2

Data Cleansing: A Prelude to Knowledge Discovery

Jonathan I. Maletic[1] and Andrian Marcus[2]

[1] Kent State University
[2] Wayne State University

Summary. This chapter analyzes the problem of data cleansing and the identification of potential errors in data sets. The differing views of data cleansing are surveyed and reviewed and a brief overview of existing data cleansing tools is given. A general framework of the data cleansing process is presented as well as a set of general methods that can be used to address the problem. The applicable methods include statistical outlier detection, pattern matching, clustering, and Data Mining techniques. The experimental results of applying these methods to a real world data set are also given. Finally, research directions necessary to further address the data cleansing problem are discussed.

Key words: Data Cleansing, Data Cleaning, Data Mining, Ordinal Rules, Data Quality, Error Detection, Ordinal Association Rules

2.1 INTRODUCTION

The quality of a large real world data set depends on a number of issues (Wang *et al*., 1995, Wang *et al*., 1996), but the source of the data is the crucial factor. Data entry and acquisition is inherently prone to errors, both simple and complex. Much effort can be allocated to this front-end process with respect to reduction in entry error but the fact often remains that errors in a large data set are common. While one can establish an acquisition process to obtain high quality data sets, this does little to address the problem of existing or legacy data. The field errors rates in the data acquisition phase are typically around 5% or more (Orr, 1998, Redman, 1998) even when using the most sophisticated measures for error prevention available. Recent studies have shown that as much as 40% of the collected data is dirty in one way or another (Fayyad *et al*., 2003).

For existing data sets the logical solution is to attempt to cleanse the data in some way. That is, explore the data set for possible problems and endeavor to correct the errors. Of course, for any real world data set, doing this task by hand is completely out of the question given the amount of person hours involved. Some organizations spend millions of dollars per year to detect data errors (Redman, 1998). A manual

O. Maimon, L. Rokach (eds.), *Data Mining and Knowledge Discovery Handbook*, 2nd ed.,
DOI 10.1007/978-0-387-09823-4_2,

process of data cleansing is also laborious, time consuming, and itself prone to errors. Useful and powerful tools that automate or greatly assist in the data cleansing process are necessary and may be the only practical and cost effective way to achieve a reasonable quality level in existing data.

While this may seem to be an obvious solution, little basic research has been directly aimed at methods to support such tools. Some related research addresses the issues of data quality (Ballou and Tayi, 1999, Redman, 1998, Wang *et al*., 2001) and some tools exist to assist in manual data cleansing and/or relational data integrity analysis.

The serious need to store, analyze, and investigate such very large data sets has given rise to the fields of Data Mining (DM) and data warehousing (DW). Without clean and correct data the usefulness of Data Mining and data warehousing is mitigated. Thus, data cleansing is a necessary precondition for successful knowledge discovery in databases (KDD).

2.2 DATA CLEANSING BACKGROUND

There are many issues in data cleansing that researchers are attempting to tackle. Of particular interest here, is the search context for what is called in literature and the business world as "dirty data" (Fox *et al*., 1994, Hernandez and Stolfo, 1998, Kimball, 1996). Recently, Kim (Kim *et al*., 2003) proposed a taxonomy for dirty data. It is a very important issue that will attract the attention of the researchers and practitioners in the field. It is the first step in defining and understanding the data cleansing process.

There is no commonly agreed formal definition of data cleansing. Various definitions depend on the particular area in which the process is applied. The major areas that include data cleansing as part of their defining processes are: data warehousing, knowledge discovery in databases, and data/information quality management (e.g., Total Data Quality Management TDQM).

In the data warehouse user community, there is a growing confusion as to the difference between *data cleansing* and *data quality*. While many data cleansing products can help in transforming data, there is usually no persistence in this cleansing. Data quality processes ensure this persistence at the business level. Within the data warehousing field, data cleansing is typically applied when several databases are merged. Records referring to the same entity are often represented in different formats in different data sets. Thus, duplicate records will appear in the merged database. The issue is to identify and eliminate these duplicates. The problem is known as the *merge/purge problem* (Hernandez and Stolfo, 1998). In the literature instances of this problem are referred to as record linkage, semantic integration, instance identification, or the object identity problem. There are a variety of methods proposed to address this issue: knowledge bases (Lee *et al*., 2001), regular expression matches and user-defined constraints (Cadot and di Martion, 2003), filtering (Sung *et al*., 2002), and others (Feekin, 2000, Galhardas, 2001, Zhao *et al*., 2002).

Data is deemed unclean for many different reasons. Various techniques have been developed to tackle the problem of data cleansing. Largely, data cleansing is an interactive approach, as different sets of data have different rules determining the validity of data. Many systems allow users to specify rules and transformations needed to clean the data. For example, Raman and Hellerstein (2001) propose the use of an interactive spreadsheet to allow users to perform transformations based on user-defined constraints, Galhardas (2001) allows users to specify rules and conditions on a SQL-like interface, Chaudhuri, Ganjam, Ganti and Motwani (2003) propose the definition of a *reference pattern* for records using fuzzy algorithms to match existing ones to the reference, and Dasu, Vesonder and Wright (2003) propose using business rules to define constraints on the data in the entry phase.

From this perspective data cleansing is defined in several (but similar) ways. In (Galhardas, 2001) data cleansing is the process of eliminating the errors and the inconsistencies in data and solving the object identity problem. Hernandez and Stolfo (1998) define the data cleansing problem as the merge/purge problem and proposes the basic sorted-neighborhood method to solve it.

Data cleansing is much more than simply updating a record with good data. Serious data cleansing involves decomposing and reassembling the data. According to (Kimball, 1996) one can break down the cleansing into six steps: elementizing, standardizing, verifying, matching, house holding, and documenting. Although data cleansing can take many forms, the current marketplace and technologies for data cleansing are heavily focused on customer lists (Kimball, 1996). A good description and design of a framework for assisted data cleansing within the merge/purge problem is available in (Galhardas, 2001).

Most industrial data cleansing tools that exist today address the duplicate detection problem. Table 2.1 lists a number of such tools. By comparison, there were few data cleansing tools available five years ago.

Table 2.1. Industrial data cleansing tools circa 2004

Tool	Company
Centrus Merge/Purge	*Qualitative Marketing Software,* http://www.qmsoft.com/
Data Tools Twins	*Data Tools,* http://www.datatools.com.au/
DataCleanser DataBlade	*Electronic Digital Documents,* http://www.informix.com
DataSet V	*iNTERCON http://www.ds-dataset.com*
DeDuce	*The Computing Group*
DeDupe	*International Software Publishing*
dfPower	*DataFlux Corporation,* http://www.dataflux.com/
DoubleTake	*Peoplesmith,* http://www.peoplesmith.com/
ETI Data Cleanse	*Evolutionary Technologies Intern,* http://www.evtech.com
Holmes	*Kimoce,* http://www.kimoce.com/
i.d.Centric	*firstLogic,* http://www.firstlogic.com/
Integrity	*Vality,* http://www.vality.com/
matchIT	*helpIT Systems Limited,* http://www.helpit.co.uk/
matchMaker	*Info Tech Ltd,* http://www.infotech.ie/
NADIS Merge/Purge Plus	*Group1 Software,* http://www.g1.com/
NoDupes	*Quess Inc,* http://www.quess.com/nodupes.html
PureIntegrate	*Carleton,* http://www.carleton.com/products/View/index.htm
PureName PureAddress	*Carleton,* http://www.carleton.com/products/View/index.htm
QuickAdress Batch	*QAS Systems*, http://207.158.205.110/
reUnion and MasterMerge	*PitneyBowes,* http://www.pitneysoft.com/
SSA-Name/Data Clustering Engine	*Search Software America* http://www.searchsoftware.co.uk/
Trillium Software System	*Trillium Software,* http://www.trilliumsoft.com/
TwinFinder	*Omikron,* http://www.deduplication.com/index.html
Ultra Address Management	*The Computing Group*

Total Data Quality Management (TDQM) is an area of interest both within the research and business communities. The data quality issue and its integration in the entire information business process are tackled from various points of view in the literature (Fox *et al.*, 1994, Levitin and Redman, 1995, Orr, 1998, Redman, 1998, Strong *et al.*, 1997, Svanks, 1984, Wang *et al.*, 1996). Other works refer to this as the enterprise data quality management problem. The most comprehensive survey of the research in this area is available in (Wang *et al.*, 2001).

Unfortunately, none of the mentioned literature explicitly refers to the data cleansing problem. A number of the papers deal strictly with the process management issues from data quality perspective, others with the definition of data quality. The later category is of interest here. In the proposed model of data life cycles with application to quality (Levitin and Redman, 1995) the data acquisition and data usage cycles contain a series of activities: assessment, analysis, adjustment, and discarding of data. Although it is not specifically addressed in the paper, if one integrated the data cleansing process with the data life cycles, this series of steps would define it in the proposed model from the data quality perspective. In the same framework of data quality, (Fox *et al.*, 1994) proposes four quality dimensions of the data: accuracy, current-ness, completeness, and consistency. The correctness of data is defined in terms of these dimensions. Again, a simplistic attempt to define the data cleansing process within this framework would be the process that assesses the correctness of data and improves its quality.

More recently, data cleansing is regarded as a first step, or a preprocessing step, in the KDD process (Brachman and Anand, 1996, Fayyad *et al.*, 1996) however no precise definition and perspective over the data cleansing process is given. Various KDD and Data Mining systems perform data cleansing activities in a very domain specific fashion. In (Guyon *et al.*, 1996) informative patterns are used to perform one kind of data cleansing by discovering *garbage patterns* – meaningless or mislabeled patterns. Machine learning techniques are used to apply the data cleansing process in the written characters classification problem. In (Simoudis *et al.*, 1995) data cleansing is defined as the process that implements computerized methods of examining databases, detecting missing and incorrect data, and correcting errors. Other recent work relating to data cleansing includes (Bochicchio and Longo, 2003, Li and Fang, 1989).

Data Mining emphasizes data cleansing with respect to the garbage-in-garbage-out principle. Furthermore, Data Mining specific techniques can be used in data cleansing. Of special interest is the problem of outlier detection where the goal is to find out exceptions in large data sets. These are often an indication of incorrect values. Different approaches have been proposed with many based on the notion of distance-based outliers (Knorr and Ng, 1998, Ramaswamy *et al.*, 2000). Other techniques such as *FindOut* (Yu *et al.*, 2002) combine clustering and outlier detection. Neural networks are also used in this task (Hawkins *et al.*, 2002), and outlier detection in multi-dimensional data sets is also addressed (Aggarwal and Yu, 2001).

2.3 GENERAL METHODS FOR DATA CLEANSING

With all the above in mind, data cleansing must be viewed as a process. This process is tied directly to data acquisition and definition or is applied after the fact, to improve data quality in an existing system. The following three phases define a data cleansing process:

- Define and determine error types
- Search and identify error instances
- Correct the uncovered errors

Each of these phases constitutes a complex problem in itself, and a wide variety of specialized methods and technologies can be applied to each. The focus here is on the first two aspects of this generic framework. The later aspect is very difficult to automate outside of a strict and well-defined domain. The intention here is to address and automate the data cleansing process outside domain knowledge and business rules.

While data integrity analysis can uncover a number of possible errors in a data set, it does not address more complex errors. Errors involving relationships between one or more fields are often very difficult to uncover. These types of errors require deeper inspection and analysis. One can view this as a problem in outlier detection. Simply put: if a large percentage (say 99.9%) of the data elements conform to a general form, then the remaining (0.1%) data elements are likely error candidates. These data elements are considered outliers. Two things are done here; identifying outliers or strange variations in a data set and identifying trends (or normality) in data. Knowing what data is supposed to look like allows errors to be uncovered. However, the fact of the matter is that real world data is often very diverse and rarely conforms to any standard statistical distribution. This fact is readily confirmed by any practitioner and supported by our own experiences. This problem is especially acute when viewing the data in several dimensions. Therefore, more than one method for outlier detection is often necessary to capture most of the outliers. Below is a set of general methods that can be utilized for error detection.

- **Statistical**: Identify outlier fields and records using the values such as mean, standard deviation, range, based on Chebyshev's theorem (Barnett and Lewis, 1994) and considering the confidence intervals for each field (Johnson and Wichern, 1998). While this approach may generate many false positives, it is simple and fast, and can be used in conjunction with other methods.
- **Clustering**: Identify outlier records using clustering techniques based on Euclidian (or other) distance (Rokach and Maimon, 2005). Some clustering algorithms provide support for identifying outliers (Knorr *et al.*, 2000, Murtagh, 1984). The main drawback of these methods is a high computational complexity.
- **Pattern-based**: Identify outlier fields and records that do not conform to existing patterns in the data. Combined techniques (partitioning, classification, and clustering) are used to identify patterns that apply to most records (Maimon and

Rokach, 2002). A pattern is defined by a group of records that have similar characteristics or behavior for $p\%$ of the fields in the data set, where p is a user-defined value (usually above 90).

- **Association rules:** Association rules with high confidence and support define a different kind of pattern. As before, records that do not follow these rules are considered outliers. The power of association rules is that they can deal with data of different types. However, Boolean association rules do not provide enough quantitative and qualitative information. Ordinal association rules, defined by (Maletic and Marcus, 2000, Marcus *et al.*, 2001), are used to find rules that give more information (e.g., ordinal relationships between data elements). The ordinal association rules yield special types of patterns, so this method is, in general, similar to the pattern-based method. This method can be extended to find other kind of associations between groups of data elements (e.g., statistical correlations).

2.4 APPLYING DATA CLEANSING

A version of each of the above-mentioned methods was implemented. Each method was tested using a data set comprised of real world data supplied by the Naval Personnel Research, Studies, and Technology (NPRST). The data set represents part of the Navy's officer personnel information system including midshipmen and officer candidates. Similar data sets are in use at personnel records division in companies all over the world. A subset of 5,000 records with 78 fields of the same type (dates) is used to demonstrate the methods. The size and type of the data elements allows fast and multiple runs without reducing the generality of the proposed methods.

The goal of this demonstration is to prove that these methods can be successfully used to identify outliers that constitute potential errors. The implementations are designed to work on larger data sets and without extensive amounts of domain knowledge.

2.4.1 Statistical Outlier Detection

Outlier values for particular fields are identified based on automatically computed statistics. For each field, the mean and standard deviation are utilized, and based on Chebyshev's theorem (Barnett and Lewis, 1994) those records that have values in a given field outside a number of standard deviations from the mean are identified. The number of standard deviations to be considered is customizable. Confidence intervals are taken into consideration for each field. A field f_i in a record r_j is considered an outlier if the value of $f_i > \mu_i + \varepsilon\sigma_i$ or the value of $f_i < \mu_i - \varepsilon\sigma_i$, where μ_i is the mean for the field f_i, σ_i is the standard deviation, and ε is a user defined factor. Regardless of the distribution of the field f_i, most values should be within a certain number ε of standard deviations from the mean. The value of ε can be user-defined, based on some domain or data knowledge.

In the experiments, several values were used for ε (i.e., 3, 4, 5, and 6), and the value 5 was found to generate the best results (i.e., less false positives and false negatives). Among the 5,000 records of the experimental data set, 164 contain outlier values detected using this method. A visualization tool was used to analyze the results. Trying to visualize the entire data set to identify the outliers by hand would be impossible.

2.4.2 Clustering

A combined clustering method was implemented based on the group-average clustering algorithm (Yang *et al.*, 2002) by considering the Euclidean distance between records. The clustering algorithm was run several times adjusting the maximum size of the clusters. Ultimately, the goal is to identify as outliers those records previously containing outlier values. However, computational time prohibits multiple runs in an every-day business application on larger data sets. After several executions on the same data set, it turned out that the larger the threshold value for the maximum distance allowed between clusters to be merged, the better the outlier detection. A faster clustering algorithm could be utilized that allows automated tuning of the maximum cluster size as well as scalability to larger data sets. Using domain knowledge, an important subspace could be selected to guide the clustering to reduce the size of the data. The method can be used to reduce the search space for other techniques.

The test data set has a particular characteristic: many of the data elements are empty. This particularity of the data set does not make the method less general, but allowed the definition of a new similarity measure that relies on this feature. Here, strings of zeros and ones, referred to as *Hamming value* (Hamming, 1980), are associated with each record. Each string has as many elements as the number of fields in the record. The Hamming distance (Hamming, 1980) is used to cluster the records into groups of similar records. Initially, clusters having zero Hamming distance between records were identified. Using the Hamming distance for clustering would not yield relevant outliers, but rather would produce clusters of records that can be used as search spaces for other methods and also help identify missing data.

2.4.3 Pattern-based detection

Patterns are identified in the data according to the distribution of the records per each field. For each field, the records are clustered using the Euclidian distance and the k-mean algorithm (Kaufman and Rousseauw, 1990), with k=6. The six starting elements are not randomly chosen, but at equal distances from the median. A pattern is defined by a large group of records (over p% of the entire data set) that cluster the same way for most of the fields. Each cluster is classified according to the number of records it contains (i.e., cluster number 1 has the largest size and so on). The following hypothesis is considered: if there is a pattern that is applicable to most of the fields in the records, then a record following that pattern should be part of the cluster with the same rank for each field.

This method was applied on the data set and a small number of records (0.3%) were identified that followed the pattern for more than 90% of the fields. The method can be adapted and applied on clusters of records generated using the Hamming distance, rather than the entire data set. Chances of identifying a pattern will increase since records in clusters will already have certain similarity and have approximately the same empty fields. Again, real-life data proved to be highly non-uniform.

2.4.4 Association Rules

The term *association rule* was first introduced by (Aggarwal *et al.*, 1993) in the context of market-basket analysis. Association rule of this type are also referred to in the literature as *classical* or *Boolean* association rules. The concept was extended in other studies and experiments. Of particular interest to this research are the *quantitative association rules* (Srikant *et al.*, 1996) and *ratio-rules* (Korn *et al.*, 1998) that can be used for the identification of possible erroneous data items with certain modifications. In previous work we argued that another extension of the association rule – *ordinal association rules* (Maletic and Marcus, 2000, Marcus *et al.*, 2001) – is more flexible, general, and very useful for identification of errors. Since this is a recently introduced concept, it is briefly defined.

Let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of records, where each record is a set of k attributes $(a_1, \ldots, a_k)$. Each attribute a_i in a particular record r_j has a value $\phi(r_j, a_i)$ from a domain D. The value of the attribute may also be empty and is therefore included in D. The following relations (partial orderings) are defined over D, namely less or equal ($\leq$), equal (=) and, greater or equal ($\geq$) all having the standard meaning.

Then $(a_1, a_2, a_3, \ldots, a_m) \Rightarrow (a_1 \mu_1 a_2\ \mu_2 a_3\ \ \ldots\ \mu_{m-1} a_m)$, where each $\mu_i \in \{\leq, =, \geq\}$, is a an *ordinal association rule* if:

1. $a_1 \ldots a_m$ occur together (are non-empty) in at least s% of the n records, where s is the *support* of the rule;
2. and, in a subset of the records $R' \subseteq R$ where $a_1 \ldots a_m$ occur together and $\phi(r_j, a_1)\ \mu_1 \ldots \mu_{m-1}\ \phi(r_j, a_m)$ is true for each $r_j \in R'$. Thus $|R'|$ is the number of records that the rule holds for and the *confidence*, c, of the rule is the percentage of records that hold for the rule $c = |R'|/|R|$.

The process to identify potential errors in data sets using ordinal association rules is composed of the following steps:

1. Find ordinal rules with a minimum confidence c. This is done with a variation of *apriori* algorithm (Aggarwal *et al.*, 1993).
2. Identify data items that broke the rules and can be considered outliers (potential errors).

Here, the manner in which support of a rule is important differs from typical data-mining problem. We assume all the discovered rules that hold for more than two records represent valid possible partial orderings. Future work will investigate user-specified minimum support and rules involving multiple attributes.

The method first normalizes the data (if necessary) and then computes comparisons between each pair of attributes for every record. Only one scan of the data set is required. An array with the results of the comparisons is maintained in the memory. Figure 2.1 contains the algorithm for this step. The complexity of this step is only

$O(N * M^2)$ where N is the number of records in the data set, and M is the number of fields/attributes. Usually M is much smaller than N. The results of this algorithm are written to a temporary file for use in the next step of processing.

In the second step, the ordinal rules are identified based on the chosen minimum confidence. There are several researched methods to determine the strength including interestingness and statistical significance of a rule (e.g., minimum support and minimum confidence, chi-square test, etc.). Using confidence intervals to determine the minimum confidence is currently under investigation. However, previous work on the data set (Maletic and Marcus, 2000) used in our experiment showed that the distribution of the data was not normal. Therefore, the minimum confidence was chosen empirically, several values were considered and the algorithm was executed. The results indicated that a minimum confidence between 98.8 and 99.7 provide best results (less number of false negative and false positives).

```
Algorithm compare items.
    for each record in the data base (1...N)
        normalize or convert data
        for each attribute x in (1...M-1)
            for each attribute y in (x+1...M-1)
                compare the values in x and y
                update the comparisons array
            end for.
        end for.
        output the record with normalized data
    end for.
    output the comparisons array
end algorithm.
```

Fig. 2.1. The algorithm for the first step

The second component extracts the data associated with the rules from the temporary file and stores it in memory. This is done with a single scan (complexity $O(C(M,2))$. Then for each record in the data set, each pair of attributes that correspond to a pattern it is checked to see if the values in those fields are within the relationship indicated by the pattern. If they are not, each field is marked as possible error. Of course, in most cases only one of the two values will actually be an error. Once every pair of fields that correspond to a rule is analyzed, the average number of possible error marks for each marked field is computed. Only those fields that are marked as possible errors more times than the average are finally marked as having likely errors. Again, the average value was empirically chosen as a threshold to prune the possible errors set. Other methods to find such a threshold, without using domain knowledge or multiple experiments, are under investigation. The time complexity of this step is *O(N*C(M,2))*, and the analysis of each record is done entirely in the main memory. Figure 2.2 shows the algorithm used in the implementation of the second component. The results identify which records and fields are likely to have errors.

```
Algorithm analyze records.
    for each record in the data base (1N)
        for each rule in the pattern array
            determine rule type and pairs
            compare item pairs
            if pattern NOT holds
                then mark each item as possible
                     error
        end for.
        compute average number of marks
        select the high probability marked
               errors
    end for.
end algorithm.
```

Fig. 2.2. Algorithm for the second step

Using a 98% confidence, 9,064 records in 971 fields that had high probability errors were identified out of the extended data set of 30,000 records. These were compared with those outliers identified with statistical methods. These possible errors not only matched most of the previously discovered ones, but 173 were errors unidentified by the previous methods. The distribution of the data influenced dramatically the error identification of the data process in the previous utilized methods. This new method is proving to be more robust and is influenced less by the distribution of the data. Table 2.2 shows an error identified by ordinal association rules and missed with the previous methods. Here two patterns were identified with confidence higher than 98%: values in field 4 $\leq$ values in field 14, and values in field 4 $\leq$ values in field 15. In the record no. 199, both fields 14 and 15 were marked as high probability errors. Both values are in fact minimum values for their respective fields. The value in field 15 was identified previously as outlier but the value in field 14 was not because of the high value of the standard deviation for that field. It is obvious, even without consulting a domain expert, that both values are in fact wrong. The correct values (identified later) are 800704. Other values that did not lie at the edge of the distributions were identified as errors as well.

Table 2.2. A part of the data set. An error was identified in record 199, field 14, which was not identified previously. The data elements are dates in the format YYMMDD.

Record Number	Field 1	...	Field 4	...	Field 14	Field 15	...
199	600603	...	780709	...	**700804**	700804	...

2.5 CONCLUSIONS

Data cleansing is a very young field of research. This chapter presents some of the current research and practice in data cleansing. One missing aspect in the research is the definition of a solid theoretical foundation that would support many of the existing approaches used in an industrial setting. The philosophy promoted here is that a data cleansing framework must incorporate a variety of such methods to be used in conjunction. Each method can be used to identify a particular type of error in data. While not specifically addressed here, taxonomies like the one proposed in (Kim *et al.*, 2003) should be encouraged and extended by the research community. This will support the definition and construction of more general data cleansing frameworks.

Unfortunately, little basic research within the information systems and computer science communities has been conducted that directly relates to error detection and data cleansing. In-depth comparisons of data cleansing techniques and methods have not yet been published. Typically, much of the real data cleansing work is done in a customized, in-house, manner. This behind-the-scenes process often results in the use of undocumented and ad hoc methods. Data cleansing is still viewed by many as a “black art” being done “in the basement”. Some concerted effort by the database and information systems groups is needed to address this problem.

Future research directions include the investigation and integration of various methods to address error detection. Combination of knowledge-based techniques with more general approaches should be pursued. In addition, a better integration of data cleansing in the data quality processes and frameworks should be achieved. The ultimate goal of data cleansing research is to devise a set of general operators and theory (much like relational algebra) that can be combined in well-formed statements to address data cleansing problems. This formal basis is necessary to design and construct high quality and useful software tools to support the data cleansing process.

References

Aggarwal, C. C. & Yu, P. S. Outlier detection for high dimensional data. Proceedings of ACM SIGMOD international Conference on Management of Data; 2001 May 21-24; Santa Barbara, CA. 37-46.

Agrawal, R., Imielinski, T., & Swami, A. Mining Association rules between Sets of Items in Large Databases. Proceedings of ACM SIGMOD International Conference on Management of Data; 1993 May; Washington D.C. 207-216.

Ballou, D. P. & Tayi, G. K. Enhancing Data Quality in Data Warehouse Environments, Communications of the ACM 1999; 42(1):73-78.

Barnett, V. & Lewis, T., Outliers in Statistical Data. John Wiley and Sons, 1994.

Bochicchio, M. A. & Longo, A. Data Cleansing for Fiscal Services: The Taviano Project. Proceedings of 5th International Conference on Enterprise Information Systems; 2003 April 22-26; Angers, France. 464-467.

Brachman, R. J., Anand, T., The Process of Knowledge Discovery in Databases — A Human–Centered Approach. In Advances in Knowledge Discovery and Data Mining, Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uth-urasamy, R., eds. MIT Press/AAAI Press, 1996.

Cadot, M. & di Martion, J. A data cleaning solution by Perl scripts for the KDD Cup 2003 task 2, ACM SIGKDD Explorations Newsletter 2003; 5(2):158-159.

Chaudhuri, S., Ganjam, K., Ganti, V., & Motwani, R. Robust and efficient fuzzy match for online data cleaning. Proceedings of ACM SIGMOD International Conference on Management of Data; 2003 june 9-12; San Diego, CA. 313-324.

Dasu, T., Vesonder, G. T., & Wright, J. R. Data quality through knowledge engineering. Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2003 August 24-27; Washington, D.C. 705-710.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P., From Data Mining to Knowledge Discovery: An Overview. In Advances in Knowledge Discovery and Data Mining, Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurasamy, R., eds. MIT Press/AAAI Press, 1996.

Fayyad, U. M., Piatetsky-Shapiro, G., & Uthurasamy, R. Summary from the KDD-03 Panel - Data Mining: The Next 10 Years, ACM SIGKDD Explorations Newsletter 2003; 5(2):191-196.

Feekin, A. & Chen, Z. Duplicate detection using k-way sorting method. Proceedings of ACM Symposium on Applied Computing; 2000 Como, Italy. 323-327.

Fox, C., Levitin, A., & Redman, T. The Notion of Data and Its Quality Dimensions, Information Processing and Management 1994; 30(1):9-19.

Galhardas, H. Data Cleaning: Model, Language and Algoritmes. University of Versailles, Saint-Quentin-En-Yvelines, Ph.D., 2001.

Guyon, I., Matic, N., & Vapnik, V., Discovering Information Patterns and Data Cleaning. In Advances in Knowledge Discovery and Data Mining, Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurasamy, R., eds. MIT Press/AAAI Press, 1996.

Hamming, R. W., Coding and Information Theory. New Jersey, Prentice-Hall, 1980.

Hawkins, S., He, H., Williams, G. J., & Baxter, R. A. Outlier Detection Using Replicator Neural Networks. Proceedings of 4th International Conference on Data Warehousing and Knowledge Discovery; 2002 September 04-06; 170-180.

Hernandez, M. & Stolfo, S. Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem, Data Mining and Knowledge Discovery 1998; 2(1):9-37.

Johnson, R. A. & Wichern, D. W., Applied Multivariate Statistical Analysis. Prentice Hall, 1998.

Kaufman, L. & Rousseauw, P. J., Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990.

Kim, W., Choi, B.-J., Hong, E.-K., Kim, S.-K., & Lee, D. A taxonomy of dirty data, Data Mining and Knowledge Discovery 2003; 7(1):81-99.

Kimball, R. Dealing with Dirty Data, DBMS 1996; 9(10):55-60.

Knorr, E. M. & Ng, R. T. Algorithms for Mining Distance-Based Outliers in Large Datasets. Proceedings of 24th International Conference on Very Large Data Bases; 1998 New York. 392-403.

Knorr, E. M., Ng, R. T., & Tucakov, V. Distance-based outliers: algorithms and applications, The International Journal on Very Large Data Bases 2000; 8(3-4):237-253.

Korn, F., Labrinidis, A., Yannis, K., & Faloustsos, C. Ratio Rules: A New Paradigm for Fast, Quantifiable Data Mining. Proceedings of 24th VLDB Conference; 1998 New York. 582–593.

Lee, M. L., Ling, T. W., & Low, W. L. IntelliClean: a knowledge-based intelligent data cleaner. Proceedings of Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2000 August 20-23; Boston, MA. 290-294.

Levitin, A. & Redman, T. A Model of the Data (Life) Cycles with Application to Quality, Information and Software Technology 1995; 35(4):217-223.

Li, Z., Sung, S. Y., Peng, S., & Ling, T. W. A New Efficient Data cleansing Method. Proceedings of Database and Expert Systems Applications (DEXA 2002); 2002 September 2-6; Aix-en-Provence, France. 484-493.

Maimon, O. and Rokach, L. Improving supervised learning by feature decomposition, Proceedings of the Second International Symposium on Foundations of Information and Knowledge Systems, Lecture Notes in Computer Science, Springer, 2002, 178-196

Maletic, J. I. & Marcus, A. Data Cleansing: Beynod Integrity Analysis. Proceedings of The Conference on Information Quality (IQ2000); 2000 October 20-22; Massachusetts Institute of Technology. 200-209.

Marcus, A., Maletic, J. I., & Lin, K. I. Ordinal Association Rules for Error Identification in Data Sets. Proceedings of Tenth International Conference on Information and Knowledge Management (CIKM 2001); 2001 November 3-5; Atlanta, GA. to appear.

Murtagh, F. A Survey of Recent Advances in Hierarchical Clustering Algorithms, The Computer Journal 1983; 26(4):354-359.

Orr, K. Data Quality and Systems Theory, Communications of the ACM 1998; 41(2):66-71.

Raman, V. & Hellerstein, J. M. Potter's wheel an interactive data cleaning system. Proceedings of 27th International Conference on Very Large Databases 2001 September 11-14; Rome, Italy. 381–391.

Ramaswamy, S., Rastogi, R., & Shim, K. Efficient Algorithms for Mining Outliers from Large Data Sets. Proceedings of ACM SIGMOD International Conference on Management of Data; 2000 Dallas. 427-438.

Redman, T. The Impact of Poor Data Quality on the Typical Enterprise, Communications of the ACM 1998; 41(2):79-82.

Rokach, L., Maimon, O. (2005), Clustering Methods, Data Mining and Knowledge Discovery Handbook, Springer, pp. 321-352.

Simoudis, E., Livezey, B., & Kerber, R., Using Recon for Data Cleaning. In Advances in Knowledge Discovery and Data Mining, Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurasamy, R., eds. MIT Press/AAAI Press, 1995.

Srikant, R., Vu, Q., & Agrawal, R. Mining Association Rules with Item Constraints. Proceedings of SIGMOD International Conference on Management of Data; 1996 June; Montreal, Canada. 1-12.

Strong, D., Yang, L., & Wang, R. Data Quality in Context, Communications of the ACM 1997; 40(5):103-110.

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; 2002 November 04-09; McLean, VA. 76-83.

Svanks, M. Integrity Analysis: Methods for Automating Data Quality Assurance, EDP Auditors Foundation 1984; 30(10):595-605.

Wang, R., Storey, V., & Firth, C. A Framework for Analysis of Data Quality Research, IEEE Transactions on Knowledge and Data Engineering 1995; 7(4):623-639.

Wang, R., Strong, D., & Guarascio, L. Beyond Accuracy: What Data Quality Means to Data Consumers, Journal of Management Information Systems 1996; 12(4):5-34.

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

Yang, Y., Carbonell, J., Brown, R., Pierce, T., Archibald, B. T., & Liu, X. Learning Approaches for Detecting and Tracking News Events, IEEE Intelligent Systems 1999; 14(4).

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

Zhao, L., Yuan, S. S., Peng, S., & Ling, T. W. A new efficient data cleansing method. Proceedings of 13th International Conference on Database and Expert Systems Applications; 2002 September 02-06; 484-493.