Data Cleansing: Beyond Integrity Analysis[1]

Jonathan I. Maletic
Andrian Marcus

Division of Computer Science
The Department of Mathematical Sciences
The University of Memphis
Campus Box 526429
Memphis, TN 38152

jmaletic@memphis.edu, amarcus@memphis.edu

Abstract: *The paper analyzes the problem of data cleansing and automatically identifying potential errors in data sets. An overview of the diminutive amount of existing literature concerning data cleansing is given. Methods for error detection that go beyond integrity analysis are reviewed and presented. The applicable methods include: statistical outlier detection, pattern matching, clustering, and data mining techniques. Some brief results supporting the use of such methods are given. The future research directions necessary to address the data cleansing problem are discussed.*

Keywords: data cleansing, data cleaning, data quality, error detection.

1. Introduction

The quality of a large real world data set depends on a number of issues [9, 39, 40], 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 given 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. Unless an organization takes extreme measures in an effort to avoid data errors the field errors rates are typically around 5% or more [28, 31].

For existing data sets, the logical solution to this problem 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 [30]. A manual process of data cleansing is also laborious, time consuming, and itself prone to errors. The need for 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 an existing data set.

While this may seem to be an obvious solution, very little basic research has been directly aimed at methods to support such tools. Some related research addresses the issues of data quality [2, 30, 31, 39] and tools to assist in "by hand" data cleansing and/or relational data integrity analysis

[1] This research was supported in part by a grant from the Office of Naval Research (N00014-99-1-0730).

(e.g., [6, 8, 10, 29, 36-38]). In this paper, the differing views of data cleansing are surveyed and reviewed. A general framework of the data cleansing process is presented and a set of general methods that can be used to address the problem is presented. The experimental results of applying these methods to a real world data set are also given. Finally, future research directions necessary to address the data cleansing problem are discussed.

2. An Overview of Data Cleansing

Data cleansing is a relatively new research field. The process is computationally expensive on very large data sets and thus it was almost impossible to do with old technology. The new faster computers allow performing the data cleansing process in acceptable time on large amounts of data. There are many issues in the data cleansing area that researchers are attempting to tackle. They consist of dealing with missing data, determining record usability, erroneous data, etc. Different approaches address different issues. Of particular interest here, is the search context for what is called in literature and the business world as “dirty data” [9, 12, 13, 18, 21]. There is no commonly agreed definition of the 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 (also termed data mining), and data/information quality management (e.g., TDQM).

Within the data warehousing field, data cleansing is applied especially when several databases are merged. Records referring to the same entity are represented in different formats in the different data sets or are represented erroneously. 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* [15, 18, 26]. Instances of this problem appearing in literature are called record linkage, semantic integration, instance identification, or object identity problem.

From this perspective data cleansing is defined in several (but similar) ways. In [15] data cleansing is the process of eliminating the errors and the inconsistencies in data, and solving the object identity problem. The [18] paper defines the data cleansing problem as the merge/purge problem and proposes the basic sorted-neighborhood method to solve it. The proposed method is the basis for the DataBlade module of the DataCleanser tool [8].

Data cleansing is much more than simply updating a record with good data. Serious data cleansing involves decomposing and reassembling the data. According to [21] 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 the current technology for data cleansing are heavily focused on customer lists [21]. In this area, three companies dominate the data cleansing marketplace [21], and all three specialize in cleaning large customer address lists. The three companies are Harte-Hanks Data Technologies [36], Innovative Systems Inc., and Vality Technology [37]. Recently, companies have started to produce tools and offer data cleaning services that do not address specifically the customer address lists but instead rely on domain specific information provided by the customer: Centrus Merge/Purge Module [6], DataCleanser [8], etc. A very good description and design of a framework for assisted data cleansing within the merge/purge problem is available in [15].

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 (e.g., [13, 14, 24, 28-31, 34, 35, 40]). Other work refers to the same problem as the enterprise data quality management [12]. The most comprehensive survey of the research in this area is available in [39].

Unfortunately, none of the mentioned papers refer explicitly to the data cleansing problem. Some of the papers deal strictly with the process management issues from data quality perspective, others with definition of data quality. The later category is of interest to this research. In the proposed model of data life cycles with application to quality [24] 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 [13] 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 the framework would be the process that assesses the correctness of data and improve its quality.

More recently, data cleansing is regarded as a first step, or a preprocessing step, in the KDD process [5, 11]. Though 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 [16] discovering of informative patterns is 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 [32] data cleansing is defined as the process that implements computerized methods of examining databases, detecting missing and incorrect data, and correcting errors. The Recon Data Mining system is used to assist the human expert to identify a series of error types in financial data systems.

3. General Methods for Data Cleansing

Certainly, with all above in mind, data cleansing must be viewed as a process. This process may be tied directly to data acquisition and definition or it may be 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. 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.

Many of the aforementioned data cleansing tools utilize integrity analysis to locate data errors. Given a data set (data base) that adheres to the relational model, the data integrity analysis [7] can be used as a simple data cleansing operation. Relational data integrity, including entity, referential, and column integrity, can be accomplished using relational data base queries (e.g., SQL). Many database systems (e.g., Oracle, MS Access) support this type of data cleansing to some degree. The database administrator should do this type of data cleansing, unfortunately the administrator often lacks the domain knowledge and/or clearly defined responsibility to correctly carry out this task. There are tools, usable by non-database experts, which support this type of cleansing. For example, Wang [38] has developed a tool that supports data integrity analysis within the framework of TDQM.

While data integrity analysis can uncover a number of possible errors in a data set, it does not address errors that are more complex. Errors that involve 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. Put simply: 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. But, the fact of the matter is that real world data often is very diverse and rarely conforms to any standard statistical distribution. This 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**: Identifying outlier fields and records using the values of mean, standard deviation, range, etc., based on Chebyshev's theorem [3, 4], considering the confidence intervals for each field [19].
- **Clustering**: Identify outlier records using clustering based on Euclidian (or other) distance. Existing clustering algorithms provide little support for identifying outliers [22, 27, 42]. However, in some cases clustering the entire record space can reveal outliers that are not identified at the field level inspection [19]. The main drawback of this method is computational time. The clustering algorithms have high computational complexity. For large record spaces and large number of records, the run time of the clustering algorithms is prohibitive.
- **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. A pattern is defined by a group of records that have similar characteristics ("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 were defined by the authors [25] and used to find rules that gave more information (e.g., ordinal relationships between data elements). The ordinal association rules yield a special type of patterns, so this method is, in general, similar with the pattern-based method. This method can be extended to find other kind of associations between groups of data elements (e.g., statistical correlations).

4. Experiments

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 5000 records with 78 fields of the same type (dates) from this data set is used to demonstrate the methods. The size of the data and the type of the data elements allowed fast and multiple runs without reducing much the generality of the proposed methods.

The goal of the experiments is to prove that these methods can be successfully used to identify outliers that constitute potential errors. The implementations were designed to work on larger data sets and without large amounts of domain knowledge. The only information needed from the user is the size of the data set and the values of some threshold parameters.

4.1. Statistical Outlier Detection

Outlier values for particular fields are identified based on automatically computed statistics. For each field the average and the standard deviation are utilized and based on Chebyshev's theorem [3] 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. Each field f_i can be considered as a variable with as many realizations as the number of records in which it has a value. 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, or theoretically using Chebyshev's theorem.

In the experiments, several values were used for ε (3, 4, 5, and 6), and the value 5 was found to generate the best results (less false positives and false negatives). Among the 5000 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.

4.2. Clustering

A combined clustering method was implemented based on the group-average clustering algorithm [41] considering the Euclidean distance between records. The clustering algorithm was run several times adjusting the maximum size of the clusters. Ultimately, the goal was to identify as outliers at least those records that were identified before as 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 that are to be merged together the better the outlier detection. A faster clustering algorithm could be utilized that may allow automated tuning of the maximum cluster size, as well as scalability to larger data sets. Also, using some domain knowledge, an "important" subspace can 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 identified clusters combine together records that bear a certain similarity. Therefore, it is highly probable that the records in a cluster would follow a certain pattern.

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* [17], are associated with each record. Each string has as many elements as the number of fields. A “1” in the string represents a non-empty field on the same position in the record as the 1 in the string. A “0” in the string represents an empty field on the same position in the record as the 0 in the string. The Hamming distance [17] is utilized to cluster the records into groups of similar records. Initially, clusters having zero Hamming distance between records were identified. Unfortunately, the number of identified clusters was too high (4631 clusters for 5000 records). The largest cluster had only 98 records and the second largest only 29. Since the results were not encouraging, a hierarchical clustering method is considered for implementation to determine clusters of records with a diameter larger than zero. 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 the following methods.

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 [20], with k=6. The six starting elements are not randomly chosen, but at equal distances from the mean, one of them being an empty field. 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). Then 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% of total number of records) were identified that followed the pattern for more than 90% of the fields. The tool will be adapted and applied on clusters of records generated using the Hamming distance, rather than on 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 fields empty. Again, real-life data proved to be highly non-uniform.

4.4. Association Rules

The term *association rule* was first introduced by Agrawal et. al. [1] 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 interest to this research are in particular the *quantitative association rules* [33] and *ratio-rules* [23] that can be used for the identification of possible erroneous data items with certain modifications. In a previous work was argued that another extension of the association rule – *ordinal association rules* [25] – is more useful for identification of outliers and errors. Since this is a recently introduced concept, it is briefly defined.

Let $B = \{b_1, b_2, \ldots, b_n\}$ a *data set*, where each *record* $b_i \subseteq I$ is a collection of *items*, and $I = \{i_1, i_2, \ldots, i_k\}$ is a set of k items. Each item i_i has the same numerical *domain* D ($i_i \in D$) and the following relationships are defined in D: ≤ - less or equal, = - equal, ≥ - greater or equal. Then $i_1, i_2 \Rightarrow i_1\ op\ i_2$, where $op \in \{\leq, =, \geq\}$, is a an *ordinal association rule* if:

1. i_1 and i_2 occur together in at least *s%* of the n records, where *s* is the *support* of the rule;
2. and, in *c%* of the records where i_1 and i_2 occur together $i_1\ op\ i_2$ is true, and where *c* is the *confidence* of the rule.

This definition extends easily to $J \Rightarrow j_1\ op\ j_2\ op \ldots op\ j_m$, $op \in \{\leq, =, \geq\}$, $m \in \{1, \ldots, k\}$, where $J = \{j_1, j_2, \ldots, j_m\}$ is a set of *m* items, $J \subseteq I$.

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 [1].
2. Identify data items that broke the rules and can be considered outliers (potential errors).

At this point, there is no need to consider a minimum acceptable support. Future work will investigate user-specified minimum support. However, this will only change the number of initially identified patterns. Since only pairs of items are considered, there can be at most C(M,2) patterns. Where M is the number of attributes (fields) of the data set.

Record no.	Field 1	…	Field 4	…	Field 14	Field 15	…
199	600603	…	780709	…	700804	700804	…

Table 1: 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.

Using a 98% confidence, 971 fields that have high probability errors were identified out of 5000 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 not influenced as much by the distribution of the data and is proving to be more robust.

Table 1 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 ≤ values in field 14, and values in field 4 ≤ 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 one in field 15 was identified previously as outlier, but the one 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.

5. Conclusions

A set of methods was presented which addresses the problem of automatic identification of errors in data sets. The methods were implemented and the results showed that some of the methods could be successfully applied to real-world data, while others need fine-tuned and improvement. Each of the proposed methods has strength and weakness.

Unfortunately, little basic research within the information systems and computer science communities has been conducted which is directly related to error detection and data cleansing. Few in-depth comparisons of data cleansing techniques and methods have been published. Typically, much of the real data cleansing work is done in a very customized, in house, manner. This behind the scenes process often results in the use of undocumented and ad hoc methods. Some concerted effort by the database and information systems committees is needed to address this problem.

Future research, by the authors, will investigate integration of various methods to address error detection. Also, knowledge-based techniques can be utilized for detection and correction in many situations. The best solution for automatic error detection will be an integrated approach, which utilizes a number of methods. Methods that are based on the analysis of groups of correlated fields (e.g., based on statistical correlation) should also prove powerful.

The ultimate goal of this 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.

6. References

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

[2] Ballou, D. and Tayi, K., "Methodology for Allocating Resources for Data Quality Enhancement," *CACM*, vol. 32, no. 3, 1989, pp. 320-329.

[3] Barnett, V. and Lewis, T., *Outliers in Statistical Data*, John Wiley and Sons, 1994.

[4] Bock, R. K. and Krischer, W., *The Data Analysis Briefbook*, Springer, 1998.

[5] Brachman, R. J. and 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., and Uthurasamy, R., Eds., MIT Press/AAAI Press, 1996, pp. 97-58.

[6] Centrus, "Qualitative Marketing Software Centrus Merge/Purge Module," Centrus, Webpage, Date Accessed: 01/15/2000, http://www.qmsoft.com/solutions/Merge.htm, 2000.

[7] Date, C. J., *An Introduction to Database Systems*, Addison Wesley, 1990.

[8] EDD, "Home page of DataCleanser tool," EDD, Webpage, Date Accessed: 01/15/2000, http://www.npsa.com/edd/, 2000.

[9] English, J., "Plain English on Data Quality," DM Review, Webzine, Date Accessed: 02/10/99, http://www.dmreview.com, 1999.

[10] ETI, "ETI-Data Cleanse tool," E.T. International, Webpage, Date Accessed: 01/15/2000, http://www.evtech.com/products/dc2.html, 2000.

[11] Fayyad, U., Piatetsky-Shapiro, G., and 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., and Uthurasamy, R., Eds., MIT Press/AAAI Press, 1996, pp. 1-36.

[12] Flanagan, T. and Safdie, E., "A Practical Guide to Achieving Enterprise Data Quality," Webpage, Date Accessed: 12/01/99, http://www.techguide.com/, 1999.

[13] Fox, C., Levitin, A., and Redman, T., "The notion of Data and Its Quality Dimensions," *Information Processing and Management*, vol. 30, no. 1, 1994, pp. 9-19.

[14] Fox, C., Levitin, A., and Redman, T., "Data and Data Quality," in *Encyclopedia of Library and Information Science*,, 1995.

[15] Galhardas, H., Florescu, D., Shasha, D., and Simon, E., "An Extensible Framework for Data Cleaning," Institute National de Recherche en Informatique ét en Automatique, Technical Report 1999.

[16] Guyon, I., Matic, N., and Vapnik, V., "Discovering Information Patterns and Data Cleaning," in *Advances in Knowledge Discovery and Data Mining*, Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurasamy, R., Eds., MIT Press/AAAI Press, 1996, pp. 181-203.

[17] Hamming, R. W., *Coding and Information Theory*, New Jersey, Prentice-Hall, 1980.

[18] Hernandez, M. A. and Stolfo, J. S., "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem," *Journal of Data Mining and Knowledge Discovery*, vol. 2, 1998, pp. 9-37.

[19] Johnson, R. A. and Wichern, D. W., *Applied Multivariate Statistical Analysis*, 4th ed., Prentice Hall, 1998.

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

[21] Kimball, R., "Dealing with Dirty Data," *DBMS*, vol. 9, no. 10, September 1996, pp. 55.

[22] Knorr, E. M. and Ng, R. T., "A Unified Notion of Outliers: Properties and Computation," in Proceedings of KDD 97, 1997, pp. 219-222.

[23] Korn, F., Labrinidis, A., Yannis, K., and Faloustsos, C., "Ratio Rules: A New Paradigm for Fast, Quantifiable Data Mining," in Proceedings of 24th VLDB Conference, New York, 1998, pp. 582--593.

[24] Levitin, A. and Redman, T., "A Model of the data (life) cycles with application to quality," *Information and Software Technology*, vol. 35, no. 4, 1995, pp. 217-223.

[25] Marcus, A. and Maletic, J. I., "Utilizing Association Rules for the Identification of Errors in Data," The University of Memphis, Division of Computer Science, Memphis, Technical Report TR-14-2000, May 2000.

[26] Moss, L., "Data Cleansing: A Dichotomy of Data Warehousing," *DM Review* February 1998.

[27] Murtagh, F., "A Survey of Recent Advances in Hierarchical Clustering Algorithms," *The Computer Journal*, vol. 26, no. 4, 1983, pp. 354-359.

[28] Orr, K., "Data Quality and Systems Theory," *CACM*, vol. 41, no. 2, February 1998, pp. 66-71.

[29] Pak, S. and Pando, A., "Data Quality Analyzer: A Software Tool for Analyzing Data Quality in Data Manufacturing Systems," Webpage, Date Accessed: 02/10/99, http://web.mit.edu/tdqm/papers/93/pass1/93-10.html, 1993.

[30] Redman, T., *Data Quality for the Information Age*, Artech House, 1996.

[31] Redman, T., "The Impact of Poor Data Quality on the Typical Enterprise," *CACM*, vol. 41, no. 2, February 1998, pp. 79-82.

[32] Simoudis, E., Livezey, B., and Kerber, R., "Using Recon for Data Cleaning," in Proceedings of KDD, 1995, pp. 282-287.

[33] Srikant, R., Vu, Q., and Agrawal, R., "Mining Association Rules with Item Constraints," in Proceedings of SIGMOD International Conference on Management of Data, Montreal, Canada, June 1996, pp. 1-12.

[34] Strong, D., Yang, L., and Wang, R., "Data Quality in Context," *CACM*, vol. 40, no. 5, May 1997, pp. 103-110.

[35] Svanks, M., "Integrity Analysis: Methods for Automating Data Quality Assurance," *EDP Auditors Foundation*, vol. 30, no. 10, 1984, pp. 595-605.

[36] Trillium, "Trillium Software System for data warehousing and ERP," Trillium Software, Webpage, Date Accessed: 01/15/2000, http://www.trilliumsoft.com/products.htm, 2000.

[37] Vality, "INTEGRITY Data Re-engineering Environment," Vality Technology, Webpage, Date Accessed: 01/15/2000, http://www.vality.com/, 2000.

[38] Wang, R., Reddy, M., P., and Gupta, A., "An Object-Oriented Implementation of Quality Data Products," in Proceedings of WITS, 1993

[39] Wang, R., Storey, V., and Firth, C., "A Framework for Analysis of Data Quality Research," *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 4, August 1995, pp. 623-639.

[40] Wang, R., Strong, D., and Guarascio, L., "Beyond Accuracy: What Data Quality Means to Data Consumers," *Journal of Management Information Systems*, vol. 12, no. 4, Spring 1996, pp. 5-34.

[41] Yang, Y., Carbonell, J., Brown, R., Pierce, T., Archibald, B. T., and Liu, X., "Learning approaches for Detecting and Tracking News Events," *IEEE Intelligent Systems*, vol. 14, no. 4, July/August 1999.

[42] Zhang, T., Ramakrishnan, R., and Livny, M., "BIRCH: A New Data Clustering Algorithm and Its Applications," *Data Mining and Knowledge Discovery*, vol. 1, 1997, pp. 141-182.