Mining Domain Knowledge: Using Functional Dependencies to Profile Data

Research-in-Progress

Derek Legenzoff
The University of Alabama
Tuscaloosa, AL, USA
dtlegenzoff@crimson.ua.edu

Teagen Nabity
The University of Alabama
Tuscaloosa, AL, USA
tmnabity@crimson.ua.edu

Abstract

Poor data quality is one of the primary issues facing big data projects. Cleaning data and improving quality can be expensive and time-intensive. In data warehouse projects, data cleaning is estimated to account for 30% to 80% of the project's development time and budget. Data quality mining is one method used to identify errors that has become increasingly popular in the past 20 years. Our research-in-progress aims to identify multi-field errors via the mining of functional dependencies. Existing research on data quality mining and functional dependencies has focused on improving algorithms to identify a higher percentage of complex errors. The proposed process strives to introduce an efficient method for expediting error identification and increasing a user's domain knowledge in order to reduce the costs associated with cleaning; the process will also include an assessment of when further cleaning is unlikely to be cost effective.

Keywords: Data mining, data quality, data cleaning, functional dependency, data cleaning process, error identification, domain knowledge

Introduction

The exponential growth of data over the past few decades has had far reaching effects. In 2010, the White House announced big data – in addition to healthcare and national security – was a national challenge (Kaisler et al., 2012). One of the greatest issues facing big data is poor data quality. A report from the Data Warehousing Institute estimated poor data quality costs business more than $600 billion in the United States alone (Eckerson, 2002). Additionally, Gartner (2007) estimates that 25% of critical data in Fortune 1000 organizations is flawed. In an age when competitive advantage is gained through insights from data, poor data quality threatens the success of an organization. Most datasets will require some level of cleaning to allow for effective analysis.

Improving data quality is a non-trivial issue because data cleaning is both costly and time-consuming; data cleaning is a labor-intensive process estimated to account for 30% to 80% of development time and budget in data warehouse projects (Bohannon et al., 2007; Fan et al., 2008). Poor data quality can be a result of manual data entry, data migrations, changes to source systems, etc. As a result, errors can vary in both complexity and cost. Complex errors, if undetected, can be detrimental to analysis. Removing such errors can be expensive and requires significant domain knowledge and expertise. The growth of data, coupled with increased employee turnover, has created an environment where significant domain knowledge is often unavailable. Furthermore, data sources often lack adequate documentation (Sapia et al., 1999), compounding the issue.

Because of this, data quality mining has become an increasingly popular and necessary method for profiling data and identifying errors in data quality projects over the past two decades. These algorithms can be used to discover the semantics of the data source, in addition to finding complex errors (Sapia et al., 1999), making them particularly useful when data is foreign to those working with it. Mining functional dependencies – a prominent technique within data quality mining – permits the exploration of informal

relationships in the dataset and thus provides insights into the dataset's domain. While identifying errors is the direct focus of our research, increased domain knowledge is a valuable byproduct as it is important to the success of any data quality project.

Most research in the field of data quality mining has focused on improving algorithms to identify a higher percentage of complex errors. Conversely, the goal of our research is to optimize the process of complex error identification. With 88% of data warehousing projects failing or going over budget (Marsh, 2005), in large part because of poor data quality (Bohannon et al., 2007; Fan et al., 2008), it is essential to minimize costs while maximizing error detection. The aim of this research is ***to design a process for timely and effective identification of multi-field errors through the mining of functional dependencies. This will expedite the error identification process, lead to the mining of domain knowledge, and contribute to successful data quality projects.***

Background

Data Quality

Extensive research has been conducted on what constitutes data quality, in part due to the widespread problem of poor data quality. It is estimated that between 0.5% and 30% of a company's field level data is erroneous (Redman, 1998). More than 50% of companies do not claim to be very confident in the quality of their data and organizations tend to overestimate the quality of their data (Eckerson, 2002; Marsh, 2005). Although estimating costs accrued from poor data quality is difficult, 75% of organizations have identified costs stemming from 'dirty data' (Marsh, 2005); three proprietary studies estimate the costs range from 8% to 12% of revenue (Redman, 1998) and an authority on data quality issues, Larry English, estimates this range to be higher – between 10% and 25% of revenue (Eckerson, 2002).

Despite several variations of the attributes contributing to data quality, many researchers appear to agree on the following set (Eckerson, 2002; IBM, 2012; Marsh, 2005; Michnik and Lo, 2009; Pipino et al., 2002; Singh and Singh, 2010):

- Accuracy – the degree to which the data objects represents real-world objects.
- Consistency – the degree to which distinct occurrences of the data are in agreeance and presented in the same format.
- Completeness – the degree to which the data is present for each record.
- Validity – the degree to which the data is correct and reasonable--such as determining if the values fall within the possible or acceptable range as defined by the business.
- Integrity – the degree to which data important to relationship linkages is present.
- Timeliness – the degree to which the data is available when needed.
- Accessibility – the degree to which the data is understandable, usable, and obtainable; this attribute also considers the security of the data and moderating who can and cannot access parts or any of the data.

It is clear from the research poor data quality is detrimental to organizations' decision making, financial resources, human resources, customer relations, operations, and reputation (Eckerson, 2002; IBM, 2012; Marsh, 2005; Redman, 1998); however, error-free data should not be the goal as it would be cost-prohibitive and unmaintainable (Eckerson, 2002; Lucas et al., 2014; Haug et al., 2011).

Definition of Errors

Data errors differ in terms of complexity. One of the simplest classifications of data errors is random vs systematic.

Random Errors – data points that depart from expected values due to uncontrolled variation in collection or data input. Random errors are less likely to have a significant negative impact on data analysis as they tend to cancel each other (BusinessDictionary, 2016). Some random errors may be simple to identify with outlier analysis while others may be more time-intensive and require domain knowledge.

Systematic Errors – a group of data points that depart from expected values due to a flaw in the system for collecting or inputting data. These errors are more likely to have a significant negative impact on data analysis as the errors do not cancel each other (BusinessDictionary, 2016) and can present inaccurate trends. Data cleaning approaches utilizing pattern-based detection are ideal for identifying this error type.

Another approach to classifying data errors is based on the effort required to correct it. Lucas, et al (2014) present a hierarchy of errors using this approach. Based on this hierarchy, we divide errors into three categories:

Field Errors – simple errors that tend to be easy to identify and correct.

Multi-Field Errors – moderately complex errors that are more difficult to identify and correct. Multi-field errors may occur due to contradictory information in two related records or within the same record on a single table. These errors involve deeper consideration of the data and what it represents, particularly in the case of contradictory information where it must be decided which source is correct.

Domain Errors – highly complex errors that are possible to correct only with significant effort, a specialized skillset, or tacit knowledge of what the data represents. Domain errors are the most difficult to identify and correct.

Costs and Challenges

Eppler and Helfert (2004) identify three categories of costs associated with improving and ensuring data quality: prevention, detection, and repair costs. Prevention costs are the lowest (Eckerson, 2002; Van den Broeck et al., 2005) and arise from training, monitoring, and developing and deploying appropriate systems and processes. Detection costs stem from analysis and reporting; repair costs relate to planning and implementing corrections to the data.

As the complexity of an error increases, the costs associated with detection and repair increase. Moderately and highly complex errors require additional resources to correct than simple errors (Lucas et al., 2014). Given the nature of these more complex errors, greater levels of expertise or time with the data are required – particularly with domain errors – which would entail increased labor costs from either higher-salary employees addressing the errors or lower-salary employees spending a considerably greater amount of time on the task.

Perfect data quality is an unrealistic expectation for multiple reasons, the most relevant of which is costs. An optimal level of quality exists where the total data costs (the cost to maintain and clean added to the costs incurred from having 'dirty data') are lowest (Eppler and Helfert, 2004; Haug et al., 2011). Methods that reduce costs associated with error detection and achieve an optimal level of data quality should prove valuable in application.

Data Profiling

Data profiling is the first phase in the data cleaning process: define and determine error types (Maletic and Marcus, 2000). More specifically, it is the examination and assessment of a data source for quality, integrity, and consistency (Singh and Singh, 2010) and analyzes instances of individual attributes. The information gained from profiling includes data type, length, value range, frequency of discrete values, uniqueness, frequency of null values, variance, and other metadata for each attribute. This provides an understanding of the quality level of the source and the types of errors that exist. Profiling can be accomplished through database queries and is ideal for identifying field errors. More complex errors require approaches more closely associated with data mining (Rahm and D0, 2000).

Data Quality Mining

In recent years, data quality mining has been necessitated by the growth of and the analysis of big data (Hipp et al., 2004). Data quality mining was introduced in 2001 by Hipp et al. to apply data mining techniques for the purpose of measuring and improving a dataset's quality. The research was an extension of traditional data mining techniques – which became a prominent field of research in the early 1990s. A variety of techniques exist for data quality mining, including functional dependencies (FDs), association rules, bagging support vector machines (SVMs), clustering, pattern-based detection, and various statistical

procedures (Maletic and Marcus, 2000; Natarajan et al., 2009; Rahm and Do, 2000). Our research focuses on FDs.

Functional Dependencies

Research on functional dependencies began before they were used as a data quality mining technique. Early research aimed to find effective ways to discover FDs via algorithms (Ilyas et al., 2004; Huhtala et al., 1999). TANE is perhaps the most notable of these algorithms as it was later utilized as the discovery algorithm for the data cleaning process proposed by Kaewbuadee et al. (2006). Their discovery algorithm was found to identify useful FDs, identify duplicates and anomalies, and clean data about as effectively as manual processes. The preliminary test of the data was performed on a dataset of 50,000 records and 15 attributes; the authors anticipated needing additional testing with larger datasets as there were concerns with additional noise FDs and performance degradation. Despite this shortcoming, the algorithm had low false positives.

As data quality mining was adopted, researchers realized these algorithms were generating too many FDs, many of which were trivial. Various pruning and ranking techniques were introduced to quell the problem (Ilyas et al., 2004; Huhtala et al., 1999). Such techniques only allowed the most meaningful FDs to be output, optimizing the process. More recently, the technique has been extended to develop methods for finding a higher percentage of errors in data. Most prominently, FDs were extended to conditional dependencies – FDs that hold true only under certain conditions (Fan, 2008).

A functional dependency is an informal relationship that exists in a dataset when one attribute uniquely determines another attribute. X –> Y specifies that Y is functionally dependent on X. If all tuples that have matching values for X also have matching values for Y, then the functional dependency holds. In this example, X is the determinant value while Y is the dependent value. The most common example of a functional dependency is the relationship between zip code and city. City is functionally dependent on zip code such that zip code –> city. Once a functional dependency constraint is determined, violations of the constraint can be found. In many cases, these violations will be errors in the data. Take for example the data in Table 1.

Table 1. Example Data

	Name	ZipCode	City
Tuple A	Donald Smith	64105	Kansas City
Tuple B	Maddie Doe	64105	Kansas City
Tuple C	John Silver	63111	St. Louis
Tuple D	David Green	63111	St. Louis
Tuple E	Jane Error	64105	St. Louis

Table 1. An Example of Functional Dependency

Tuple E violates the functional dependency because the zip code 64105 has a different value for city than *Tuples A & B*. Such errors are the subject of our research as they can be quickly identified once the functional dependency is discovered. Potential errors can be pulled using the following SQL query:

```
SELECT DISTINCT ZipCode, City
FROM ExampleData t1, ExampleData t2
WHERE t1.ZipCode = t2.ZipCode AND t1.City <> t2.City;
```

A similar SQL query can be used to determine if a functional dependency holds. The following query will identify the number of violations for the example above:

```
SELECT Count(*) as violations
FROM ExampleData t1, ExampleData t2
WHERE t1.ZipCode = t2.ZipCode AND t1.City <> t2.City;
```

If the query returns zero, then a functional dependency holds. Additionally, if the query returns a low number relative to the number of potential violations, then there may still be a functional dependency but with errors.

Research Framework

The increasing importance of big data necessitates better quality data. Given the volume of data available in many organizations, it is impractical to expect validation on each data point; however, these datasets – whether accompanied by proper documentation and metadata or not – are accessible to a wide array of individuals for inspection and analysis (Kaisler et al., 2012). This creates two needs:

1. a way to quickly and effectively identify more complex errors in order to prepare data for analysis, and
2. a way to develop a degree of domain knowledge quickly in order to assess if the output of an analysis is logical.

Much of the research on data mining with functional dependencies focuses on developing more accurate algorithms; however, perfectly clean data is an unattainable goal as it is cost prohibitive and resource intensive (Eckerson, 2002; Lucas et al., 2014; Haug et al., 2011). Most research also is written from a perspective where the data is used by someone internal to the company and who possesses at least some domain knowledge. It is not uncommon, though, for individuals less familiar with a dataset to be working with it, particularly for research.

Given these perspectives, it is useful to provide a process to efficiently identify errors while also accommodating low degrees of domain knowledge. The process described next is intended to help develop better domain knowledge while reducing the costs associated with identifying multi-field errors.

Process

Our research aims to develop a process to optimize the identification of multi-field errors with respect to time and other monetary costs. Multi-field errors rely on relationships between fields to identify and correct 'dirty data,' presenting an ideal error type for developing an error identification approach using functional dependencies. Field errors do not rely on relationships for identification and thus would be inappropriate for a process using FDs. Identifying domain errors requires tacit knowledge of the real-world entities represented by the data; thus, this error type cannot be identified using solely FDs.

There are two steps paramount to this research goal:

1. the identification of the most useful functional dependencies, and
2. the identification of the most common errors within those functional dependencies.

Identification of Functional Dependencies

In the context of this research, the most useful FDs are the ones containing the highest number of identifiable errors. Dependencies with systematic errors are particularly valuable because once an error is identified, all instances can be eradicated throughout the relationship. Ultimately, the objective is to identify the functional dependencies that will lead to the most errors being identified at the lowest costs.

In order to develop an effective algorithm to identify meaningful FDs, our research will extend a successful discovery algorithm, TANE, and consider new factors that are meaningful in the context of this research. While the TANE algorithm outputs the strongest FDs, our extension of the TANE algorithm will output the FDs containing the most identifiable errors. The output of TANE will be considered against the metadata of the attributes involved – such as the number of distinct or null values and the data collected in our Identification of Errors process (presented next). Meaningful insights will be incorporated into our algorithm in order to re-prioritize the output and optimize it for cleaning purposes. Regression, clustering, and decision trees will all be evaluated to determine utility in solving this problem.

In more depth, the areas that will be considered in the development of the algorithm are:

1. *Refinement & Prioritization* – Previous research has worked to refine and prioritize the FD discovery process. This past research will be the starting point for our research, which will focus on refining and prioritizing FDs based on the number of identifiable errors they contain. Data collected throughout the profiling process is valuable in this step because metadata on an attribute, such as the number of distinct values or the number of null values, could potentially be used to identify relationships that would be trivial to the identification of errors. Preemptively refining the FDs to be tested could expedite the discovery process by removing noise. Inversely, factors could be identified that make a potential FD particularly valuable for error detection. Such factors should be used to prioritize FDs for cleaning.
2. *Common Error Identification* – The more prevalent an error is throughout the dataset, the more impact the error will have on analysis. In addition to this, common errors present an opportunity to clean multiple errors quickly. Thus, targeting FDs with systematic errors is important to the optimization of this process and should be considered during refinement and prioritization. Factors contributing to the likelihood of systematic errors need to be identified. Additionally, the prevalence of random versus systematic errors needs to be assessed.

Identification of Errors

After meaningful FDs have been discovered and prioritized, the process to identify errors begins. Violations of the FDs are potential errors and the subject of our investigation. To identify errors, the values of the determinant attribute and the dependent attribute need to be compared against the tuple with which it violates. The violation that is found in the Table 1 data is shown in Table 2.

Table 2. Example Violation			
Determinant 1	Dependent 1	Determinant 2	Dependent 2
64105	St. Louis	64105	Kansas City

Table 2. Format for Examining Violation

Previous research has worked to automate the cleaning process with preliminary success in regards to matching the effectiveness of manual processes (Kaewbuadee et al., 2006). Our preliminary research will involve manual examination of the violations. This decision was made to achieve a deeper understanding of errors within FDs and to limit the scope of this research. Manual examination is necessary to collect data that would otherwise be difficult to obtain. Information produced from manual examination will include:

1. Proportion of random errors vs. systematic errors found in FDs
2. Factors leading to the prevalence of errors in FDs
3. Factors leading to successful error identification in FDs

The data from this manual examination will then be utilized alongside the current ranking of FDs (output of TANE) and the basic metadata from data profiling to craft the algorithm to identify meaningful FDs. In this way, the data from the latter half of our process will be utilized to enhance the former half of the process.

Upon manual examination, FD violations can be classified into three broad groups:

False Positive – the violation is not an error but instead simply an exception to the dependency. False positives should be ignored.

Error – the violation is an error. These errors can be identified and often fixed without further research. In many cases, similar tuples can be examined to determine the correct value.

Unknown – not enough information is known to determine whether the violation is an error or false positive. Correcting potential errors in this category has a significantly higher cost due to the time required to determine their proper categorization.

Our research will focus on finding and correcting errors. The preliminary design for an error detection algorithm is presented in Figure 1.

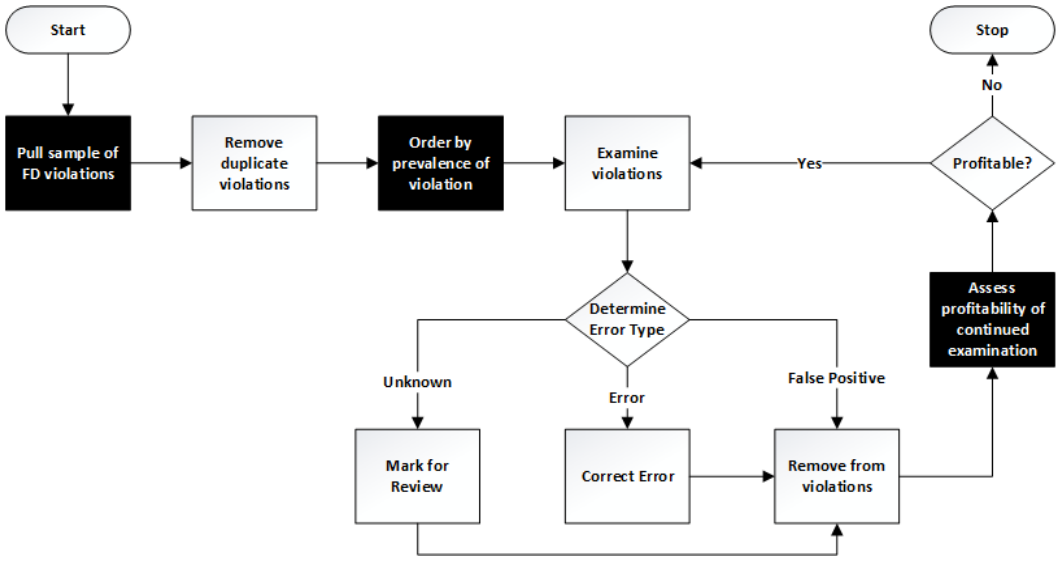

Figure 1. Preliminary Process for Error Identification and Removal

Boxes shown in black represent areas of the process that will undergo further research. To begin the process, a sample of violations will be retrieved from the dataset. Only unique errors will be pulled as to simplify the examination process. Once the errors are retrieved, they will be prioritized to optimize the time spent examining violations and then the manual process of examining errors will begin. During the examination process, the relative frequency of false positives, unknowns, and errors will be recorded to draw further conclusions using a GUI interface that has been designed and developed to facilitate this phase. At some point, it will be more useful to move to another FD than to continue cleaning the current one, signifying the end of the error identification process for an FD. This process will be further refined as research continues, following these three directions:

1. *Random Sampling* – In large databases, retrieving all violations could be a time consuming process. In order to expedite this task, random sampling will be used – which is consistent with research performed by Ilyas et al. (2004). We hope to extend this and utilize a random sampling technique to retrieve violations within FDs. The ability to assess the prevalence and nature of an error using random sampling is of particular importance; however, further research needs to be conducted to determine its viability and effectiveness. Multiple samples will be pulled from FDs and compared to ensure sampling results in adequate coverage of the population. Additionally, different sampling sizes will be used and analyzed for effectiveness to determine the optimal approach.
2. *Ranking Violations* – Violations need to be prioritized for review based on both the likelihood the violation is an error and the magnitude of the violations within the FD. Ensuring the examination of the most prevalent violations will allow for the most potential errors to be examined in the least amount of time. As a starting point, violations will be prioritized strictly by prevalence. As research progresses, the method to rank violations will develop to further prioritize those likely to result in identifiable and correctable errors. Characteristics of the violation, including the number of unique dependent values per determinant value, will be considered in the prioritization process.
3. *Assessing Profitability* – At some point in the error identification process, it becomes more valuable to remove errors from another FD than to continue the error identification in the current one. Determining the optimal point at which to move to another FD is within the scope of our research. Profitability will be assessed by investigating the rate of diminishing marginal return on the number of errors identified as examination continues.

The output of this phase of the research will be data collected from the examined FDs and an optimized process for the manual identification of errors within FDs. The process will likely resemble Figure 1 (above), but include optimized algorithms – created for random sampling, ranking violations, and assessing profitability – to improve the speed and ease of error identification.

Acquisition of Domain Knowledge

Throughout the process, the primary goal is to identify errors in the dataset. However, domain errors exist that cannot be cleaned using FDs. These errors require extensive knowledge of the domain to correct (Lucas et al., 2014). Often, datasets are not fully understood by those cleaning or analyzing them. A name and datatype is not always enough information to determine the meaning of an attribute and a frequent lack of adequate documentation further complicates the issue (Sapia et al., 1999).

Fortunately, insight into the dataset can be gained throughout this process. Establishing connections within and between tables using FDs – instead of relying on foreign keys – leads to an understanding of the semantics of a dataset as well as knowledge of the informal relationships within the dataset. This additional understanding can lead to significant cost reductions as the data cleaning process continues.

Determinacy diagrams can be drawn to visualize the informal relationships in the database. The diagram will allow unfamiliar users insights into the domain of the data and can be maintained as part of the dataset's metadata, particularly if adequate documentation is lacking. In addition to this, data validation rules can be created from dependencies to ensure proper data quality. Errors can then be corrected before entering the dataset, further improving data quality with little expense.

While neither technique will lead directly to the identification of domain errors, both of these techniques will lead to an improved understanding of the data that can be used to enhance overall data quality. Continuing to find more ways to leverage the information gained throughout our process is an important aspect of our research.

Conclusion

In this research-in-progress, we have set a groundwork and ongoing direction for research into a process for efficient multi-field error detection through functional dependencies. We have established the relationship between data profiling and data quality mining in relation to data cleaning. We have also discussed previous research foci in regards to functional dependencies; in doing so, we have explained how our research extends existing work in considering the impact of cost, encouraging the use of FDs to improve domain knowledge, and recognizing the impracticality of perfectly clean data.

The proposed process will rely on previous research and data profiling to refine and prioritize the FDs to be used; prioritization will be based primarily on the prevalence of errors, particularly systematic errors, within each FD. Then, random sampling will be used to identify specific violations; these will be ranked on the likelihood of being an error – as opposed to a false positive – and the magnitude of the violations. As research continues, the ranking will evolve to also consider the identifiability and correctability of errors. After examining a violation, an assessment of the profitability to continue cleaning within that FD will be made; this recognizes, at some point, it is more profitable to end examination of violations on the current FD and move onto the next.

The next phase of this research is to begin testing the proposed process with a large dataset. This will allow us to determine the practicality of this approach and to assess needed adjustments in 1) refining the list of FDs to be mined and in 2) ranking the violations to be examined. We anticipate having preliminary results by the start of the conference.

Acknowledgements

We thank Jeff Lucas for his support and feedback during the early stages of writing and conceptualizing this research-in-progress. We also thank Uzma Raja for her continued support and feedback throughout the process thus far and her many questions that have helped refine our focus.

References

Bohannon, P., Fan, W., Geerts, F., Jia, X., and Kementsietsidis, A. 2007. "Conditional Functional Dependencies for Data Cleaning," in *2007 IEEE 23rd International Conference on Data Engineering* IEEE, pp. 746-755.

BusinessDictionary.com. 2016. "What is Random Error," Web Finance, Inc. http://www.businessdictionary.com/definition/random-error.html

BusinessDictionary.com. 2016. "What is Systemic Error," Web Finance, Inc. http://www.businessdictionary.com/definition/systemic-error.html

Eckerson, W. 2002. "Data Quality and the Bottom Line," *The Data Warehousing Institute*.

Eppler, M. and Helfert, M. 2004. "A Classification and Analysis of Data Quality Costs," in *Proceedings of the Ninth International Conference on Information Quality*, pp. 311-325.

Fan, W. 2008. "Dependencies revisited for improving data quality," *PODS*, pp. 159-170.

Fan, W., Geerts, F., and Jia, x. 2008. "A Revival of Integrity Constraints for Data Cleaning," in *Proceedings of the VLDB Endowment* (1:2), pp. 1522-1523.

Gartner, Inc. 2007. "'Dirty Data' is a Business Problem, Not an IT Problem, Says Gartner." [Press Release].

Haug, A., Zachariassen, F., and Liempd, D. 2011. "The Costs of Poor Data Quality," *Journal of Industrial Engineering and Management* (4:2), pp. 168-193.

Hipp, J., Güntzer, U., and Grimmer, U. 2001. "Data Quality Mining: Making a Virtue of Necessity," in *Proceedings of the 6th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 52-57.

Huhtala, Y., Kärkkäinen, J., Porkka, P., and Toivonen, H. 1999. "TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies," *The Computer Journal* (42:2), pp. 100-111.

IBM. 2012. "Successful information governance through high-quality data," pp. 1-12

Ilyas, I., Markl, V., Hass, P., Brown, P., and Aboulnaga, A. 2004. "CORDS: Automatic Discovery of Correlations and Soft Functional Dependencies," in *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, ACM, New York, NY, pp. 647-658.

Kaewbuadee, K., Temtanapat, Y., and Peachavanish, R. 2006. "Data Cleaning Using FD from Data Mining Process," *International Journal of Computer Science and Information System* (1:2), pp. 117-131.

Kaisler, S., Armour, F., Espinosa, J., and Money, W. 2012. "Big Data: Issues and Challenges Moving Forward," in *2013 46th Hawaii International Conference on Systems Science* IEEE Computer Society, pp. 995-1004.

Lucas, J., Raja, U., and Ishfaq, R. 2014. "How Clean is Clean Enough? Determining the Most Effective Use of Resources in the Data Cleansing Process," in *Thirty Fifth International Conference on Information Systems*, pp. 1-10

Maletic, J. and Marcus, A. 2000. "Data Cleansing: Beyond Integrity Analysis," in *Proceedings of the 2000 Conference on Information Quality*, pp. 200-209.

Marsh, R. 2005. "Drowning in dirty data? It's time to sink or swim: a four-stage methodology for total data quality management," *The Journal of Database Marketing & Customer Strategy Management* (12:2), pp. 105-112.

Michnik, J. and Lo, M. 2009. "The assessment of the information quality with the aid of multiple criteria analysis," *European Journal of Operational Research* (195:3), pp. 850-856.

Natarajan, K., Li, J., and Koronios, A. 2009. "Data Mining Techniques for Data Cleaning," in *Proceedings of the 4th World Congress on Engineering Asset Management*, pp. 796-804.

Pipino, L., Lee, Y., and Wang, R. 2002. "Data Quality Assessment," *Communications of the ACM* (45:4), pp. 211-218.

Rahm, E. and Do, H. H. 2000. "Data Cleaning: Problems and Current Approaches," *IEEE Data Eng. Bull.* (23:4), pp. 3-13.

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

Sapia, C., Höfling, G., Müller, M., Hausdorf, C., Stoyan, H., and Grimmer, U. (1999). "On Supporting the Data Warehouse Design by Data Mining Techniques." Venue: GI-Workshop: Data Mining and Data Warehousing, September 27-28.

Singh, R. and Singh, K. 2010. "A Descriptive Classification of Causes of Data Quality Problems in Data Warehousing," *International Journal of Computer Science Issues*, (7:3:2), pp. 41-50.

Van den Broeck, J., Argeseanu Cunningham, S., Eeckels, R., and Herbst, K. 2005. "Data Cleaning: Detecting, Diagnosing, and Editing Data Abnormalities," *PLoS Med* (2:10): e267. Doi: 10.1371/journal.pmed.0020267