Dynamic Approach for Data Scrubbing Process

Israr Ahmed
Department of Computer Sciences
University of Central Punjab
Main Gulberg, Lahore, Pakistan
israr.ahmed@ucp.edu.pk

Abdul Aziz
Department of Computer Sciences
University of Central Punjab
Main Gulberg, Lahore, Pakistan
aziz@ucp.edu.pk

***Abstract*—It is very difficult to over-emphasize the benefits of accurate data. Errors in data are generally the most expensive aspect of data entry, costing the users even much more compared to the original data entry. Unfortunately, these costs are intangibles or difficult to measure. If errors are detected at an early stage then it requires little cost to remove the errors. Incorrect and misleading data lead to all sorts of unpleasant and unnecessary expenses. Unluckily, it would be very expensive to correct the errors after the data has been processed, particularly when the processed data has been converted into the knowledge for decision making. No doubt a stitch in time saves nine i.e. a timely effort will prevent more work at later stage. Moreover, time spent in processing errors can also have a significant cost. One of the major problems with automated data entry systems are errors. In this paper we discuss many well known techniques to minimize errors, different cleansing approaches and, suggest how we can improve accuracy rate. Framework available for data cleansing offer the fundamental services such as attribute selection, formation of tokens, selection of clustering algorithms, selection of eliminator functions etc.**

Keywords: Data warhouse, data cleansing, data quality, data mining,error detection, data anomalies

I. INTRODUCTION

Data quality means error free mechanism in data warehouse. The quality of data needs to be improved by using the data cleaning techniques. Existing data cleaning techniques used to identify record duplicates, missing values, record and field similarities and duplicate elimination [1]. 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 [2]. Clean data is crucial for a wide variety of applications in many industries. When data has kept increasing in an explosive rate, a task to keep data correct and consistent can be overwhelming. 'Duplicate elimination functions' are used to determine whether two or more records represent the same real world object [3]. 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.

II. DATA QUALITY

To be processable and interpretable in an effective and efficient manner, data has to satisfy a set of quality criteria. Data satisfying those quality criteria is said to be of high quality. In general, data quality is defined as an aggregated value over a set of quality criteria [4]. We describe the set of criteria that are affected by comprehensive data cleansing and define how to assess scores for each one of them for an existing data collection. To measure the quality of a data collection scores have to be accessed for each of the quality criteria. The assessment of scores for quality criteria can be used to quantify the necessity of data cleansing for a data collection as well as the success of a performed data cleansing process on a data collection. Quality criteria can also be used within optimization of data cleansing by specifying priorities for each of the criteria which in turn influences the execution of data cleansing methods affecting the specific criteria.

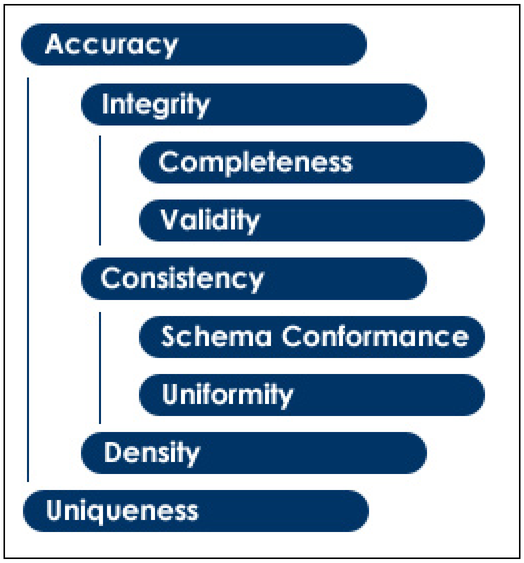

Fig 1 Data quality criteria hierarchy

Data must conform to the set of quality criteria, Figure 1 describes the set of criteria of data quality, and data must fulfill each criterion to ensure its quality.

A. Accuracy

Accuracy is described as an aggregated value over the quality criteria integrity, consistency and density. It can also be termed as quotient of the number of correct values to the total number of values.

B. Integrity

It defines how the data records are consistent and merged to give the formal look. It can be classified into completeness and validity and aggregating these results data quality. An integral data collection contains representations of all the entities in the mini-world and only of those, i.e., there are no invalid tuples or integrity constraint violations as well as no missing tuples. Data integrity can be compromised in a number of ways: Human errors when data is entered, Errors that occur when data is transmitted from one computer to another, Software bugs or viruses, Hardware malfunctions, such as disk crashes

C. Completeness

It is defined as the quotient of entities from M being represented by a tuple in r and the overall number of entities in M. Achieving this form of completeness is not a primary data cleansing concern but more of a data integration problem [5]. We achieve completeness within data cleansing by correcting tuples containing anomalies and not just deleting these tuples if they are representations of entities from M.

D. Validity

It is the quotient of entities from M being represented by tuples in r and the total amount of tuples in r, i.e., the percentage of tuples in r representing (valid) entities from M. The identification of invalid tuples is complicated and sometimes impossible because of the inability or high cost for repeating measurements to verify the correctness of a measured value.

E. Consistency

It deals with the syntactical anomalies as well as contradictions. It is further divided into schema conformance and uniformity forming another aggregated value over quality criteria. Intuitively a consistent data collection is syntactically uniform and free of contradictions. In brief, data consistency we mean validity, accuracy, usability and integrity of over all data across the entire enterprise. We make it sure that each user observes a consistent view of the data.

F. Schema Conformance

Quotient of tuples in r conforms to the syntactical structure defined by schema R and the overall number of tuples in r. Some systems do not enforce the complete syntactical structure thus allowing for tuples within the collection that are not absolutely format conform [6]. This is especially true for the relational database systems where the adherence of domain formats is incumbent to the user.

G. Uniformity

It is directly related to irregularities, i.e., the proper use of values within each attribute. Uniformity is the quotient of attributes not containing irregularities in their values.

H. Density

It is quotient of missing values in the tuples of r and the number of total values that ought to be known because they exist for a represented entity. There still can be values or properties non-existent that have to be represented by null values having the exact meaning of not being known. These are no downgrades of data quality. It would be a downgrade if we try to estimate a value for them.

I. Uniqueness

It is the quotient of tuples representing the same entity in the mini-world and the total number of tuples in r. A collection that is unique does not contain duplicates. Recalling the definition of accuracy as a collection not containing any anomalies except duplicates, a data collection being accurate and unique does not contain any of the anomalies.

III. Data Cleansing

A process used to determine inaccurate, incomplete, or unreasonable data and then improving the quality through correction of detected errors and omissions. The process may include format checks, completeness checks, reasonableness checks, limit checks, review of the data to identify outliers (geographic, statistical, temporal or environmental) or other errors, and assessment of data by subject area experts (e.g. taxonomic specialists) [7]. These processes produce the results in the form of flagging, subsequent checking and correction of suspect records. The general framework for data cleansing is:

- Define and determine errors
- Search and identify error instances
- Correct the errors
- Error instance and document error types
- Updating the entry mechanism to avoid future errors

There are a number of terms used by different researchers to refer largely to the same process. It is a matter of preference what one uses. The most common terms include Error checking, detection, validation, cleansing, scrubbing and correction.

IV. Data Cleansing Process

Data cleansing is defined as the entirety of operations performed on existing data to remove anomalies and receive a data collection being an accurate and unique representation of the mini-world. It is a (semi) automatic process of operations performed on data that perform, preferable in the order (i) format adaptation for tuples and values, (ii) integrity constraint enforcement, (iii) derivation of missing values from existing ones, (iv) removing contradictions within or between tuples, (v) merging and eliminating duplicates, and (vi) detection of outliers, i.e., tuples and values having a high potential of being invalid. Data cleansing may include structural transformation, i.e. transforming the data into a format that is better manageable or better fitting the mini-world [8]. Data cleansing process comprises the three major steps (i) auditing data to identify the types of anomalies reducing the data quality, (ii) choosing appropriate methods to automatically detect and remove them, and (iii) applying the methods to the tuples in the data collection. Steps (ii) and (iii) can be seen as specification and execution of a data cleansing workflow. We add another task (iv), the post-processing or control step where we exam the results and perform exception handling for the tuples not corrected within the actual processing. Figure 2 shows the steps within the data cleansing process. The specification of the data cleansing process and the control of its execution is sponsors) done by one or more domain experts, i.e., experts with knowledge about the mini-world and its regularities and peculiarities.

The process of data cleansing normally never finishes, because anomalies like invalid tuples are very hard to find and eliminate. Depending on the intended application of the data it has to be decided how much effort is required to spend for data cleansing.

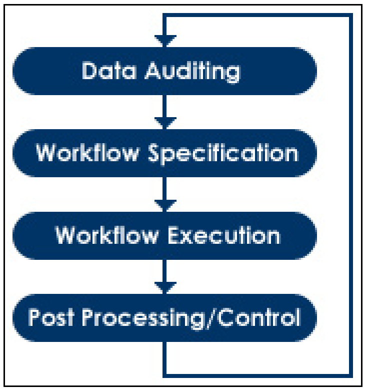

Fig 2 Data cleansing process

A. Data Auditing

First step in data cleansing process is auditing the data to find the types of anomalies contained within it. The data is audited using statistical methods and parsing the data to detect syntactical anomalies. The instance analysis of individual attributes (data profiling) and the whole data collection (data mining) derives information such as minimal and maximal length, value range, frequency of values, variance, uniqueness, occurrence of null values, typical string patterns as well as patterns specific in the complete data collection (functional dependencies and association rules).

The results of auditing the data support the specification of integrity constraints and domain formats. Integrity constraints are depending on the application domain and are specified by domain expert. Each constraint is checked to identify possible violating tuples. For one-time data cleansing only those constraints that are violated within the given data collection has to be further regarded within the cleansing process [9].
Data auditing in the data cleansing process should be an indication for each of the possible anomalies to whether it occurs within the data collection and with which kind of characteristics. For each of these occurrences a function, called tuple partitioner, for detecting all of its instances in the collection should be available or directly inferable.

B. Workflow Specification

Detection and elimination of common order problems is done by applying the multiple operations over the data. This is called the data cleansing workflow. It is specified after auditing the data to gain information about the existing anomalies in the data collection at hand. One of the main challenges in data cleansing insists in the specification of a cleansing workflow that is to be applied to the dirty data automatically eliminating all anomalies in the data.

For the specification of the operations intending to modify erroneous data the cause of anomalies have to be known and closely considered. The causes for anomalies are manifold. Typical causes for anomalies are impreciseness in measurement or systematic errors in experimental setup, false statements, and inconsistent use of abbreviations, misuse or misinterpretation of data input fields, incorrect or careless interpretation of the analysis results leading to invalid tuples results and to a propagation of errors.

C. WorkflowEexecution

The data cleansing workflow is executed after specification and verification of its correctness. The implementation should enable an efficient performance even on large sets of data.

D. Controlling and Post-processing

On the successful execution of cleansing workflow, retrieved results are checked again to verify the correctness of the specified operations. Within the controlling step the tuples that could not be corrected initially are inspected intending to correct them manually [10]. The output is new cycle of cleansing process, starting by auditing the data and searching for characteristics in exceptional data. This might be supported by learning sequences of cleansing operations for certain anomalies i.e., the expert cleanses one tuple by example and the system learns from this to perform the cleansing of other occurrences of the anomaly automatically.

V. Methods of Data Cleansing

Data cleansing requires a person or team reading through a set of records and verifying their accuracy. Typographical errors and spelling errors are corrected, incomplete or missing entries are completed. In complex situation, data cleansing can be achieved by software automatically. These data cleansing software can verify the data with a number of rules and procedures laid down by the user. A data cleansing program can be configured to correct any misspelled words, and remove duplicate tuples. A more sophisticated data cleansing program may be able to fill in a missing value of certain attributes like city based on a correct zip code. Some types of errors require deeper inspection and analysis. One can view this as of the data elements conform to a general form. Two things are done here; identifying outliers or strange variations in data. What data is supposed to look like allows errors to be uncovered. Below is the set of general methods that can be used to detect errors.

A. Statistical

Identify outlier fields and records using as mean, standard deviation, range, based on Chebyshev's theorem and considering the confidence intervals for each field. While this approach may generate many false positives, it is simple and fast, and can be used in conjunction with other methods.

B. Clustering

Identify outlier records using clustering techniques based on Euclidian (or other) distance. Some clustering algorithms provide support for identifying outliers. The main drawback of these methods is a high computational complexity.

C. Pattern Based

Identify outlier fields and rows that do not fit into the existing identified patterns in the data. By integrating the techniques like classification, partitioning is used to find out the patterns that apply to most records.

D. Association Rules

Association rules with high confidence and support do follow these rules are considered outliers. The power of association rules is that they can deal with data of different types.

VI. Framework of Data Cleansing

Framework of data cleansing performs its execution in sequential order. Figure 3 represents the framework to clean the data in sequential order. Each step of framework is well suited for different purposes [11]. 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 the framework is as follows:

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

The steps performed are as follows:

- Selection of attributes
- Formation of tokens
- Selection of clustering algorithm
- Similarity computation for selected attributes
- Selection of elimination function
- Merge

A. 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. This step is the foundation step for all the remaining steps [12]. 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.

B. 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. 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 [13].

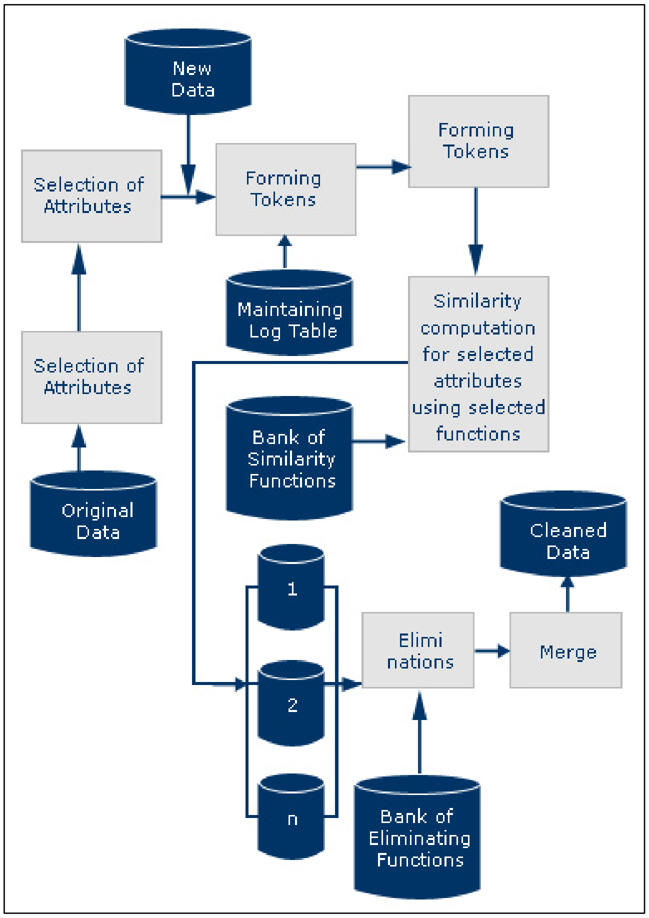

Fig 3 Data cleansing Framework

C. 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 [14]. The number of record comparisons will be larger. After the application of clustering algorithms to the same, the numbers of records compared are reduced.

D. 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.

E. 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 [15]. 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 [16]. 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.

VII. Data Cleansing Problems

There exist severe data quality problems that can be solved by data cleansing and transformation. As we will see, these problems are closely related and should thus be treated in a uniform way. Data transformations [17] are needed to support any changes in the structure, representation or content of data. These transformations become necessary in many situations, e.g., to deal with schema evolution migrating a legacy system to a new information system, or when multiple data sources are to be integrated. As shown in Figure 4 we roughly distinguish between single-source and multi-source problems and between schema- and instance-related problems. Schema-level problems of course are also reflected in the instances; they can be addressed at the schema level by an improved schema design (schema evolution), schema translation and schema integration. Instance-level problems, on the other hand, refer to errors and inconsistencies in the actual data contents which are not visible at the schema level [18]. They are the primary focus of data cleaning. Figure 4 also indicates some typical problems for the various cases. While not shown in Figure 4, the single-source problems occur (with increased likelihood) in the multi-source case, too, besides specific multi-source problems.

A. *Single Source Problems*

The data quality of a source largely depends on the degree to which it is governed by schema and integrity constraints controlling permissible data values. For sources without schema, such as files, there are few restrictions on what data can be entered and stored, giving rise to a high probability of errors and inconsistencies. Database systems, on the other hand, enforce restrictions of a specific data model (e.g., the relational approach requires simple attribute values, referential integrity, etc.) as well as application-specific integrity constraints. Schema related data quality problems thus occur because of the lack of appropriate model-specific or application-specific integrity constraints, e.g., due to data model limitations or poor schema design, or because only a few integrity constraints were defined to limit the overhead for integrity control. Instance-specific problems relate to errors and inconsistencies that cannot be prevented at the schema level (e.g., misspellings).

For both schema- and instance-level problems we can differentiate different problem scopes: attribute (field), record, record type and source; examples for the various cases are shown in Table 1.

Table 1: Single source problems

Scope/Problem		**Dirty Date**	**Reason**
Attribute	Illegal values	bdate = 35.10.80	Value outside domain range
Record	Violated attribute dependencies	Age = 35 ,bdate = 15.05.85	Age= current year – birth year should hold

B. *Multi source Problems*

The problems present in single sources are aggravated when multiple sources need to be integrated. Each source may contain dirty data and the data in the sources may be represented differently, overlap or contradict. This is because the sources are typically developed, deployed and maintained independently to serve specific needs. This results in a large degree of heterogeneity with respect to data management systems, data models, schema designs and the actual data.

At the schema level, data model and schema design differences are to be addressed by the steps of schema translation and schema integration, respectively. The main problems with respect to schema designs are naming and structural conflicts [19]. Naming conflicts arise when the same name is used for different objects (homonyms) or different names are used for the same object (synonyms). Structural conflicts occur in many variations and refer to different representations of the same object in different sources, e.g., attribute vs. table representation, different component structure, different data types, different integrity constraints, etc.

Table 2: Multi source problems

No	Last Name	First Name	Zip	Phone
1	Alice	Christ	5250	123-456-789
2	Bob	Smith	54522	852-456-1

Table 2 represents both relational formats but exhibit schema and data conflicts. At the schema level, there are name conflicts (synonyms Customer/Client, Cid/Cno, Sex/Gender) and structural conflicts (different representations for names and addresses). At the instance level, we note that there are different gender representations (”0”/”1” vs.”F”/”M”) and presumably a duplicate record (Kristen Smith). The latter observation also reveals that while Cid/Cno is both source-specific identifiers, their contents are not comparable between the sources; different numbers (11/493) may refer to the same person while different persons can have the same number (24).

VIII. DATA CLEANSING APPROACHES

With the definition of anomalies occurring in data, the quality criteria affected by them, and a description of the data cleansing process and the methods used within it, we are now able to compare existing data cleansing approaches.

A. *AJAX*

AJAX is an extensible and flexible framework attempting to separate the logical and physical levels of data cleansing. The logical level supports the design of the data cleansing workflow and specification of cleansing operations performed, while the physical level regards their implementation. AJAX major concern is transforming existing data from one or more data collections into a target schema and eliminating duplicates within this process [20]. For this purpose a declarative language based on a set of five transformation operations is defined. The transformations are mapping, view, matching, clustering, and merging.

B. FraQL

FraQL is another declarative language supporting the specification of a data cleansing process. The language is an extension to SQL based on an object-relational data model. It supports the specification of schema transformations as well as data transformations at the instance level, i.e., standardization and normalization of values [21]. This can be done using user-defined functions. The implementation of the user defined function has to be done for the domain specific requirements within the individual data cleansing process.

C. Potter's Wheel

Potter's Wheel is an interactive data cleansing system that integrates data transformation and error detection using spreadsheet-like interface. The effects of the performed operations are shown immediately on tuples visible on screen. Error detection for the whole data collection is done automatically in the background. A set of operations, called transforms, are specified that support common schema transformations without explicit programming [22]. These are value translations, which apply a function to every value in a column, One-to-one mappings that are column operations transforming individual rows, and Many-to-many mappings of rows solving schematic heterogeneities where information is stored partly in data values, and partly in the schema. The anomalies handled by this approach are syntax errors and irregularities.

D. Intelli Clean

Intelli Clean is a rule based approach to data cleansing with the main focus on duplicate elimination. The proposed framework consists of three stages [23]. In the Pre- Processing stage syntactical errors are eliminated and the values are standardized in format and consistency of used abbreviations. It's not specified in detail; how this is accomplished [24]. The processing stage represents the evaluation of cleansing rules on the conditioned data items that specify actions to be taken under certain circumstances [25]. There are four different classes of rules. Duplicate identification rules specify the conditions under which tuples are classified as duplicates. Merge/Purge rules specify how duplicate tuples are to be handled. It is not specified how the merging is to be performed or how its functionality can be declared. If no merge/purge rule has been specified, duplicate tuples can also manually be merged at the next stage. Update rules specify the way data is to be updated in a particular situation [26]. This enables the specification of integrity constraint enforcing rules. For each integrity constraint an update rule defines how to modify the tuple in order to satisfy the constraint. Update rules can also be used to specify how missing values ought to be filled-in. Alert rules specify conditions under which the user is notified allowing for certain actions.

IX. Conclusion

Data cleansing is applied with different framework and within different areas of the data integration and management process. It is defined as the sequence of operations intending to enhance to overall data quality. There is only a rough description of the procedure in data cleansing as it is highly domain dependent and explorative. Existing data cleansing approaches mostly focus on the transformation of data and the elimination of duplicate records. Some approaches enable the declarative specification of a more comprehensive data cleansing processes, still leaving most of the implementation details for the cleansing operation. Different approaches differ from each other on the basis of mechanism they use for data cleansing process. Approaches also differ on the basis of problem domain and also the problem nature so the data cleansing approach should be selected that best suited for your problem domain. No doubt a stitch in time saves nine i.e. a timely effort will prevent more work at later stage.

References

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

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

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

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

[5] 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), 2002, pp 17–28

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

[7] 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.

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

[9] 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.

[10] 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, pp 63 – 71

[11] Bernstein, P.A.; Bergstraesser, T.: Metadata Support for Data Transformation Using Microsoft Repository. ACM SIGMOD International conference on Management of data. Maryland, USA,2005, pp :9- 14

[12] Andritsos, P. et al. Information-Theoretic Tools for Mining Database Structure from Large Data Sets, ACM SIGMOD international conference on Management of data. Paris, France, 2004, pp. 731-742.

[13] Bohannon, P "A Cost-Based Model and Effective Heuristic for Repairing Constraints" ACM SIGMOD International conference on Management of data. Maryland, USA,2005, pp. 143-154.

[14] Ilyas, I. F. et al, 2004. CORDS: Automatic Discovery of Correlations and Soft Functional Dependencies, ACM SIGMOD international conference on Management of data. Paris, France, 2004 , pp. 647-658.

[15] Fayyad, U.: Mining Database: Towards Algorithms for Knowledge Discovery. IEEE Techn. Bulletin Data Engineering 21(1), 1998.

[16] H. Galhardas, D. Florescu, D. Shasha, E. Simon, C.-A. Saita "Improving data cleaning quality using a data lineage facility", Proceedings of the 3rd International Workshop on Design and Management of Data Warehouses, Interlaken, Switzerland, June 2001

[17] J.I. Maletic, A. Marcus "Data Cleansing: Beyond Integrity Analysis" Proceedings of the Conference on Information Quality, October 2000.

[18] E. Rahm, Hong Hai Do "Data Cleaning: Problems and current approaches", IEEE Bulletin of the Technical Committee on Data Engineering, 2000.

[19] K.-U. Sattler, E. Schallehn "A Data Preparation Framework based on a Multidatabase Language" International Database Engineering Applications Symposium (IDEAS), Grenoble, France, 2001.

[20] Sung, S. Y., Li, Z., & Sun, P. A fast filtering scheme for large database cleansing, Proceedings of Eleventh ACM International Conference on Information and Knowledge Management, VA. 76-83. 2002 November 04-09

[21] Wang, R., Ziad, M., & Lee, Y. W., Data Quality. Kluwer, 2001.

[22] Yu, D., Sheikholeslami, G., & Zhang, A. FindOut: Finding Outliers in Very Large Datasets, Knowledge and Information Systems, 4(4):387-412, 2002

[23] G. Ozsoyoglu, D. A. Singer, and S. S. Chung, "Anti-tamper databases: Querying encrypted databases," in Proceedings of the 17th Annual IFIP WG 11.3 Working Conference on Database and Applications Security, Estes Park, Colorado, Aug. 4-6 2003.

[24] H. Hacigumus, B. R. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data data in the database-service-provider model," in Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, June 4-6 2002, pp. 216–227

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

[26] Bohannon "A Cost-Based Model and Effective Heuristic for Repairing Constraints" ACM SIGMOD International conference on Management of data. Maryland, USA,2005, pp. 143-154

AUTHORS PROFILE

Israr Ahmed is MS student in Computer Science. He got his BS degree in CS from University of Central Punjab. Apart from studying he is working as Senior Software Engineer for a multinational company and has developed and deployed a large number of software. His research interest includes Data Mining, Advance Database Management Systems and Data warehousing.

Prof. Dr. Abdul Aziz did his M.Sc. from University of the Punjab, Pakistan in 1989; M.Phil and Ph.D in Computer Science from University of East Anglia, UK. He secured many honors and awards during his academic career from various institutions. He is currently working as full Professor at the University of Central Punjab, Lahore, Pakistan. He is the founder and Chair of Data Mining Research Group at UCP. Dr. Aziz has delivered lectures at many universities as guest speaker. He has published text books and large number of research papers in different refereed international journals and conferences. His research interests include Knowledge Discovery in Databases (KDD) - Data Mining, Pattern Recognition, Data Warehousing and Machine Learning.

He is member of editorial board for various well known international journals/ conferences including IEEE publications. (e-mail: aziz@ucp.edu.pk).