CLUSTERING APPROACH IN CONTEXT FREE DATA CLEANING

Sohil D. Pandya, Dr. Paresh V. Virparia

ABSTRACT

In this era of Knowledge, organizations can gain competitive advantage only by proficient data analysis. This paper emphasizes on application of clustering in context free data cleaning by correcting values of attributes, using various sequence similarity metrics, where reference data set is not available, to improve the quality of data which in turn lead to eminent data analysis. Authors propose an algorithm to examine suitability of value to correct other values of attributes. Various sequence similarity metrics were used, to find distance of two values of attributes, to test the data and generate results. Experimental results show how the approach can effectively clean the data without reference data.

Keywords: Clustering, Context free data cleaning, Sequence similarity metrics.

1. INTRODUCTION

In this era of Knowledge, information is the most crucial factor to gain competitive advantage for the majority of organizations. Using various Information Systems, organizations are trying to manage vast pool of data entering and exiting of them and making their business decisions. But even the best implemented Information Systems failed to provide required information and lack to generate, preserve, and disseminate organizational knowledge [3]. Hence organizations are now interested to have Knowledge Management Systems (KMS) to fulfill the needs of stakeholders, which has a major process called Knowledge Discovery in Databases (KDD). KDD aims at discovering *interesting* and *useful* patterns from large databases. In general, KDD comprises three major phases:

1. Data preprocessing: consists of cleaning of imperfect or noisy data, integration of multiple data sources, selection of relevant data and transformation.
2. Data mining: consists of application of various techniques of data mining like association mining, classification, clustering, outlier analysis, etc.
3. Pattern evaluation & presentation: includes identification and evaluation of *interesting* patterns and presenting them in *useful* formats.

In the above phases, data preprocessing is the crucial phase, which cannot be ignored, because outcome of KMS depends on how clean your data is. The presence of imperfect or noisy data can significantly distort an analysis of the data, which also could lead to poor results.

In most of the information systems, before the data is stored in databases it passes through various steps including human interventions & computations, where data can become noisy or incorrect due to typographic errors, manipulation errors, improper interfaces, incomplete data etc. Hence, it is required to preprocess data for cleaning, to improve the quality of data, before starting any analysis. The data quality measures (allow quantifying data in order to achieve high performance results of various analyses) like completeness, valid, consistent, timeliness, accurate, relevant etc. Hence, varieties of research have been carried out over the last decade on various aspects of data cleaning to improve quality of data.

The purpose of this paper is to find out one of the possible applications of clustering in data cleaning to correct values of attributes. The next section of the paper discusses how and why clustering is helpful in the process and later section describes an algorithm, its experimental results and concluding remarks.

2. APPLICATION OF CLUSTERING IN DATA CLEANING

Generally data cleaning processes are tedious and require major interventions of human being [2]. Human being can validate various values of variety of attributes for their correctness and provide input for amendments based on their knowledge, experience, and mainly based on reference and/or correct data sets. Reference and/or correct data sets, (like list of countries, states, city, car models, product catalogue of a company, etc.) are generally created by domain expert, have a great utility in the field of cleaning & attribute standardization and act as validation and/or transformation rules while cleaning. But in real world reference and/or correct data sets for various attributes are not available at initial stage due to pervasive nature of values of attributes. Data mining is applications of various techniques to search previously unknown knowledge and it could be used to discover reference data set directly from data [1, 2]. Clustering is the assignment of a set of observations into subset so that observations in the same clusters are similar in some sense, which has various applications in machine learning, data mining, pattern recognition, bioinformatics, etc [5]. To apply clustering, one of the techniques of data mining, approach in context free cleaning to correct values of attributes is the major focus of the paper. Context free cleaning means, to examine value of attributes without taking into account further values of attributes. Here, the core idea is, based on the frequency of values, in descending order, they are matched with other values, and based on *matching* between them, it is decided that whether they should be transformed or not? The algorithm described in next section examines suitability of values to become member of reference dataset and transforms other *matching* values to them.

3. CONTEXT FREE DATA CLEANING

The proposed algorithm has two major components: clustering and nearest neighborness. It has an important parameter *acceptableDist*, which is the minimum acceptable distance required during matching and transforming (ranges from 0.0 to 1.0, where 0.0 is not similar sequence and 1.0 is same sequence).

To measure the distance we used following sequence similarity metrics:

1. Needleman-Wunch Algorithm
2. Jaro Winkler Distance
3. Jaccard Similarity Index
4. Euclidean Distance
5. Cosine Similarity
6. Chapman Ordered Name Similarity
7. Dice Similarity
8. Block Distance
9. Modified Levensthien Distance
10. Smith-Waterman Algorithm

And for further process and validation we put emphasis on some of the above metrics based on initial results and their methodologies, which are discussed below:

The Needleman-Wunch algorithm, as in (1) performs a global alignment on two sequences and commonly used in Bioinformatics to align protein sequences [6].

$$\begin{aligned} F_{0j} &= d * j \\ F_{i0} &= d * i \\ F_{ij} &= \max(F_{i-1,\,j-1} + S(S_{1i}, S_{2j}), F_{i,\,j-1} + d, F_{i-1,\,j} + d \end{aligned} \qquad (1)$$

Where $S(S_{1i},S_{2j})$ is the similarity of characters i and j; d is gap penalty.

The Jaro-Winkler distance, as in (2), is the major of similarity between two strings [6]. It is a variant of Jaro distance [6].

$$\begin{aligned} Jaro\text{–}Winkler(S_1, S_2) &= Jaro(S_1, S_2) + (L * p(1 - Jaro(S_1, S_2)) \\ Jaro(S_1, S_2) &= \frac{1}{3}\left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m}\right) \end{aligned} \qquad (2)$$

Where m is number of matching characters and t is number of transpositions required; L is length of common prefix and p is scaling factor (standard value 0.1).

Chapman Ordered Name Similarity tests similarity upon the most similar terms of token-based name where later name are valued higher than earlier names [6].

The Smith-Waterman algorithm, as in (3) is well-known algorithm for performing local sequence alignment, i.e. for determining similar regions

between two protein sequences. It compares segments of all possible lengths and optimizes the similarity measures using substitution matrix and gap scoring scheme [6].

$$H(i,0) = 0, 0 \le i \le m \qquad (3)$$

$$H(0, j) = 0, 0 \le j \le n$$

$$H(i, j) = \max\begin{Bmatrix} 0 \\ H(i-1, j-1) + w(S_{1i}, S_{2j}), Mismatch \\ H(i-1, j) + w(S_{1i}, -), Deletion \\ H(i, j-1) + w(-, S_{2j}), Insertion \end{Bmatrix}$$

Where S_1, S_2 are strings and m, n are their lengths; H (i, j) is the maximum similarity between strings of S_1 of length i and S_2 of length j; w(c,d) represents gap scoring scheme.

The algorithm consists of following steps:

1. Sequences for a selected attribute are transformed to uppercase.
2. All non-alpha and non-numeric characters are removed.
3. Derive frequencies in descending order, for all the distinct sequences. Refer the group of distinct values as clusters and the sequences as cluster identifiers.
4. Select any of the sequence similarity metrics for comparing two values of an attribute and decide *acceptableDist.*
5. Compare the cluster identifier with other cluster identifiers, beginning with first to last cluster, to decide distance between them.
6. If the distance is less than acceptableDist then it forms transformation and/or validation rules for particular acceptableDist that can be utilized in further cleaning process (e.g., second pass of the same algorithm, context dependant cleaning) and the values of comparables can be transformed in to comparator, else comparables remains as separate clusters.

4. EXPERIMENTAL RESULTS

The algorithm is tested using a sample data derived from Internet. The data consisting of attributes named *First Name, Middle Name, Last Name, Address, City, Pin code, District, State, Country, Phone number*, and *Email*. *District* attribute is selected the testing purpose. There were about 13,074 records out of which 551 (4.22 %) values for the selected attribute were identified as incorrect and required corrections. During the execution of algorithm, 359 clusters were identified for the selected attribute. After identification of clusters and their identifiers, algorithm is tested for various similarity metrics value. For selected similarity metrics various results, like how many records updated (total,

correctly & incorrectly), were found and are discussed in this section. Here is an example given for one similarity metrics, i.e., for Needlemen-Wunch algorithm with 0.9 similarity metrics value, we found transformation rules as shown in Table I (the table is showing data of one cluster identifier 'PANCHMAHAL' only, there are other 101 cluster identifiers which were transformed into other cluster identifiers resulting into total 221 alteration for above specification).

Table 1. EXAMPLE FOR RESULT

Original Value	Frequency (*f*)	Corrected Value
PANCHAMAHAL	108	PANCHMAHAL
PANCHMAHALS	8	PANCHMAHAL
PAMCHMAHAL	1	PANCHMAHAL
LANCHMAHAL	1	PANCHMAHAL
PANCHMAHALA	1	PANCHMAHAL
PUNCHMAHAL	1	PANCHMAHAL
PANACHMAHAL	1	PANCHMAHAL
OANCHMAHAL	1	PANCHMAHAL

Following results, percentage of correctly altered (CA %), percentage of incorrectly altered (IA %) and percentages of unaltered values (UA %) were derived as in (4).

$$CA(\%) = \frac{CA}{TotalAlteration} * 100$$
$$IA(\%) = \frac{IA}{TotalAlteration} * 100 \qquad (4)$$
$$UA(\%) = \frac{UA}{NumberofIncorrectValues} * 100$$

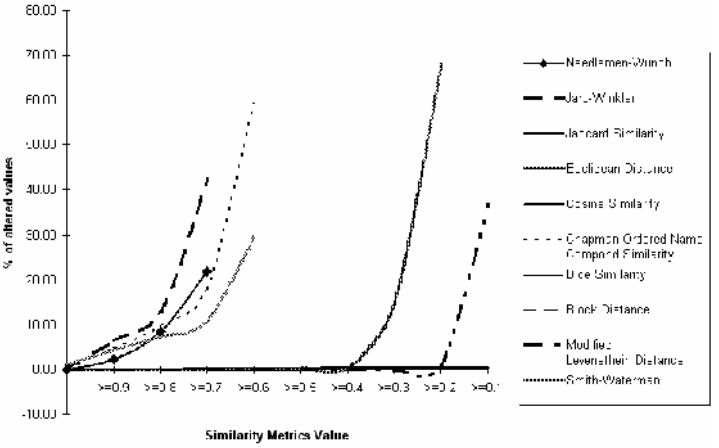

Fig. 1. Percentage alteration using various similarity metrics.

Results found on testing of algorithms are:

1. It can be observed in Fig. 1 that for various sequence similarity metrics that the percentage of values altered is growing with the increase of the *acceptableDist* parameter as the greater tolerance for matching criteria. For instance, using Needlemen-Wunch algorithm with distance values 1, 0.9, and 0.8, 07 there were 0.0%, 2.25%, 8.42%, 21.78% values altered respectively. And using Smith-Waterman algorithm with distance values 1, 0.9, 0.8, 0.7, 0.6 there were 1.22%, 4.46%, 7.39%, 10.79%, 29.39% values altered respectively.

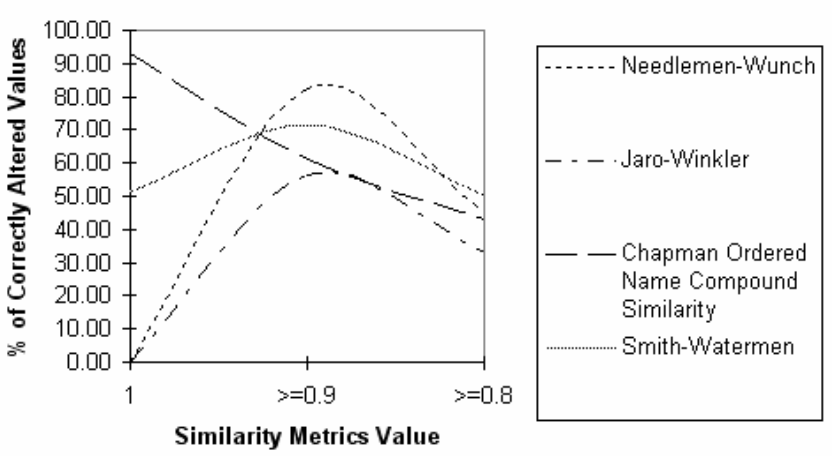

Fig. 2. Percentage of correctly altered values.

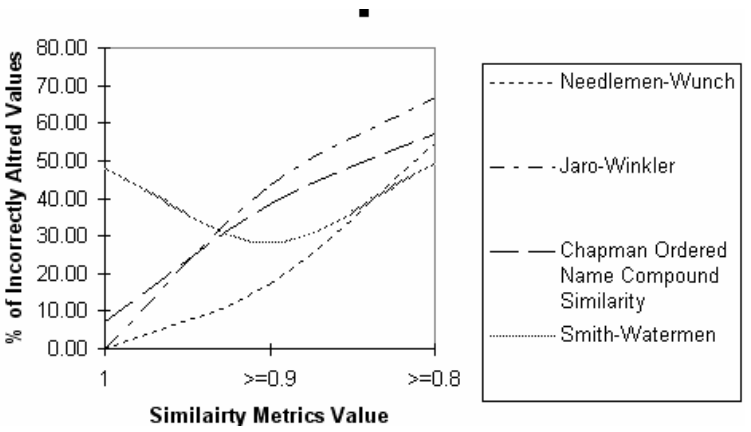

Fig. 3. Percentage of incorrectly altered values.

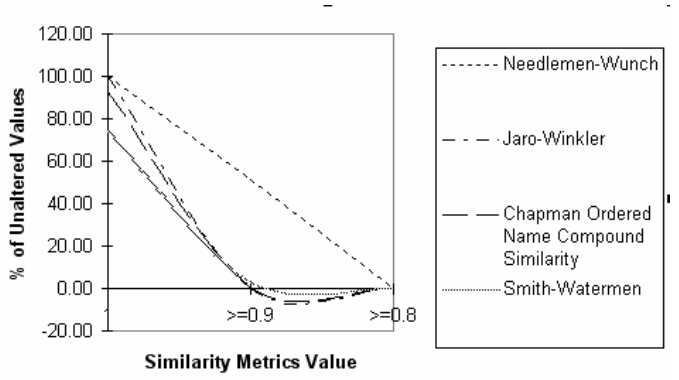

Fig. 4. Percentage of unaltered values.

2. By looking further we found that percentages of correctly altered values are increased and percentages of unaltered values are decreased as acceptableDist for various sequence similarity metrics is increasing but the percentages of incorrectly altered values are also being increased (this percentage can be decreased by more passes of algorithm) as shown in Fig. 2, Fig. 3, and Fig. 4. For instance, using Needlemen-Wunch algorithm with distance values 1, 0.9, 0.8 there were 0.0%, 82.46%, 45.31% values were altered correctly respectively, 0.0%, 17.54%, 54.69% values altered incorrectly respectively, with respected to total altered values and 100%, 51.36%, 0.0% values unaltered with respect to total noisy values
3. The major disadvantage of the algorithm is to incorrectly classify some values (generally in earlier passes) even if they are correct in real world context.

5. CONCLUSION

The results of the experiments verify the correctness of the algorithm and which motivate to use it for data cleaning. The major benefits of it where the reference or correct data set is not available with you or difficult to decide them and still wanted to clean the data. Here the results show that even the reference or correct data set are not provided it is possible to clean the data to certain percentage. However, this percentage is relative to the parameters (like *acceptableDist*) and dataset, i.e. for different dataset would require different parameter value for achieving a high success ratio in cleaning.

In above experiments various sequence similarity metrics were used. It is possible that other metrics or functions and/or various combinations of them, as per the requirements, may give better results and this should be explored in further experiments. And also other data mining techniques can be applied for more precise data cleaning could be thought.

REFERENCES:

1. Hui Xiong, Gaurav Pandey, Michael Steinbach, Vipin Kumar "Enhancing Data Analysis with Noise Removal" in IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 3, pp. 304-319, March 2006.
2. Lukasz Ciszak "Application of Clustering and Association Methods in Data Cleaning", in Proc. of Int. Multiconference on Computer Science and Information Technology, Vol. 3, pp. 97-103, 2008.
3. Sohil D Pandya, Dr. Paresh V Virparia "Data Cleaning in Knowledge Discovery in Databases: Various Approaches", in Proc. of National Seminar on Current Trends in IT (CTICT) – 2009, February 2009.
4. W Cohen, P Ravikumar, S Fienberg "A Comparison of String Distance Metrics for Name-Matching Tasks" in Proc. of the IJCAI-2003
5. http://en.wikipedia.org/
6. http://www. dcs.shef.ac.uk/~sam/simmetric.html