International Journal of Information Science: 2011; 1(1): 32-35
DOI: 10.5923/j.ijis.20110101.05

Context Free Data Cleaning and its Application in Mechanism for Suggestive Data Cleaning

Sohil D. Pandya[1,*], Paresh V. Virparia[2]

[1]MCA Department, Sardar Vallabhbhai Patel Institute of Technology (SVIT), Vasad, 388306, India
[2]G H Patel PG Dept. of Computer Science & Technology, Sardar Patel University, Vallabh Vidyanagar, 388120, India

Abstract Organizations are being flooded with massive transactional data. This data is of no use, if not analysed properly, to reach to any strategic decision and ultimately to achieve competitive advantage. The efficient data analysis is one of the success strategies. The analysis is highly dependent on the quality of the data. The clean data will lead to efficient data analysis. In this paper, authors suggest application of similarity metrics in context free data cleaning and a mechanism to suggest correct data based on learning from patterns derived in the prior phase. The sequence similarity metrics like Needlemen-Wunch, Jaro-Winkler, Chapman Ordered Name Compound Similarity and Smit-Watermen are used to find distance of two values. Experimental results show that how the approach not only effectively cleaning the data but suggesting suitable values in order to reduce the data entry errors.

Keywords Context Free Data Cleaning, Similarity Metrics

1. Introduction

1.1. Data Cleaning

Organizations are being flooded with massive transactional data every year. This data needs to be analysed carefully to achieve competitive advantage in order to achieve excellence. Some organizations have moved to and others are moving to develop business intelligence solutions by analysing this data, identifying patterns among them, and deciding strategies after evaluating the patterns. The efficiency and reliability of this analysis relies heavily on the quality of the data. The data quality measures like completeness, valid, consistent, timeliness, accurate, relevance etc. allow quantifying data in order to achieve high performance results of various data analyses. Because of human interventions and computations at various levels, noise is added in data before it got stored[4]. Noise is "irrelevant or meaningless data"[1], which leads to deterioration of outcome of data[2]. The data cleaning processes mainly focused on detection and removal of noise. Using Similarity Metrics in data cleaning process to identify and replace incorrect sequence with correct sequence based on distance between them is an interesting way of cleaning of data[5]. Here, the distance for various similarity may be based on numbers of characters, number of replacements needed to convert one sequence to another, number of re-arrangements required, most similar characters, any combinations of all above, etc. the distance between two sequence ranges from 0.0 to 1.0. For example,
the distance between "Malaysia" and "Mallayssia" for various similarity metrics is shown in table 1.

The purpose of this paper is to demonstrate application of similarity metrics in context free data cleaning and again use the learned knowledge in further cleaning. Authors developed the algorithm and function – cleanAssit to perform the said job. The later section of the paper describes the algorithm and function & their experimental results.

Table 1. Values of Distance for Various Similarity Metrics.

Similarity Metrics	Distance
Needlemen-Wunch	0.8000
Smith-Waterman	0.8750
Chapman Ordered Name Compound Similarity	0.9375
Jaro-Winkler	0.9533

1.2. Related Work

With the usage of similarity metrics, Lucasz have proposed and implemented algorithm by using Levensthein Distance[2]. He suggested usage of data mining techniques like clustering and classification for context dependent and context free cleaning of data. He later used more factors for optimizing results.

Hui-Zen, et-al. has suggested Data Cleaner for cleaning large databases.[1]

Authors themselves extended the technique suggested by Lucasz (by experimenting various similarity metrics, their permutations, etc.) and make appropriate changes for

* Corresponding author:
sohilpandya@gmail.com (Sohil Pandya)
Published online at http://journal.sapub.org/ijis

improved cleaning. Authors also extended the concept to develop a mechanism that would suggest users the best possible correct sequence based on their response. These best possible correct sequence will be decided by –

(1) The sequence set which was generated by the context free data cleaning.

(2) Users support and confidence to accept/reject the suggestions.

The above said process will deliver effective results as more usage by users and gradually by learning from users. Over a period of time the system become more & more mature enough to generate near to perfect results.

2. Algorithm

2.1. Assumption

1. Typographic errors are less (ranges 5% to 20%) and similar in nature.

2. Results may depend upon dataset is to be cleaned and the distance metric is to be used.

2.2. Important Terms

1. The function M : S x S $\rightarrow$ R2 is a distance metric between the elements of the set S, such that

$$M(s_m, s_n) = (d_{mn}, r_{mn})$$

Where,

$d_{mn} = D_w(v_m, v_n)$, $d_{mn} \in (0,1)$ (D_w is measure of similarity between two text strings)

$$r_{mn} = \max\left(\frac{om}{on}, \frac{on}{om}\right), r_{mn} \in (0, \infty)$$

2. Similarity relation '~' over the set is a relation defined in that way that:

$s_m \sim s_n \Leftrightarrow d_{mn} \geq \varepsilon \wedge r_{mn} \geq \alpha \wedge o_m < o_n$, where is ε threshold distance - acceptableDist and α is occurrence ratio - occurRatio.

3. $p_c = (n_c / n_a) * 100$ (percentage of correctly altered values)

4. $p_i = (n_i / n_a) * 100$ (percentage of incorrectly altered values)

5. $p_0 = (n_{00} / n_0) * 100$ (percentage of values marked during the review as incorrect, but not altered during cleaning)

(where n_c is number of correctly altered values, n_i is number of incorrectly altered values, n_a is total number of altered values, n_{00} is the number of elements marked as incorrect during the review process that were not altered, n_0 is the number of values identified as incorrect)

6. CleanAssist is a function, which suggests reference data generated during context-free cleaning process has following features:

a. For Correct Input – It suggests most similar suggestion with high updated frequency and confidence.

b. For Wrong Input –

i. Historical Mistakes – It suggest the most correct texts based on previous mistakes and also updates if user accepts.

ii. New Mistakes – It suggests the most matches by matching the entered string with previous mistakes and correct values based on distance threshold and also remember new mistakes. Next, if the user has done the same mistake, it will take it as a historical mistake.

2.3. Algorithm – Context Free Data Cleaning

The algorithm has two important components - clustering and similarity and two important parameters *acceptableDist* (which is a minimum acceptable distance required during matching and transformation) and *occurRatio* (as defined in 2.2). To measure the distance we experimented on following Similarity Metrics:

1. Needlemen-Wunch
2. Jaro-winkler

$$F_{0j} = d * j$$

$$F_{i0} = d * i$$

$$F_{ij} = \max(F_{i-1,j-1} + S(S_{1i}, S_{2j}), F_{i,j-1} + d, F_{i-1,j} + d$$

3. Smith-Watermen
4. Chapman Ordered Name Compound Similarity

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

Where S (S1i, S2j) 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].

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

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 Compound 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].

$$H(i,0) = 0, 0 \leq i \leq m$$

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

$$H(i,j) = \max\begin{Bmatrix} 0 \\ H(i-1,j-1) + w(S_{1i}, S_{2j}), Mismatch \\ H(i-1,j) + w(S_{1i}, -), Deletion \\ H(i,j-1) + w(-, S_{2j}), Insertion \end{Bmatrix} \quad (3)$$

Where S1, S2 are strings and m, n are their lengths; H (i, j) is the maximum similarity between strings of S1 of length i and S2 of length j; w (c,d) represents gap scoring scheme.

The algorithm for context free data cleaning 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 metric for comparing two values of an attribute and decide acceptableDist and occurRatio.

(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 more than acceptableDist and occurRatio it forms transformation and/or validation rules for particular acceptableDist and occurRatio, that can be utilized in further cleaning process (i.e. CleanAssist) and the values of comparables can be transformed in to comparator, else comparables remains as separate clusters.

2.4. Function– CleanAssit

After the first and/or sub-sequential run for various data set the function – cleanAssit is incorporated in the system so as to suggest the correct data values. This function has features as discussed in 2.2.

Call these steps when user entered string changes –

(1) For the entered sequence of text, display the matching comparators.

(2) If the results are not found in step-1, then display the matching comparables. Also make necessary updates, if user selects the comparable, in frequency and confidence of comparables and comparators.

(3) If the results are not found in step-2, then display the most matched comparator by matching entered text to comparators and/or comparables based on acceptableDist. Also make necessary inserts with entered sequence and selected comparator.

2.5. Implementation

The implementation of the above algorithm and function is done using NetBeans IDE in Java language with back-end as MySQL.

2.6. Experimental Results & Discussion

The experimental results and discussion is section 3.

2.7. Limitations

1) The algorithm depends on the selection of distance metric. Different distance metrics have different approaches and distant measure. So it depends on application which distance metric is suitable for which type of data.

2) The results may vary when same acceptableDist and occurRatio but with a different Similarity Metric upon same dataset, which may lead to confusion upon selection of Similarity Metric.

3) The algorithm may alter the data which may be correct in real world.

4) The accuracy and efficiency of the cleanAssist function depends on the –

a. The reference set generated during context-free data cleaning.

b. The support and confidence of the user on selecting/rejecting the suggestions.

2.8. Applications

1) Cleaning of data where reference data set is not available or difficult to construct.

2) Natural Language Processing.

3) Genetics to identify matching and/or replacement of sequence.

4) Spell Checkers

5) Understanding the demographic mindsets of data entry operators in injecting data to the system.

2.9. Future Scope

1) This algorithm can be extended in the direction of automatic selection of similarity metric based on the nature of the data.

2) The algorithm may allow little intervention of user before transforming of comparable to comparator (it may be time consuming, but can be allowed as it is not a daily process).

3) The cleanAssist function can be optimized and linked with real world data, if available.

3. Experimental Results & Discussion

To test the above algorithm and function, authors have used data downloaded from Internet. The data has following attributes – student id, name, address, city, district, state, country, phone no, and email address. The experiment was done on district attribute.

The experimental results were tested for the measures p_c, p_i, and p_0 (as discussed in 2.2).

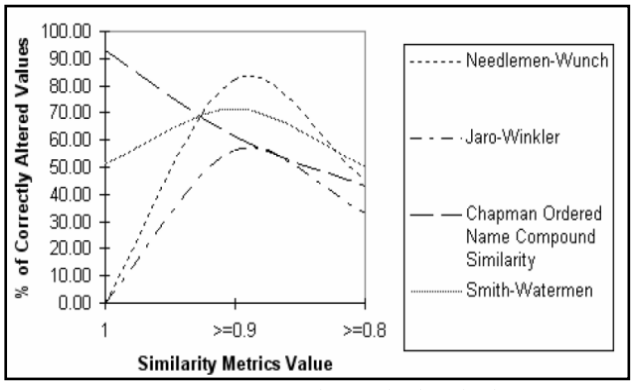

Figure 1. Percentage of correctly altered values.

1) It can be observed in Fig. 1 that various similarity metrics the percentage of value altered is growing with the

increase of *acceptableDist* and *occurRatio*. With various datasets algorithm updates counts of historical mistakes and add new mistakes (See Fig. 2). Here, the count value supports while suggesting the best suitable values. The value having the highest count value leads to be the first in the suggestion array and so on.

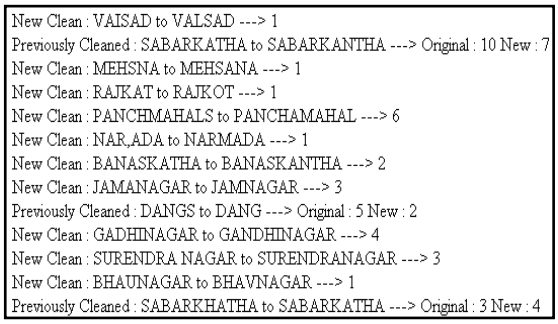

Figure 2. Example of context free data cleaning by an algorithm.

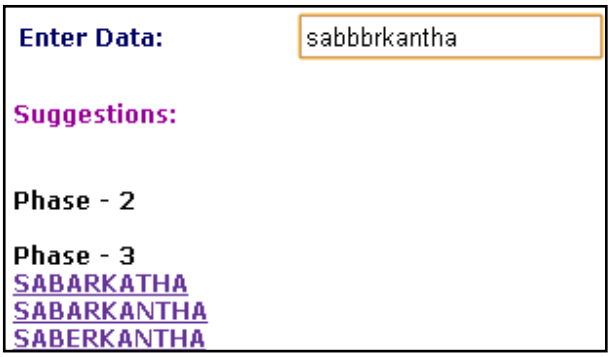

Figure 3. Example of suggestions by cleanAssist.

2) Based on the altered values the cleanAssit keep suggesting the best possible suggestions by matching with historical mistakes and/or to be mistakes and also remembers new mistakes. This new mistakes will be taken in to consideration when new dataset will be taken to context free data cleaning and so on.

3) By keeping track of user's decision for accepting/rejecting suggestions, the cleanAssist make decisions for maintaining numbering of suggestion array, adding/removing suggestions, and consideration in future cleaning.

4) Gradually, the cleanAssit learns for generating best suitable suggestions to users who are making or about to make typographic mistakes (See Fig. 3). And over a period of time injecting false/dirty data tends to minimal.

4. Conclusions

The above said algorithm, its implementation and results were motivating. Still there are future scope lies in the said mechanism (as discussed in 2.8). Authors wish to apply above algorithm in various applications (as discussed in 2.7) and test the applicability. Overall the algorithm and function helps to clean the data where reference data set is not available.

REFERENCES

[1] Hui Xiong, Gaurav Pandey, Michael Steinbach, and Vipin Kumar, Enhancing Data Analysis with Noise Removal, IEEE Transactions on Knowledge and Data Engineering, Vol 18, No. 3, pp. 304-319, 2006

[2] Lukasz Ciszak, Application of Clustering and Association Methods in Data Cleaning. In proceedings of the International Multi-conference on Computer Science and Information Technology, 2008

[3] Sohil Pandya and Dr. P. V. Virparia, Testing Various Similarity Metrics and their Permutations with Clustering Approach in Context Free Data Cleaning, Inter-national Journal on Computer Science & Security, Vol. 3, No 5, pp. 344-350, Nov. 2009

[4] Sohil Pandya and Dr. P. V. Virparia, Data Cleaning in Knowledge Discovery in Databases: Various Approahces, In proceedings of National Seminar on Current Trends in ICT, India, Feb. 2009

[5] W Cohen, P Ravishankar, and S Fienberg, A Comparison of String Distance Metrics for Name Matching Tasks, In the proceedings of IJCAI 2003

[6] http://en.wikipedia.org

[7] http://www.dcs.shef.ac.uk/~sam/simmetric.html