A Unified Framework and Sequential Data Cleaning Approach for a Data Warehouse

J. Jebamalar Tamilselvi[†] **and Dr. V. Saravanan** [††],
PhD Research Scholar Associate Professor & HOD
Department of Computer Application
Karunya University
Coimbatore – 641 114,Tamilnadu, INDIA

Summary
The data cleaning is the process of identifying and removing the errors in the data warehouse. Data cleaning is very important in data mining process. Most of the organizations are in the need of quality data. The quality of the data needs to be improved in the data warehouse before the mining process. The framework available for data cleaning offers the fundamental services for data cleaning such as attribute selection, formation of tokens, selection of clustering algorithm, selection of similarity function, selection of elimination function and merge function. This research paper deals about the new framework for data cleaning. It also presents a solution to handle data cleaning process by using a new framework design in a sequential order.

Key words:
Data cleaning, Data quality, Data warehousing and Data mining.

1. Introduction

Data quality refers an 'error-free' approach in the data warehouse. The quality of data needs to be increased by using the data cleaning techniques. Existing data cleaning techniques used to identify record duplicates, missing values, record and field similarities and duplicate elimination [3]. The main objective of data cleaning is to reduce the time and complexity of the mining process and increase the quality of datum in the data warehouse. There are several existing data cleaning techniques that are being used for different purposes. 'Similarity functions' are used to find the similarity between records and fields [19]. 'Duplicate elimination functions' are used to determine whether two or more records represent the same real world object [4]. All the existing approaches need to be combined to perform the data cleaning work in a sequential order. This paper proposes a new framework for data cleaning that comprises all the existing data cleaning approaches and new approaches to reduce the complexity of data cleaning process and to clean with more flexibility and less effort.

2. Related Literature

In the merge/purge method duplicates within a single table are detected using application-specific rules. To reduce computation complexity, the table is sorted using a domain-specific key, and only records within a sliding window are compared [20].

Duplicate detection in data cleansing frameworks for fuzzy duplicates *is* pertaining two or more tuples that describe the same real-world entity using different syntaxes. Eliminating fuzzy duplicates is applicable in any database but is critical in data-integration and analytical-processing domains, which involve data warehouses, data mining applications, and decision support systems [27].

Record linkage follows a probabilistic approach [30], [9]. For each record pair, a comparison vector is produced by comparing corresponding attribute values. The record pairs are classified as matched, possibly matched, and unmatched using a linkage rule that assigns each observed comparison vector with a probability for each class. To reduce the number of comparisons, application specific blocking criteria can be used.

Several recent approaches incorporate machine learning into the duplicate detection process. Tejada et al. use a decision tree forest to learn both duplicate detection rules and weights for string transformations, which are used for comparing fields. The string edit distance is a metric commonly used in duplicate detection procedures. Bilenko and Mooney have shown that machine learning techniques increase the accuracy of the field matching task when string edit distance is used, and in some cases even when token based measures are used [5].

Manuscript received May 5, 2008
Manuscript revised May 20, 2008

3. Framework Design

Figure1 shows the framework to clean the data in a sequential order. Each step of the framework is well suited for the different purposes. Some of the data cleaning techniques will be suited for the particular work of the data cleaning process. This framework offers the user interaction by selecting the suitable algorithm. The user has to know each step clearly. This framework will be effective in handling noisy data.

The principle on this framework is as follows:

i. There is a clear need to identify and select attributes. These selected attributes to be used in the other steps.
ii. The well suited token is created to check the similarities between records as well as fields.
iii. Clustering algorithm or blocking method is selected to group the records based on the blocking/clustering key.
iv. There is a need to select similarity functions based on the data type.
v. The elimination function is selected to eliminate the duplicates.
vi. The result or cleaned data is merged by using the merge techniques.

The steps are as follows:

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

3.1 Selection of attributes

Data warehouse can have millions of records and hundreds of columns. The amount of records and attributes and their relativity is unknown to the users. Attribute selection is very important when comparing two records [15]. This step is the foundation step for all the remaining steps. Therefore time and effort are two important requirements to promptly and qualitatively select the attribute to be considered. The attribute itself may cause inconsistencies and redundancies, due to the use of different names to represent the same attribute or same name for different attributes.

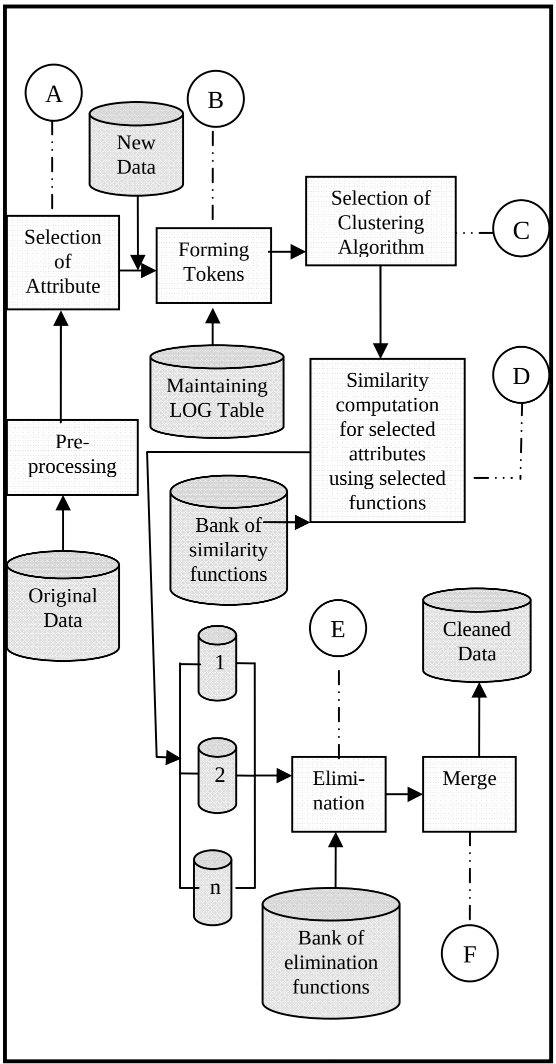
A
New Data
B
Selection of Attribute
Forming Tokens
Selection of Clustering Algorithm
C
Maintaining LOG Table
Similarity computation for selected attributes using selected functions
D
Pre-processing
Bank of similarity functions
Original Data
E
Cleaned Data
1
2
n
Elimi-nation
Merge
Bank of elimination functions
F

Figure 1: Framework for Data Cleaning

The user needs to identify the attributes that must be included in the analysis; relationship with the other attributes; types of data and the number of distinct field values. Based on the above information user needs to assign a ‘weight’ or ‘rank value’ for the selected attributes. Finally, the highest priority attributes are selected [14].

The amount of data may be enormous: hundreds of thousands of records, each with hundreds of parameters (features, fields, variables, attributes). The data cleaning process will be complex with this large amount of data in the data warehouse. The attribute selection is very important to reduce the time and effort for the further work such as record similarity and elimination process etc. This step is to apply a feature subset selection algorithm for reducing the dimensionality of the input set.

3.2 Formation of Tokens

This step makes use of the selected attribute field values to form a token. The tokens can be created for a single attribute field value or for combined attributes. For example, contact name attribute is selected to create a token for further cleaning process. The contact name attribute is split as first name, middle name and last name. Here first name and last name is combined as contact name to form a token.

Tokens are formed using numeric values, alphanumeric values and alphabetic values. The field values are split. Unimportant elements are removed [title tokens like Mr., Dr. and so on [8].

Numeric tokens comprise only digits [0 – 9]. Alphabetic tokens consist of alphabets (aA - zZ). The first character of each word in the field is considered and the characters are sorted. Alphanumeric tokens comprise of both numeric and alphabetic tokens. It composes a given alphanumeric element into numeric [22].

This step is eliminates the need to use the entire string records with multiple passes, for duplicate identification. It also, solves similarity computation problem in a large database by forming token key from some selected fields, to reduce the number of comparisons.

3.3 Selection of Clustering Algorithm

This step selects an algorithm to cluster or group the records based on the block-token key. This block-token key is generated by selecting the first three characters from any one the field of selected attributes. There are several clustering algorithms available to group records that are similar or dissimilar to the objects belonging to another cluster. At present, many data cleaning tools have been developed using blocking methods. Potentially each record in a data set has to be compared with all the records in the data set. The number of record comparisons will be larger. After the application of clustering algorithms to the same, the numbers of records compared are reduced. As well as, the blocking methods can also be used to reduce the huge number o f comparisons. It works in a blocking fashion. i.e They use a record attribute to split the data sets into blocks. There are four blocking methods available such as Standard blocking methods, Bigram indexing, sorted neighborhood method and canopy clustering with TF/IDF [2]. The user can choose clustering algorithm or blocking method to reduce comparisons. The user should select a method according to the data set. The choice of a good blocking method or clustering algorithm can greatly reduce the number of record pair evaluations to be performed and so the user can achieve significant performance analysis.

Data Mining primarily works with large databases. Sorting the large datasets and data duplicate elimination process with this large database faces the scalability problems. The clustering techniques are used to cluster or group the dataset into small group based on the distance values or some threshold values to reduce the time for the elimination process. The data which is not included in the cluster is called as outlier. These clustering techniques will be useful in the elimination process.

3.4 Similarity Computation for Selected Attributes

This step chooses a specific similarity function for each selected attribute. Record linkage algorithms fundamentally depend on string similarity functions for record fields as well as on record similarity functions for string fields. Similarity computation functions depend on the data type. Therefore the user must choose the function according to the attribute’s data type, for example numerical, string and so on.

Different similarity functions are available to calculate similarity between strings. Similarity functions can be categorized into two groups: sequence based similarity functions and token-based similarity functions. Sequence-based similarity functions allow contiguous sequences of mismatched characters. It is defined as the minimum number of insertions, deletions or substitutions necessary to transform one string into another. Sequence-based similarity functions are Hamming distance, String Edit distance, Jaro Winkler string similarity and Jaro. All the above similarity functions are designed and suited for different kinds of strings. For example some functions are suited for short strings or long strings and some functions are based on data types [5].

Sequence-based similarity functions become complicated for larger strings. The token based model avoids these problems by viewing strings as tokens. Token-based similarity functions can be used as the simplest method than the sequence-based similarity functions. The token based similarity functions are Jaccard coefficient, TF-IDF cosine similarity, n-grams and so on [7]. Tokenization is typically performed by treating each individual word of certain minimum length as a separate token or by taking first character from each word. In step2, token has been created for the selected attributes. Each function measures the similarity of selected attributes with other record fields and assigns a similarity value for each field. In the next step, the clustering techniques have been selected to group the fields based on the similarity values.

Accurate similarity functions are important for clustering and duplicate detection problem. Better string distance might also be useful to pair the record as match or non- match. This matching and non-matching pairs is used to group as cluster and eliminate the duplicates.

3.5 Selection of Elimination function

In step5, the user selects the elimination function to eliminate the records. During the elimination process, only one copy of exact duplicated records should be retained and eliminate other duplicate records [1] [7]. The elimination process is very important to produce a cleaned data. The above steps are used to identify the duplicate records. This step is used to detect or remove the duplicate records from one cluster or many clusters. Before the elimination process, the user should know the similarity threshold values for all the records which are available in the data set. The similarity threshold values are important for the elimination process. In the elimination process, select all possible pairs from each cluster and compare records within the cluster using the selected attributes. Most of the elimination processes compare records within the cluster only. Sometimes other clusters may have duplicate records, same value as of other clusters. The comparisons of all the clusters are not at all possible due to the time constraint and efficiency.

The same comparison process has been done in Step4. Step5 also requires a comparison process. Therefore the selection of clustering algorithm or blocking method is important. To reduce a comparison, the user needs to store a threshold value or similarity value as a LOG file. The user should assign threshold criteria to eliminate the duplicate records to find which record is having the lowest threshold value. The elimination process is done in many data cleaning tools based on the threshold values. Several rule-based approaches are proposed for the duplicate elimination process. The distance criteria is the mostly used in the rule based approaches. The commonly available rule based approaches are the ‘Bayes decision rule’ for minimum error, Decision with a Reject Region, ‘Equational theory’ and so on. The user must identify which record has to retain and so on. Finally user needs to select one record to be retained as prime representative and maintain this value in the log file. This primary copy will be used for the incremental cleaning process also for further work.

The duplicate record detection techniques are crucial for improving the quality of the extracted data with imprecise and noisy. This approach can substantially reduce the probability of false mismatches, with a relatively small increase in the running time.

3.6 Merge

This step merges the detected duplicates. The different merging strategies are used to group the record as a single cluster [12]. The user must maintain the merged record and the prime representative as a separate file in the data warehouse. This information helps the user for further changes in the duplicate elimination process.

This merge step will be useful for the incremental data cleaning. Incremental data cleaning deals about checking the new data with the already created file when a new data enters into the data warehouse. So, this reduces the time for the data cleaning process.

4. Conclusion

In the existing data cleaning techniques some of the cleaning methods are implemented. But those existing techniques are good in some part of cleaning process. For example duplicate elimination cleaning tools are suited for data elimination process and similarity cleaning tools is well suited for field similarity and record similarity. To overcome these problems, a new framework is proposed and implemented to comprise all the techniques as a single data cleaning tool.

This new framework consists of six elements: Selection of attributes, Formation of tokens, Selection of clustering algorithm, Similarity computation for selected attributes, Selection of elimination function and Merge. This framework will be useful to develop a powerful data cleaning tool by using the existing data cleaning techniques in a sequential order.

To compare this new framework with previous approaches the token concept is included to speed up the data cleaning process and reduce the complexity. The blocking/Clustering function is used to block the records based on the key vale to reduce the number of comparison and increase speed of the data cleaning process. Each step of this new framework is specified clearly in a sequential order by means of the six data cleaning process offered such as attribute selection, token formation, similarity computation, clustering, elimination, and merge. This will reduce the effort taken by the user. This framework will be flexible for all kind of data in the relational databases.

Reference

[1] R. Ananthakrishna, S. Chaudhuri, and V. Ganti, *Eliminating Fuzzy Duplicates in Data Warehouses*. VLDB, pages 586-597, 2002.

[2] Andrew Borthwick, Martin Buechi, and Arthur Goldberg, *Automated Database Blocking and Record Matching*, U.S. Patent # 7,152,060. Filed April 11, 2003. Awarded December 19, 2006.

[3] Arthur D. Chapman 2005, Principles and *Methods of Data Cleaning – Primary Species and Species-Occurrence Data*, version 1.0. Report for the Global Biodiversity Information Facility,Copenhagen, 2005.

[4] Bitton D and DeWitt, *Duplicate Record Elimination in Large Data Files*, ACM Transactions on Database Systems 8 (1983), No. 2, 255-265. #312.

[5] M. Bilenko and R. J. Mooney. *Adaptive duplicate detection using learnable string similarity measures*. ACM SIGKDD, 39-48, 2003.

[6] Cochinwala M, Dalal S, Elmagarmid A, and Verykios, V, *Record Matching: Past, Present and Future*. Submitted to ACM Computing Surveys, 2003.

[7] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. *Duplicate Record Detection: A Survey*. IEEE TKDE, 19(1):1-16, 2007.

[8] C.I. Ezeife and Timothy E. Ohanekwu, *Use of Smart Tokens in Cleaning Integrated Warehouse Data*, the International Journal of Data Warehousing and Mining (IJDW), Vol. 1, No. 2, pp. 1-22, Ideas Group Publishers, April-June 2005.

[9] M. G. Elfeky, V. S. Verykios, and A. K. Elmagarmid. *TAILOR: A record linkage toolbox*. In Proceedings of the International Conference on Data Engineering (ICDE), pages 17–28, 2002.

[10]Helena Galhardas, Daniela Florescu, and Dennis Shasha, *An Extensible Framework for Data Cleaning,* Proc. of Int. Conf. on Data Engineering (ICDE) ,San Diego , February , 2000.

[11] Helena Galhardas, Daniela Florescu, Eric Simon , Cristian-Augustin Saita, Dennis Shasha, *Declarative Data Cleaning: Language, Model, and Algorithms,* Proc. of the Int. Conf. on Very Large Data Bases (VLDB) ,Rome, Italy , September , 2001

[12] A. M. Hernández and J. S. Stolfo. *The Merge/Purge Problem for Large Databases,* ACM SIGMOD, pages 127-138, 1995.

[13] Jonathan I. Maletic and Andrian Marcus, *Data Cleansing: Beyond Integrity Analysis,* In Proceedings of the Conference on Information Quality (IQ2000).

[14] Jiawei Han, Micheline Kamber, *D*ata Mining: Concepts and Techniques, Publisher: Elsevier Science & Technology Books, March 2006, ISBN-13: 9781558609013

[15] I. Kononenko, S. J. Hong, *Attribute Selection for Modeling,* In: Future Generation Computer Systems, ISSN 0167 - 739X, 13 (2 - 3), pp. 181 - 195, 1997

[16] E. M. Knorr and R. T. Ng, *Algorithms for Mining Distance-Based Outliers in Large Datasets,* Proc. of the 24th International Conference on Very Large Data Bases (VLDB), pages 392-403, 1998.

[17] M. L. Lee, H. Lu, T. W. Ling, and Y. T. Ko. *Cleansing Data for Mining and Warehousing*, DEXA, 751-760, 1999.

[18] Lifang Gu, Rohan Baxter, Deanne Vickers, and Chris Rainsford, *Record Linkage: Current Practice and Future Directions*, CMIS Technical Report No. 03/ 83, Apr. 2003.

[19] A. E. Monge and C. P. Elkan. *The field matching problem: Algorithms and applications,* SIGMOD workshop on research issues on knowledge discovery and data mining, pages 267-270, 1996.

[20] Mauricio Hernandez, Salvatore Stolfo, *Real World Data is Dirty: Data Cleansing and The Merge/Purge Problem,* Journal of Data Mining and Knowledge Discovery, 1(2), 1998.

[21] Mong-Li Lee, Tok Wang Ling, Hongjun Lu, and Yee Teng Ko, *Cleansing Data for Mining and Warehousing*, In Proceedings of the International Conference on Database and Expert Systems Applications (DEXA), volume 1677 of LNCS, pages 751-760, Florence, Italy, 1999.

[22] 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.

[23] Oded Maimon, Jonathan I. Maletic and Andrian Marcus, *Data Cleansing, Data Mining and Knowledge Discovery Handbook*, Springer US, Pages 21-36, 2005.

[24] D Pyle, *Data Preparation for Data Mining*, Morgan Kaufmann Inc., 1999, ISBN 1-55860-529-0.

[25] Rahm, Erhard; Do, Hong-Hai, Data *Cleaning: Problems and Current Approaches,* IEEE Bulletin of the Technical Committee on Data Engineering, Vol 23 No. 4, December 2000

[26] Rohan Baxter, Peter Christen and Tim Churches, *A Comparison of Fast Blocking Methods for Record Linkage, Workshop on Data Cleaning, Record Linkage and Object Consolidation,* KDD, August 24-27, 2003.

[27] H.H. Shahri; S.H. Shahri, *Eliminating Duplicates in Information Integration: An Adaptive, Extensible Framework,* Intelligent Systems, IEEE, Volume 21, Issue 5, Sept.-Oct. 2006 Page(s):63 – 71

[28] Verykois, V.S. and Elmagarmid, A.K., *Automating the Approximate Record Matching Process*, Computer Sciences Dept., Purdue University, West Lafayette, IN, Jun. 15, 1999.

[29] Vijayshankar Raman and Joseph M. Hellerstein, *An Interactive Framework for Data Cleaning,* Computer Science Division (EECS), University of California, Report No. UCB/CSD-0-1110, September 2000

[30] W. E. Winkler. *Matching and record linkage.* In Business Survey Methods. Wiley-Interscience, 1995.