Analysis of Data Quality Problem Taxonomies

Arturs Zogla[1], Inga Meirane[2] and Edgars Salna[2]
[1]National Library of Latvia, Mukusalas 3, Riga, Latvia
[2] Datorzinibu centrs JSC, Lacplesa str. 41, Riga, Latvia,
arturs.zogla@lnb.lv, {inga.meirane, edgars.salna}@dzc.lv

Keywords: Data Quality, Data Anomalies.

Abstract: There are many reasons to maintain high quality data in databases and other structured data sources. High quality data ensures better discovery, automated data analysis, data mining, migration and re-use. However, due to human errors or faults in data systems themselves data can become corrupted. In this paper existing data quality problem taxonomies for structured textual data and several improvements are analysed. A new classification of data quality problems and a framework for detecting data errors both with and without data operator assistance is proposed.

1 INTRODUCTION

Errors in data occur for various reasons and are caused both by human data operators and by mistakes in information systems themselves. Low quality data in turn limits the ways it can be searched, analysed, and re-used. In a wider sense some data quality problems are not errors. If a database for some products lists prices in US dollars and for other – in Euros, this might be considered a data quality issue, but not necessarily an error.

A lot of research has been done on detecting and resolving data quality problems. Most of this research, however, has been on data quality problems in non-structured or semi-structured data: images, audio signals, network packets, etc (Chandola V. et. al., 2009).

These data quality problems are usually referred to as data anomalies. It has been suggested to group all data anomalies into three classes (Chandola V. et. al., 2009):

- point anomalies for individual objects that can be considered as anomalous relative to other objects;
- context-sensitive anomalies for data that is anomalous only in certain conditions (freezing temperatures in summer);
- collective anomalies for several objects that are anomalous only when viewed together (two passengers having tickets for the same seat in an airplane).

This paper focuses on data quality (DQ) problems in structured textual data like relational databases, CSV and XML documents. Several attempts have been made to provide extensive taxonomies of data quality problems in structured textual data. One such taxonomy groups data quality problems into a hierarchy of 33 classes (Kim W., et. al., 2003).

Another taxonomy contains 35 data quality problem classes grouped by context in which a particular data quality problem appears (Oliveira P., et. al., 2005). The existing research on DQ problems in structured data, however, has been mostly theoretical or has been applied for a particular purpose – as part of data migration, data mining, data transofrmation, etc.

Therefore, the goal of the research is to apply existing, theoretically proposed DQ problem taxonomies to real life data and provide necessary improvements to DQ problem taxonmy based on real data errors.

In the research one of the existing DQ problem taxonomies was chosen and validated against several databases used at National Library of Latvia. In the process new error classes were identified.It is also suggested that some DQ problem classes might be merged, because of almost identical DQ problem instances.

2 TAXONOMY OF DATA QUALITY PROBLEMS IN STRUCTURED DATA

By analysing different publications of DQ problem taxonomies (Kim W., et. al., 2003), (Li L., 2011), (Hernandez M. A., Stolfo S. J., 1998) and (Rahm E., Hai Do H., 2000) one was selected as being the most appropriate for structured textual data (Oliveira P., et. al., 2005). It provides the most detailed structure with 35 DQ problem classes. It also has a clear and accessible structure that makes it easy for data operators to classify individual instances of data errors.

This taxonomy groups all DQ problems into 6 subgroups by the context in which a particular DQ problem occurs:

- An attribute value of a single tuple;
- The values of a single attribute;
- The attribute values of a single tuple;
- The attribute values of several tuples;
- Multiple relations;
- Multiple data sources;

For example, a typical DQ problem in "The values of a single attribute" group was a Synonyms existence problem, which occurs when several objects contain different values to express the same concept in a particular attribute (using both "F" and "Female" as a value for Gender attribute).

2.1 Validation of Data Quality Problem Taxonomy

To validate the data quality taxonomy several databases of National Library of Latvia were chosen. These databases are considered to be particularly suitable for collecting DQ problem instances for following reasons:

- Some of the databases have been around for more than 20 years;
- Data has been input by many data operators with different approaches towards how detailed objects should be described;
- Methodology of describing the same type of objects has changed many times over the years;
- In some databases little to no data validation has been enforced;
- Data has been migrated between different systems and different versions of the same system.

All of these can be considered as prerequisites for DQ problems to occur.

2.1.1 Validation Methods

Following databases were selected for testing purposes (see Table 1).

Table 1: Databases used for validating DQ problem taxonomies.

Database	Number of data objects
Library's electronic catalogue	4 100 000
Authority data	205 000
Digital object management system	58 000
Digital collection "Lost Latvia"	30 000

It is worth noting that Library's electronic catalogue is integrated with the Digital object management system. First one describes physical collection of NLL while the other one – digitized versions of these objects. This means that multiple data source DQ problems could also be validated.

Two approaches were used to validate DQ problem taxonomy: qualitative and quantitative methods. Qualitative method is used to refer to a process of identifying as many different DQ problem class instances as possible. In the best case scenario this would mean providing at least one example for each DQ problem class.

To achieve this goal, it was decided to choose the most likely people to identify different DQ problems – the most experienced data operators for each of the selected databases. It was clear that for some DQ problem classes it will be hard or impossible to find instances in real data. Mostly, because DQ problems, if detected, are almost always immediately corrected. So two kinds of DQ problem instances were allowed:

- Real DQ problems, where data operator could provide an existing object with a particular DQ problem;
- Hypothetical DQ problem, where data operator could only describe or recall an instance of a DQ problem or suggest a possible scenario of a DQ problem occurring.

Quantitative method refers to a process of analysing data and looking for accidental errors and collecting as many DQ problems as possible. The reason for doing this kind of analysis was to identify most prominent and most frequently occurring DQ problems. Unlike for qualitative method, quantitative analysis is best performed by data operators relatively unfamiliar with data sources as they might notice DQ problems, which an experienced data operators might overlook. Another reason for doing quantitative analysis was to possibly identify new DQ problem classes that didn't exist in the selected DQ problem taxonomy.

2.1.2 National Library of Latvia's Most Important Data Quality Problems

By performing qualitative analysis data operators collected a total of about 200 instances of DQ problems. Real or hypothetical data errors were found for almost every DQ problem class from the selected DQ problem taxonomy with an exception of Different measurement unit error in a case of multiple data sources. This data error may occur, for example, when in case of two related data sources, one uses the metric system, but the other – imperial measurement system. Because all data sources at National Library of Latvia use the same measurement systems no data errors of this kind were found or even suggested. However, it is obvious that this error may exist in other cases and therefor this DQ problem class should be kept.

One task of qualitative analysis was to determine how well-founded are each of the DQ problems from the original taxonomy. It was determined that data operators often confused several DQ problem classes and required detailed explanations of the meanings of these classes. The most frequently confused DQ problem class pairs were:

- Syntax violation and Misspelled error classes;
- Set violation and Interval violation classes;
- Outdated value and Value items beyond the attribute context classes.

This suggests that several DQ problem classes from the original taxonomy might be merged to avoid misunderstandings in classifying particular instances of DQ problems.

Quantitative analysis was performed by a third-party relative to the ownership of data sources. Their task was to identify DQ problems that stood out for a person relatively unfamiliar with the data source. One obvious type of DQ problem that can be identified this way is the Set violation error, when a data field contains a value outside of a pre-defined set of values. In a data field with a limited set of data values those that occur relatively rare compared to others can be suspected for being incorrect.

Table 2 provides values of a Document Type data field and number of times each was used in one of the tested data sources. In several cases this data field also contained the dimensions of the object, although a separate data field was available for this information.

Table 2: Values in a Document Type data field in digital collection "Lost Latvia".

DQ problem	Value	Used number of times
Syntax error	1 postcard	1
Misspelled error	1 potstcard	1
None	1 postcard	393
Value items beyond the attribute context	1 postcard, 8,5 x 13,5 cm	1
Value items beyond the attribute context	1 postcard, 8,5 x 13,5 cm	5
Value items beyond the attribute context	1 postcard, 8,7 x 13,6 cm	1
Value items beyond the attribute context	1 postcard, 8,7 x 13,7 cm	1
Value items beyond the attribute context	1 postcard, 8,8 x 13,5 cm	1
Value items beyond the attribute context	1 postcard, 8,8 x 13,8 cm	1
Value items beyond the attribute context	1 postcard, 9 x 14 cm	1
Syntax error	1. postcard	1

Even without knowing the content requirements of the particular data field, it is obvious that the data value that appeared 393 times is probably the "normal" one, while others should be considered at least suspicious and should be provided for manual review. In fact, when provided with this table of values, data operators at the National Library of Latvia indeed admitted that the most frequently used value is the correct one, while others are results of human errors.

Finally, an empirical study was performed where data operators were asked to name what they consider 3 most important DQ problems each in their particular data sources. For data sources used in this research following DQ problems were named by at least two data operators as important for their data source:

- Value syntax error;
- Set violation error;
- Outdated values;
- Duplicate objects.

Note that these are not necessarily the most frequently occurring data errors, but the DQ problems that data operators consider have the most overall impact on the operation of data source. For example, Set violation error and Outdated values error cause deterioration of data search and retrieval performance.

2.2 Proposed Improvements to Data Quality Problem Taxonomy

While performing validation of the existing DQ problem taxonomy (Oliveira P., et. al., 2005), it is noticed that some improvements to the taxonomy are necessary.

2.2.1 Identification and Correction of Data Quality Problems

First of all, for future development, it is important to understand, instances of which DQ problem classes can be identified and corrected automatically and which will require a partial or full manual assistance from data operators.

Table 3 summarizes our initial estimates of how many DQ problem classes can be processed automatically and how many would require partial or complete manual assistance.

Table 3: Detection and correction of DQ problem class instances.

Detection and correction method	Number of DQ problem classes that can be detected	Number of DQ problem classes that can be corrected
Completely automatically	19	7
Partially manually (with data operator assistance)	4	8
Only manually	12	20
Total	35	35

As expected detecting a DQ problem in general is much easier than correcting that error. For example, it is easy to check whether all mandatory data fields have values, but if a particular one does not, in most cases without data operator assistance, it is impossible to guess the missing value.

Only one DQ problem class is identified where correcting error is considerably easier than identifying it. It might be very hard just by looking at data to detect that two data sources use different measurement units. However, once established that data source X uses one measurement unit and data source Y another one, values can be easily transformed using simple arithmetic. For example, a comparatively simple correspondence exists between metric and imperial measurement systems, where conversion between those systems is done by simple multiplication.

2.2.2 Modification of the DQ Problem Class Structure

By analysing how well data operators could separate between different DQ problem classes from the original taxonomy (Oliveira P., et. al., 2005), we identified following DQ problem classes where no meaningful data error example could be provided to separate the two: Inadequate value to the attribute context and Value items beyond the attribute context. So merging of these two DQ problem classes can be suggested.

The first error describes cases where a value is input into a wrong data field, while the second error describes cases where data field contains a complex value where parts of it would most appropriately have been input in other data fields. These two errors represent just a slightly more general case of Redundancy errors.

A new DQ problem class can be proposed that does not appear in the original taxonomy – Factual errors. Such errors may appear in data fields that contain natural language data values and consequently may contain factual information. An example of this DQ problem would be a data field containing value: “The painting is located in the capital of France – London,” where the statement that London is capital of France is clearly a factual error.

Original DQ problem taxonomy considered situations where only individual errors occur. However, in real life scenarios a combination of two or several different DQ problems might be simultaneously present even in a single data field.

In fact, Misspelling error can cause almost every other kind of DQ problem as well. For example, if a person’s birth year is misspelled as “19743” instead of “1974”, this will be both a Misspelling error and an Interval violation error. Other DQ problem combinations may exist, like: Syntax error/Set violation, Set violation/Outdated value, etc.

The fact that DQ problem combinations may exist requires establishing a certain order in which DQ problems are identified in order to minimize the number of suspected data errors. For example, typically correcting misspelling errors first will also automatically correct other suspected DQ problems as well.

Following order can be proposed in which DQ problems from a category “an attribute value of a single tuple” should be processed:

- Missing value;
- Misspelling error;
- Syntax error;
- Interval violation;
- Outdated value;
- Set violation;
- Redundant value;
- Meaningless value;
- Factual error.

This, for example, means that factual errors should only be checked after all other kinds of DQ problems have been resolved.

Here the processing order of "an attribute value of a single tuple" DQ problems is presented as they are the most widely encountered, but such order obviously can be established for other groups of DQ problems as well.

While validating the original DQ problem taxonomy, we identified several DQ problems that can only be detected if sufficient external information is provided. For example, to detect a Set violation problem the set of all possible values must be provided.

3 FRAMEWORK FOR DETECTING DATA QUALITY PROBLEMS

Section 2 of this paper describes the evaluation of an existing DQ problem taxonomy with suggested improvements. While granularity of this taxonomy can be considered as optimal for classification of individual instances of DQ problems, it is too detailed when defining criteria for automatic identification of data errors.

Although DQ problems like Set violation, Outdated value and Syntax error all might have very different causes, the algorithms used to identify these DQ problems are almost identical. It can be considered that these DQ problems are examples of a more general type of data error: Incorrect value.

The detailed version of DQ problem taxonomy is valuable for analysing reasons that certain errors appear more frequently than others, however, DQ problems for identification purposes can be grouped in more general classes.

A new approach of grouping all DQ problems into 8 classes can be proposed, where each class of DQ problems is identified with a particular type of algorithms:

- Empty data field (for mandatory and semi-mandatory data fields);
- Incorrectly formatted value;
- Incorrect use of special characters;
- Contradicting values in linked data fields;
- Outdated values;
- Set violation;
- Spelling/grammar errors;
- Object duplicates.

Furthermore, for each of these DQ problem classes two types of algorithms can be provided:

- Assisted algorithm. If additional information is available;
- Autonomous algorithm. If no additional information is available and the algorithm essentially must "guess" that a particular DQ problem exists.

For example, an assisted algorithm for an Incorrectly formatted value DQ problem class would receive as an input list of data fields that have specifically formatted values and a regular expression that describes the expected format of values. An autonomous algorithm on the other hand for the same DQ problem class would "guess" both which data fields contain specifically formatted values and the corresponding regular expressions for value formats.

Although autonomous algorithms in general are more complex than assisted algorithms, prototypes for these algorithms have been developed for each of the DQ problem classes, except for Spelling/Grammar error class, where at least an external vocabulary is required to perform a meaningful detection of this kind of DQ problem.

4 CONCLUSIONS

This paper reflects a research in progress on finding and correcting data quality problems in structured textual data. Existing DQ problem taxonomies were evaluated for structured data and one that was most detailed and easy to use was adopted.

This taxonomy was then tested on real data from several data sources of National Library of Latvia. Some suggestions on how to improve the original taxonomy are given in Section 2.2 of this paper.

Finally, a new regrouping of DQ problems is proposed based on what algorithms can be used to detect individual DQ problems. A new DQ problem classification consisting of 8 groups of DQ problems is proposed which can then be detected using two types of algorithms: assisted and autonomous.

Like in the case of detecting DQ problems, algorithms and methods for correcting DQ problems will be developed. In general not always this will be possible without additional input from data operators (see Table 3). However, with sufficient external information many DQ problem classes can be corrected.

A new universal tool will be developed that will take any kind of structured data as input and will detect and, where possible, correct DQ problems based on criteria provided by data operators. This tool will be validated by detecting and correcting DQ problems in databases of National Library of Latvia.

ACKNOWLEDGEMENTS

This research is part of the project "Competence Centre of Information and Communication Technologies" run by IT Competence Centre, contract No. L-KC-11-0003, co-financed by European Regional Development Fund, Research No. 1.18 “Data array quality analysis and enhancement technologies”. More information at http://www.itkc.lv/.

REFERENCES

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly Detection: A Survey. *ACM Computing Surveys (CSUR)*, 41 (3). ACM New York, NY, USA. pp. 1-72.

Li, L., Peng, T., Kennedy, J., 2011. A Rule Based Taxonomy of Dirty Data. *GSTF International Journal on Computing*, 1 (2). Singapore. pp. 140-148.

Oliveira, P., Rodrigues, F., Henriques P., Galhardas H., 2005. A Taxonomy of Data Quality Problems. In *2nd Int. Workshop on Data and Information Quality (in conjunction with CAiSE 2005)*, Porto, Portugal, June 14, 2005.

Kim W., et. al., 2003. A Taxonomy of Dirty Data. *Data Mining and Knowledge Discovery*, 7. Kluwer Academic Publishers, 2003. Manufactured in The Netherlands. pp. 81–99.

Rahm, E., Hai Do, H., 2000. Data Cleaning: Problems and Current Approaches. *Bulletin of the Technical Committee on Data Engineering*, 23 (4).

Hernandez, M. A., Stolfo, S. J., 1998. Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. *Data Mining and Knowledge Discovery*, 2. Kluwer Academic Publishers, 1998. pp. 9–37.