particular block. So the overall number of comparisons is getting reduced. In the past years numbers of blocking algorithms have been proposed by researchers [10], [11], [12], [13], [14]. These techniques typically form blocks or groups of observations using sorting or indexing. For subsequent similarity computations this allows efficient selection of instance pairs from each block. Some blocking methods are based on the similarity metric.

3. SORTED NEIGHBORHOOD METHODE AND WINDOWING

The most important representative for windowing is Sorted Neighborhood Method (SNM). It has three phases:

1) Key selection: Sorting key is assigned to each record. The key is generated by concatenating two or more values of attributes.

2) Sorting: All records are sorted according to key.

3) Windowing: Slides a window over sorted data. Within particular window all records pairs are compared and duplicates are marked [4].

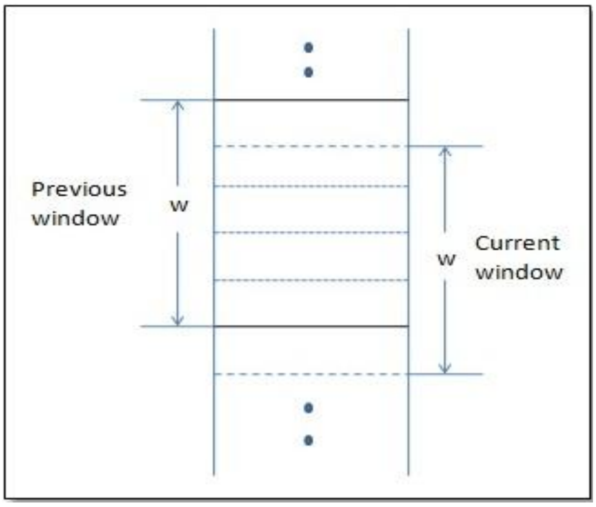


Fig 1: The representation of Window in Sorted Neighborhood Method

Fig 1. Shows the representation of Window in SNM. The Single scan of SNM over total 'n' number of records with 'w' number of records per window yields n − w + 1 blocks. Since every block acquires w − 1 number of record comparisons. Due to the comparatively faster running time compared to the model approach and the easier implementation, the SNM approach has been a standard choice of duplication detection algorithm in many data applications [5].

A disadvantage of the Sorted Neighborhood Method is the fixed window size. Some duplicates might be missed when selected window size is too small. On the other hand, unnecessary comparison carried out when window size too large. To achieve effectiveness adaptive window size is used [3], [5], [9], [11].

In order to make duplication detection solution applicable, consider that adaptively plays important role. So in this paper we focuses on adaptively and dynamically changing parameters of duplication detection during execution. To maintain effectiveness and efficiency we compare the Incrementally Adaptive SNM (IA-SNM) and Accumulatively adaptive SNM (AA-SNM) algorithms [5], [14].

4. SIMILARITY MEASURES

One of the most common resources of mismatches in information source records is the typographical modifications of sequence information. Therefore, copy recognition generally depends on sequence evaluation techniques to deal with typographical modifications. Several techniques have been designed for this process, and each method works well for particular types of mistakes. While mistakes might appear in number areas as well, the related exploration is still in its early stages. In this area, we explain methods that have been applied for relevant areas with sequence information in the duplicate record recognition perspective [4].

There are two types of record matching; the first is lexical heterogeneity and the second is structural heterogeneity. The databases with similar structure but different representation of data are Lexical heterogeneity, such as 'V. Wandhekar', 'Varsha W.'and 'Wandhekar, Varsha'. The problem of matching two databases with different domain structures is Structural heterogeneity. For e.g. a customer education stored in the attribute 'education' in one database but represented in attributes 'class', 'degree', and 'branch' in another database [8] . Three types of Similarity Measures as follows:

4.1 Character-Based Similarity Measure

The problem of wrong matches in databases is due to the typographical dissimilarities of entered data. The process of duplicate detection depends on approximate string matching techniques to handle such problems. Character-based similarity metrics deal with typographical errors for strings.

4.1.1 Edit Distance Measure:

The edit distance between two strings 1 and 2 is the minimum number of edit operations of individual characters expected to change the string 1 into 2.

There are three edit operations:

- insert a character into the string,
- remove a character from the string, and
- replace one character with an alternate 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].

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 2 some basic algorithmic steps required to compute:

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

The number of transpositions is calculated as follows: We compare the ith common character in 1 with the ith common character in 2.Each non-similar character is a transposition. all previous will be in two columns [16].

The Jaro-Wrinkler is extention of the Jaro distance metric [17].

Table 1. Comparison of String Comparators

Two Strings		**String Comparator Values**		
		Edit Distance	***Jaro***	***Wrinkler***
SHACKLEF	SHACKELF	0.818	0.970	0.982
JONES	JOHNSON	0.667	0.790	0.832
MASSEY	MASSIE	0.667	0.889	0.933
ABROMS	ABRAMS	0.833	0.889	0.922
ITMAN	SMITH	0.000	0.000	0.000
JON	JAN	0.667	0.000	0.000
VARSHA	VARSHA	1.000	1.000	1.000

Table I. 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.

4.2 Token-Based Similarity Measure

Typographical conventions sometimes cause rearrangement of words e.g. ("Varsha Wandhekar" versus "Wandhekar, Varsha"). In such cases, similarity between strings not well when we use character based similarity measure. So, to overcome the problem of character-based similarity measure were introduced Token-based similarity measure. Token based similarity measure focus on string-based representation of the data. On the other hand, records consist of various fields [7].

4.3 Phonetic Similarity Measure

Some strings may be phonetically similar even they are not similar in a character or token wise. For example, the word 'Krypton' is phonetically similar to 'Cripton' even if the fact that the string representations are very dissimilar. The phonetic similarity measures are trying to discover such problems and match such strings. The Soundex code for a name based on the way a name sounds. Some rules:

1. Keep the first letter of the name and omit all other occurrences of a, e, i, o, u, y, h, w.
2. Allocate consonants with digits as follows:

b, f, p, v	1
c, g, j, k, q, s, x, z	2
d, t	3
L	4
m,n	5
R	6

3. Two contiguous letters with the same number are coded as a single number.
4. Continue while you have one letter and three numbers. If you have few letters, append with 0s.

eg.: "Tymczak" is encoded as "T522".

5. PROPOSED SYSTEM

Validation of duplicate detection is based on the similarity measure as well as windowing and blocking algorithm. The proposed system uses the adaptive windowing algorithm for maintain the effectiveness.

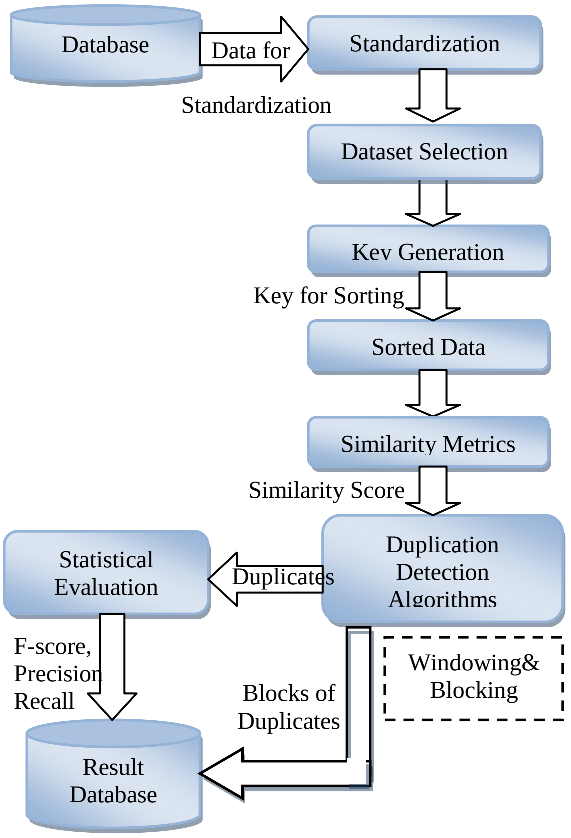


Fig 2: The Flow diagram of Proposed System

Fig. 2: shows the flow of proposed system, In this system all or various datasets are storing in the databases. This system divide in different steps as follows:

5.1 Standardization

In duplicate detection, Standardization converts the data in particular or specific standardize format as names or addresses into components that can be more easily compared. It also refers to methods for putting dates such as 15 January 2015 or Jan. 15, 2015 into a standardized MMDDYYYY format of '01152015', in address field 'Apt.' is consider as a 'Apartment', 'MH' as actually change into 'Maharashtra'[18]

5.2 Key Generation

Key Generation is every important and necessary task in detection of duplication. Key is selected as per categories of dataset [17].

Table 2. Key Generation

First Name	**Last Name**	**Address**	**Phone No.**
Varsha	Wandhekar	pune	9421234567

So Key is Concatenation of Some Fields:

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

eg: KEY: varwan567

Duplication detection algorithms in this step various algorithms are compared which are based on blocking and windowing methods. Then compare the result of each algorithm.

- *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

6.1 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

6.2 Evaluation Metrics

Every dataset was randomly split into 2 folds for crossvalidation for each experimental run. A larger number of folds is impractical since it would result in few duplicate records per fold. To create the folds, duplicate records were grouped together, and the resulting clusters were randomly assigned to the folds, which resulted in uneven folds in some of the trials. All results are reported over 20 random splits, where for each split the two folds were used alternately for training and testing.

At every cycle, the pair of records with the most astounding similarity was named a duplicate, and the transitive closure of groups of duplicates was modified. Precision, recall and Fmeasure characterized over pairs of duplicates were processed after every cycle, where precision is the part of distinguished duplicate matches that are appropriate, recall is the division of actual duplicate pairs that were recognized, and F-measure is the harmonic mean of recall and precision:

1) Precision=ofCorrectlyIdentifiedDuplicatePairs / ofIdentifiedDuplicatePairs
2) Recall = 1- (ofCorrectlyIdentifiedDuplicatePairs / ofTrueDuplicatePairs)
3) F -Measure =(2 *Precision * Recall) / (Precision + Recall)

7. RESULT ANALYSIS

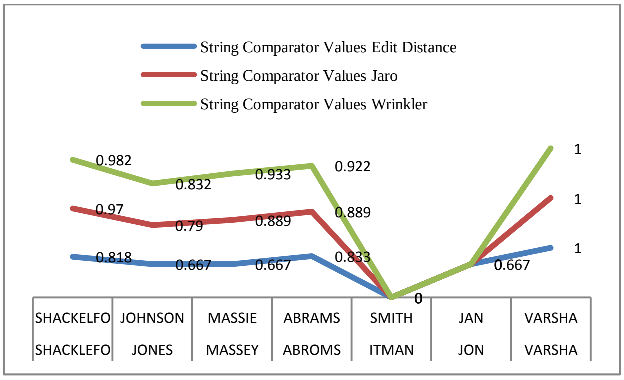
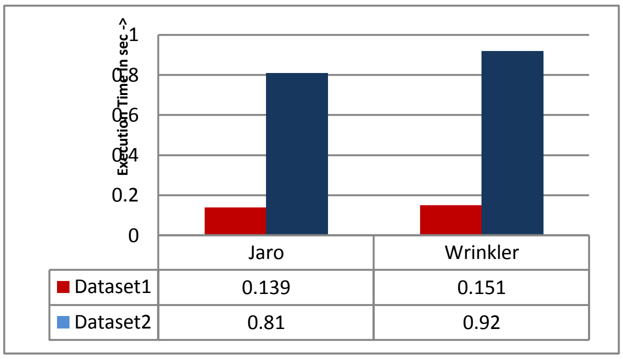


Fig 3: String Comparator Graph

Our experiments were conducted on two datasets. Dataset1 is a database of 500 names and addresses. Dataset2 is a collection of 100 person's names. Fig.3. Shows the string comparator Values between Edit distance, Jaro and Wrinkler Measures. This comparison is done on some strings of above datasets.

The deduplication system mainly concern on the Threshold value of similarity measure. In proposed system we designed the Incremental Adaptive SNM(IASNM)[5],[9],[14]. Jaro and Wrinkler are used as a similarity measure for this comparison analysis. So when the threshold value is 0.85 the execution time of system shown in Fig.4. The Jaro required less time than wrinkle. The Fig.5 and Fig.6. shows the time of execution when threshold values are 0.75 and 0.50 respectively. In both cases the system using Wrinkler is slightly faster than using Jaro.



	Jaro	Wrinkler
🟥 Dataset1	0.139	0.151
🟦 Dataset2	0.81	0.92

Fig 4: Threshold value of System is 0.85

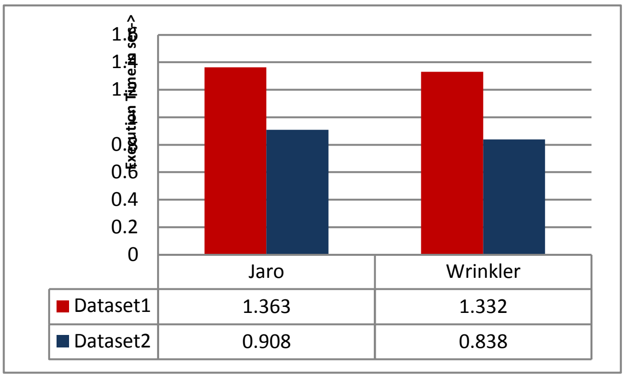


Fig 5: Threshold value of System is 0.75

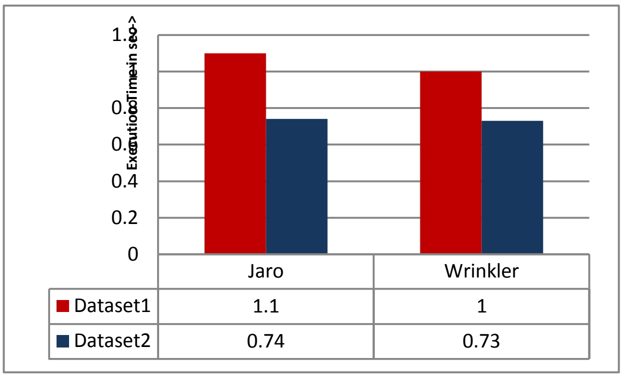


Fig 6 : Threshold value of System is 0.55

8. CONCLUSION

This paper described the deduplication detections using various similarity measures as well as windowing and blocking techniques. The three types of character based similarity measure comparison is also provided in this paper. The selection of similarity measure is also described, so anyone can select appropriate similarity measure for appropriate dataset. In this paper Wrinkler Provide better similarity rather than Edit distance and Jaro.

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 Know ledge 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.