A CLASSIFICATION OF DATA QUALITY ASSESSMENT METHODS

(Completed paper)

Alexander Borek
University of Cambridge
ab865@cam.ac.uk

Philip Woodall
University of Cambridge
phil.woodall@eng.cam.ac.uk

Martin Oberhofer
IBM Germany Research & Development
martino@de.ibm.com

Ajith Kumar Parlikad
University of Cambridge
ajith.parlikad@eng.cam.ac.uk

Abstract: Data quality (DQ) assessment can be significantly enhanced with the use of the right DQ assessment methods, which provide automated solutions to assess DQ. The range of DQ assessment methods is very broad: from data profiling and semantic profiling to data matching and data validation. This paper gives an overview of current methods for DQ assessment and classifies the DQ assessment methods into an existing taxonomy of DQ problems. Specific examples of the placement of each DQ method in the taxonomy are provided and illustrate why the method is relevant to the particular taxonomy position. The gaps in the taxonomy, where no current DQ methods exist, show where new methods are required and can guide future research and DQ tool development.

Key Words: DQ Assessment Methods, DQ Software Tools, DQ Assessment

INTRODUCTION

Data quality (DQ) assessment provides the basic foundation for improving DQ in organizations. DQ assessment in larger information systems would often be not feasible without using suitable DQ assessment methods, which are algorithms that can be automatically executed by computer systems to assess certain aspects of DQ. Many of the current DQ software tools that implement these algorithms can be applied to databases that are in use in the majority of organizations today. Automated DQ methods for databases are, therefore, the focus of this paper. A DQ software tool can encompass a number of DQ methods and the implementation of these methods can differ considerably between different software tools. Each DQ assessment method, however, focuses on a particular set of DQ problems (e.g. misspelling, correctness, duplicate data etc.) and is only applicable in certain circumstances, for instance, some DQ assessment methods assess the quality of single values in databases, whereas other methods

assess DQ using multiple systems. In this paper, we build on an existing taxonomy of DQ problems [17] to show how DQ assessment methods can be classified in this taxonomy. For each classified item, we discuss the applicability of the DQ method by giving an illustrative example from a practitioner's perspective. Moreover, the paper identifies the gaps in the taxonomy where no current automated DQ methods exist, which show potential pathways for further research on DQ assessment methods.

The rest of the paper is structured as follows: First, we introduce a classification of DQ problems and a list of DQ methods currently used in DQ tools. Furthermore, we provide a brief review of previous work on DQ method classifications. The centre of this paper is our proposed classification, which maps DQ assessment methods with DQ problems into a taxonomy and provides detailed examples for each mapping. The section is divided into mapping of independent DQ problems and mapping of context-dependent DQ problems. The paper ends with a conclusion and outlook for future research.

DATA QUALITY PROBLEMS

There are many types of DQ problems in organizations and several researchers have taken a closer look at DQ problems and root causes [17],[7],[13],[21]. DQ problems can be classified into problems that exist independently of a specific context, e.g. spelling errors and duplicate data, and problems that depend on the context of use, e.g. violation of company and government regulations [9]. Furthermore, these problems can be seen from a user perspective, which are the problems recognized by an information consumer, or a data perspective, which are often the root causes of the problems that information consumers have to face.

	Data Perspective	**User Perspective**
Context-independent	Spelling error Missing data Duplicate data Incorrect value Inconsistent data format Outdated data Incomplete data format Syntax violation Unique value violation Violation of integrity constraints Text formatting	The information is inaccessible The information is insecure The information is hardly retrievable The information is difficult to aggregate Errors in the information transformation
Context-dependent	Violation of domain constraints Violation of organization's business rules Violation of company and government regulations Violation of constraints provided by the database administrator	The information is not based on fact The information is of doubtful credibility The information presents an impartial view The information is irrelevant to the work The information is incomplete The information is compactly represented The information is hard to manipulate The information is hard to understand

Table 1. A Classification of DQ Problems [9]

As the paper is aiming at classifying DQ assessment methods in the context of relational databases, this

research focuses on the data perspective problems for both context dependent and independent categories and uses the classification of DQ problems identified in [5] (see Table 1). Hence, we provide a brief definition for each DQ problem in the context of our research in the following. In the context independent category, spelling errors, missing data, and incorrect values are self-explanatory DQ problems. Duplicate data problems occur when rows are duplicated or when schemas contain redundancies (that is, specify duplicate attributes in multiple databases). Data format problems occur when two or more semantically equivalent data values have different representations (this includes inconsistent and text formatting DQ problems). Syntax violation problems occur when a pre-specified format has been assigned to an attribute and a data value for this attribute does not adhere to this format (this includes the incomplete data format DQ problem in Table 1). Problems with violations of integrity constraints arise when data values do not adhere to pre-specified database integrity constraints; we also therefore include unique value violations, rather than have these as a separate problem, because unique value violations are one type of database integrity constraint. Note that, despite its position in Table 1, we treat outdated data to be a user perspective problem because whether data is out of date depends on the purpose it is used for.

For the context dependent category, the problem of violation of domain constraints is when an attribute value must be in a pre-specified context-dependent domain of values. Violation of organizational business rules is when any set of values do not adhere to a pre-specified rules assigned by the organization. Violation of company and governmental regulations is when any set of values do not adhere to a pre-specified rules assigned imposed on the organization by legislating bodies. Similarly, violation of constraints provided by the database administrator is when any set of values do not adhere to a pre-specified rules assigned by the database administrator.

DQ ASSESSMENT METHODS

DQ assessment is the process of "obtaining measurements of DQ and using these measurements to determine the level of DQ improvement required" [24]. Assessments can be viewed as a series of activities and for each activity a certain concrete method (referred to as a DQ method) is needed to carry out the activity. As noted before, the focus of this research is on the DQ methods that have been automated and are carried out in databases. Based on the existing body of knowledge in the literature and 10 years of practitioner's experience of current practices in the industry, we have created a detailed list of automated DQ assessment methods in databases, which is presented in the following in alphabetical order:

Column analysis typically computes the following information: Number of (unique) values and the number of instances per value as percentage from the total number of instances in that column, number of null values, minimal and maximal value, total and standard deviation of a value for numerical columns, median and average value scores, etc. [18]. In addition, column analysis also computes the inferred type information. For example, a column could be declared as STRING column in the physical data model, but the values found would lead to the inferred data type DATE. The frequency distribution of the values in a column is another key metric which can influence the weight factors in some probabilistic matching algorithms. Another metric is format distribution where only 5 digit numeric entries are expected for a column holding German zip codes.

Cross-domain analysis (also known as functional dependency analysis) can be applied to data integration scenarios with dozens of source systems [14]. It enables the identification of redundant data across tables from different, and in some cases even the same, sources. Cross-domain analysis is done across columns from different tables to identify the percentage of values within the columns indicating that they might hold the same data.

Data validation algorithms verify if a value or a set of values is found in a reference data set [16]. For manual data validation, a sample needs to be selected, whereas data validation which is automated can often work on the complete dataset. A typical example for automated data validation is address validation against a postal dictionary checking if a structural correct address actually exists. For data validation, good results depend on high-quality input data. For example, a validation against the postal dictionary will only produce good results if the address information has been standardized before.

Domain analysis can be applied to check if a specific data value is within a certain domain of values [18]. A domain can be a lookup table containing a series of values, some other pre-defined set of values or range conditions where values have to be within certain boundaries.

Lexical analysis is usually applied to columns containing STRING values with the intent to map unstructured content to a structured set of attributes. There are basically two major approaches: rule-based and supervised-model based techniques [4],[1]. Lexical analysis would, for example, identify "Christian" as a first name based on match against a dictionary, the gender of the first name to be "MALE" and "Müller" to be the last name. Furthermore, phonetic algorithms as, for example, NYSIIS or SOUNDEX [23],[19], can find out that the phonetic representation of "Müller" is equivalent to "Mueller". Lexical analysis is a technique often applied to name fields, address fields and text fields for product information where based on the parsed and tokenized and split content into several columns data standardization algorithms are then able to produce substantially better results from a DQ perspective.

Matching algorithms (also referred to as record-linkage algorithms) are used to identify duplicates such as two customer records that refer to the same customer [11]. To significantly reduce the runtime of matching, the "Sorted Neighbourhood Method (SNM)" has been introduced [12].

Primary key and foreign key analysis (PK/FK analysis) is applied to columns in two or more tables to detect whether or not the analyzed columns are good candidates for a PK/FK relationship which is not explicitly defined in the data model. For example, a technique for single and multi-column PK/FK discovery can be found in [15] and an efficient method avoiding limitations by implementing PK/FK discovery with a pure SQL approach can be found in [3].

Schema matching detects when two attributes are semantically equivalent. It offers help to the data modeller using appropriate tools using database schema matching algorithms, which are either schema-based matchers, instance-level matchers or hybrid approaches [22]. However, on large real-world data models with a few thousand or more attributes, current schema matching algorithms are often not very effective, although, additional improvements have been proposed in [5],[20].

Semantic profiling is used to express business rules on data for one or several columns in one more multiple tables and measures the compliance of the data against specified rules [6]. For example, a rule for the columns AGE and PROFESSION could be: IF AGE < 18 THEN PROFESSION='CHILD', which specifies that the attribute value for profession should be 'child' when the attribute value for age is less than 18.

EXISTING CLASSIFICATIONS OF DATA QUALITY METHODS

The requirement for organizations to assess and improve their DQ, has led to similar research efforts that aim to guide organizations in their selection of DQ tools. For instance, related research specifies useful criteria for the selection of DQ tools based on the functionality of the tools [10]. However, this work does

not indicate the specific DQ problems which are addressed by these tools. Our research differs in this respect, and one of the dimensions of our analysis is the DQ problems and how each DQ method addresses them. In the same subject area, Gartner research takes a different approach and provides a "Magic Quadrant" to guide organizations in their selection of DQ tools. The quadrant indicates which tools fit into the following categories: market leaders, challengers, niche players and visionaries [8]. In a survey of DQ tools, Barateiro and Galhardas divide DQ tools by their general functionality (e.g. debugging capabilities, the ability to extract from different data sources etc.) and also specify whether the tools are capable of finding problems that relate to multiple or single rows [2]. In this latter sense, the work is based partly on an existing taxonomy of DQ problems [17] and our research extends this to specify how DQ methods (rather than specific DQ tools) fit into all the elements of the taxonomy. Note that this abstraction from DQ tools to DQ methods is important because DQ tools, being software based, are likely to change with different releases of the software and may encompass any number of DQ methods in one tool. In the next section, we will present a detailed classification of DQ assessment methods.

CLASSIFYING DQ ASSESSMENT METHODS

The existing taxonomy of DQ problems consists of elements at various levels of granularity [17]. These levels relate to the well-known relational database structure which includes: attributes (fields or columns), rows (records or tuples), tables (or relations) and the database (multiple tables). Furthermore, the taxonomy also includes a level that relates to multiple databases. The different elements of the taxonomy are shown in Table 2, which includes the taxonomy element (an acronym of the element), the name of the element, and a mapping requirement. The mapping requirement (not described in the taxonomy) is used for this work to aid the mapping of the DQ methods into an element of the taxonomy for a particular DQ problem. This mapping requirement specifies what the DQ method must meet in order to find a particular DQ problem. For example, the domain analysis method only needs to consider whether the value of one attribute lies in the domain (which is external information) in order to determine whether there is an incorrect value, thus, it is classified as SAST (see the first row of Table 2) for the incorrect value DQ problem.

Taxonomy Item	***Name***	***Mapping Requirement***
SAST	Single Attribute Single Tuple	One attribute to be compared to external information
SAMT	Single Attribute Multiple Tuples	Comparing multiple rows using one attribute
MAST	Multiple Attributes Single Tuple	Comparing multiple attributes in one row
SR	Single Relation	Comparing multiple attributes between multiple rows in a single relation
MR	Multiple Relations	Comparing multiple attributes between multiple rows in multiple relations, for instance, by using the primary/foreign key links between relations.
MDS	Multiple Data Sources	Comparing data from different sources, e.g. multiple data bases, possibly with different data schemas and semantics

Table 2. Classification Mapping Requirements

Note that the domain checking method can be also applied to all rows in a table (by applying the method multiple times), but the algorithm requires only a single row and single attribute to judge if there is a DQ

problem at one given time. As this method does not use multiple rows, as required by SAMT, to reach a decision about DQ, it does not fulfil the SAMT mapping requirement. However, some data quality methods can meet the requirements of multiple taxonomy items depending on the type, context and DQ problem they address, e.g. semantic profiling can be used on single attributes and tuples, but also on multiple relations and data sources. The classification developed in this paper therefore includes both the elements of the taxonomy and DQ problems to demonstrate what problem the DQ method addresses and at what level the DQ method needs to be implemented (taxonomy element). In line with the division of DQ problems into context-independent and context-dependent (see Table 1), one classification for each division has been developed. The following two subsections describe these classifications.

Classification of Assessment Methods and Context-Dependent DQ Problems

Table 3 shows the results of the classification for the context-independent data perspective DQ problems with the taxonomy elements across the top and the DQ problems shown vertically. Each DQ method is placed in the relevant cells with a reference number in brackets.

When no example DQ problem could be found for a particular element, the cell is coloured grey. In some of these cases, the elements of the taxonomy for a given DQ problem may not be relevant, for example, a spelling error only relates to a single attribute. If an example DQ method has been found, then the cell is populated with the relevant method(s). Otherwise, cells are identified as either a gap and this indicates that a DQ example problem does exist, but no methods are currently available to address this problem.

Opportunities for new DQ Methods

Five gaps and one partial gap were identified from the classification and each gap is described below.

Gap 1: Instances of this problem occur when, for example, an organization records an item of stock as a row in a table (the full table is the inventory of stock). Items of stock may be physically present in the warehouse but do not appear in the table as a row. We consider this a gap since such scenarios cannot be detected automatically – however, we also think it will be impossible for some of these scenarios to develop methods at all. In this category, we believe there are scenarios which are only discoverable by human intervention.

Partial Gap in (16): This has been identified as only a partial gap because there are other DQ methods that can be applied to solve incorrect value problems in multiple relations. However, it is difficult today to detect homonyms in the domain value area. For example, when transferring data, the source system could use, for the marital status field, the values: 2 = married and 4 = single and the target system would use for the marital status field 2 = single and 4 = married. Then if the records are moved from the source system with the value 4 in the marital status without replacing it before loading it to the target with the value 2 (the semantically equivalent in the target), the person would get married while data is moved from one system to the next. The challenge here is that the description fields for a domain value set are typically text fields not allowing a seamless detection of what is semantically the same between the source and target system domain value sets. The detection of these incorrect values is thus difficult today in an automatic manner (but often trivial if a human being is looking at the domain value sets and their descriptions).

Gap 2: The detection of synonyms is not in general possible with algorithms today. Custom logic using data integration capabilities of the commercial tools needs to be built into projects to search for synonyms. Some enterprises started to build enterprise data dictionaries using ontologies. In such cases, a classification of business terms often includes a list of synonyms for a term if applicable. The list of available synonyms for a term as well as manually build synonym tables can be used by custom-built ETL logic to search for synonyms. Similarly, homonyms cannot be detected in general with algorithms

and again data integration capabilities are often required to build the custom logic for this.

Gap 3: A possible example (albeit almost artificial and a consequence of poor data modelling) could be something like a table containing a column "PROFESSION" and a column "JOB CATEGORY" with entries such as:

- PROFESSOR, TEACHER
- DEVELOPER, IT
- ARCHITECT, IT
- TEACHER, TEACHER
- ARCHITECT, CONSTRUCTION

In this case, the homonym 'Architect' could only be detected by looking across two columns. Again, today there are no known methods to automatically detect homonyms in all cases.

Gap 4 and Gap 5: A common area for synonyms and homonyms to occur is in the domain of reference data management which is typically stored in lookup tables. For the synonym problem, the same semantic set of country codes in a lookup table one system might use an integer-based value set for the lookup values whereas another system might use two-letter values. As a result, the country Germany might be represented with the value '62' in one system and 'DE' in another system indicating that these values are actually synonyms. Since in many cases enterprise applications are customizable regarding lookup tables and their values, a company using multiple instances of the same application might suffer substantially by different customizations of lookup tables per deployed application instance. As seen in customer situations, for roughly 200 countries in such a scenario several thousands different lookup values were customized in several instances of the same application. Not surprisingly, in the data warehousing environment where the operational data from all applications came together, revenue reports by country did not show meaningful results anymore. Similarly assigning the same lookup value across systems to different meanings can also cause substantial problems.

Discussion of the Context-independent Classification of DQ Methods

For each of the DQ methods in Table 3, the reason for classifying these methods into the particular cells is described. The numbers in braces in the table are used to give reference to the descriptions.

(1) Using lexical analysis on a single attribute, it is possible to determine whether "Martttin" is not a known first name in accordance with the dictionary used by the algorithm.

(2) A column analysis can be used to compare spellings for single attributes over all rows to identify misspellings by examining inconsistencies between rows. For example, in a table of product data, the value "fatscreen" appears once in the product name column while the majority of other rows contain the value "flatscreen" for this attribute.

(3) By specifying that "null" is not a permissible value in the domain of an attribute, domain analysis can be used on a single attribute in a single row to detect missing data.

(4) Column analysis can be extended from a single row (see 3) to identify the total number of missing values for an attribute in all rows and produce a report indicating, for example, that a "null" value appears a 100 times out of 200,000 rows.

(5) It is possible to define semantic rules, such as "If (AGE > 18) THEN (PROFESSION != NULL)" indicating that a person of age 18 must have a non-null value in the profession column. Violation of such rules indicate that values are missing in a single row.

Data Perspective (Context-independent) – Assessment Methods						
DQ Problems	***SAST***	***SAMT***	***MAST***	***SR***	***MR***	***MDS***
Spelling error	(1) Lexical Analysis	(2) Column Analysis				
Missing data	(3) Domain Analysis	(4) Column Analysis	(5) Semantic Profiling	**Gap 1**	(6) Semantic Profiling	(7) Semantic Profiling
Duplicate data		(8) Column Analysis	(9) Semantic Profiling	(10) Matching algorithms	(11) Matching Algorithms, Cross-domain analysis Schema Matching	(12) Matching Algorithms, Cross-domain analysis, Schema Matching
Incorrect value	(13) Lexical analysis, Column analysis, Domain analysis, Data Validation	(14) Column Analysis, Semantic Profiling	(15) Semantic profiling, Data validation		(16) Semantic Profiling, PK/FK analysis, Column Analysis, **Partial gap**	(17) Semantic Profiling, Domain Analysis, Column Analysis
Inconsistent data format		(18) Column Analysis	(19) Column analysis, Semantic Profiling		(20) Column analysis, Semantic Profiling	(21) Column analysis, Semantic Profiling
Syntax violation	(22) Column Analysis					
Violation of integrity constraints	(23) Domain Analysis	(24) Column Analysis	(25) Semantic Profiling	(26) Column Analysis	(27) PK/FK Analysis	(28) Semantic Profiling, Domain Analysis
Text Formatting	(29) Lexical Analysis					
Existence of Synonyms/ Homonyms		**Gap 2**	**Gap 3**		**Gap 4**	**Gap 5**

Table 3. Mapping of assessment methods and context independent DQ problems

(6) A typical customer object in an ERP systems could consist of multiple tables in a hierarchical table structure. Sometimes, in a business to business scenario, the customer object in the ERP system is customized so that for each enterprise customer there has to be at least one contact person assigned. The core customer information in the table is representing the root node of the table hierarchy and the contact person would be in a table being a child node of the root node. Using semantic profiling across the multiple relations (the parent and child table in this case), it is possible to check if for each enterprise customer record in the parent, there is at least one entry in the child table storing the contact person information. If no contact person is found, the semantic rule would flag this as a missing record in this multiple relations case.

(7) Assume you have customer information in one data source (MDM system) and assume there is another data source supporting the order entry system. A semantic rule in this case might validate if for each active customer in MDM there is at least one order in the order entry system placed in the last 2 years. Otherwise you might want to mark the customer as inactive.

(8) Column analysis can be used to create a frequency distribution counting if there are two or more occurrences of values to identify duplicates. The description of the data model for an attribute might indicate an uniqueness constraint (in database terms a UNIQUE INDEX or a PRIMARY KEY). Using column analysis, its possible to detect whether or not the values across all records for this attributes are unique and thus satisfy this requirement.

(9) In semantic profiling expressions like *"IF a1 = a2 AND a2=a3 AND … THEN record exception"* (where a1, a2, a3 are attributes) can be specified to discover if multiple columns in the same row have the same value to find duplicates. In some cases, users enter dummy data like "abc" in all mandatory fields to pass validation checks when they don't have the information available.

(10) For MDM projects, de-duplication of employee, supplier, customer, product, account, etc. records in the database tables is a requirement. Probabilistic matching techniques are a commonly used assessment technique, which work on multiple attributes and rows to check for matches.

(11) Similar to (9) master data is a typical use case in many application systems because a business object like customer consists of multiple relations. Cross-domain analysis is used across columns of different tables to identify the percentage of same data, which might be an indication that two different tables are duplicates. Schema matching can be used to detect when attributes in multiple tables are redundant.

(12) Applying data matching (i.e. record linkage) to identify the appropriate records in an enrichment data set correlated with the records in a dataset. The matching step is a necessary pre-requisite before the actual enrichment step can be done. Cross-domain analysis is useful to identify if there is possible data replication among data systems, i.e. whole datasets are redundant. Schema matching, similarly to (11) above, can be used to detect when attributes or whole tables in multiple data sources are redundant.

(13) If lexical analysis is applied to a CHAR(150) field for the address "Hirschkopfstrasse 13 71149 Bondorf 07457948953" it might report that "07457948953" as a token cannot be recognize as an element of an address and is thus an incorrect value. A column analysis report might show the value "abc" in a column, which according to the metadata should be of data type INT, and, thus, is obviously a wrong value in that column. If we have a column supposedly containing titles ("Mr.", "Mrs.", "Dr.", "Prof.",) and a domain analysis would find the value "Herr", it would report this value to be incorrect because it is not part of the valid domain values. Another example of domain analysis could be a data integrity constraint on a column storing AGE information that the permissible range of values is > 0 and < 140 so that any value outside this range is flagged as incorrect. Data validation might find out that a syntactically correct value might not be correct since it does not exist in an external data dictionary.

(14) There might be a business decision that the products should be numbered sequentially by a product number increased by a fixed value or fixed pattern (e.g. 1, 2, 3, 4, 5, … or 000010, 000020, 000030, 000040, …). Using column analysis or semantic rules can help to discover if data complies to such a number scheme.

(15) An example for semantic profiling could be a rule like: IF (GENDER = MALE AND RELIGION = ROMAN-CATHOLIC AND PROFESSION = PRIEST THEN MARITAL STATUS = SINGLE).
Data validation helps to detect whether or not syntactically correct data is valid. For example, if we have: STREET = OXFORD STREET, HOUSE NUMBER = 185, ZIP CODE = W1D 3DG, CITY = LONDON and COUNTRY= UK then this would be a syntactically correct address. Performing data validation against the UK postal address dictionary would flag that all values are correct except for the HOUSE NUMBER for which number 185 does not exist because the largest number in Oxford Street is 157.

(16) It is possible to check business rules across tables using semantic profiling. For example, a semantic rule could be used to ensure that each sales employee is associated with at least one customer. Sometimes a primary/foreign key relationship exists between two tables, which is not mentioned in the data model documentation nor made explicit in the metadata stored in the database dictionary, because the FOREIGN KEY constraint has not been created as part of the SQL DDL statements. Primary/foreign key analysis can be used to detect if there is one column, between the tables, which would qualify as a primary key foreign key. This column may contain incorrect values requiring correction, however. In this case the database system can be used to automatically enforce the integrity constraints to detect the incorrect values. Finally, the format analysis part of column analysis might expose that the syntax of data fields is not consistent across tables (e.g. 12/31/2010 in one table and 31/12/201 in another table).

(17) Some of the examples for (15) like semantic rules across multiple tables can also be used in a scenario where the tables come from multiple sources. Additionally, semantic profiling might be used if complementary data sources are involved to check the correctness of data. Domain analysis against the domain value set of the target system for a field might expose that none of the source systems is using the same domain values for this field. For instance, without transcoding “DE” to “62” while moving this record from the source system 1 to the target system, the target system would be loaded with a data quality defect, since it is not understanding the value “DE”, being not part of its domain value set for that field. Similarly if source system 2 would use “GER” for Germany, a different transcoding rule needs to be implemented to map “GER” to “DE” for the records coming from source system 2.
Column analysis might be used to detect different formats across multiple sources for dates, etc.

(18): Inconsistent formats, e.g. a German phone number should be compliant with the format of 0049+area code+phone number, can be detected using column analysis, similar to the previous examples.

(19): A table might contain contact information with multiple fields for phone numbers such as a field for office phone, mobile phone and home phone. If someone uses just one mobile phone and wants to be contacted only with the mobile phone number, then the values are semantically equivalent and should appear in the same format. Column analysis or semantic profiling using for example regular expressions might be used to detect whether or not the format is applied consistently across the columns.

(20) and **(21)**: In both cases, it could be that tables in the same or multiple data sources store similar information like price information, dates, etc. For example, there might be a table (maybe in the product development system) with product prices with a field for price which is a DECIMAL(31, 5). In an order item table (maybe an e-Commerce system) containing the individual items of an order the price field might be a fixed-length STRING(32) with leading, padding whitespaces if the price does not have 31 digits. This and similar issues could be detected with column analysis and/or semantic profiling

techniques.

(22) Inconsistent syntax for dates can also be detected using column analysis, e.g. 12/31/2010 instead of 31/12/2010.

(23) Domain analysis helps to determine if all values for an attribute are within an allowed range (e.g. col1 > 0 AND col1 < 100). Similar, using domain analysis, you can check if a value in a field is in a given value set of the lookup table supporting this field.

(24) Column analysis can be used to detect if all the values in a column are unique.

(25) With semantic profiling business constraints across multiple attributes can be detected. An example could be: selling price = base price * sales tax.

(26) Column analysis can be applied to a set of fields to detect whether or not they violate a composite primary key constraint which requires unique value combinations across all records in the fields of the composite key.

(27) Primary/Foreign key analysis can be used to detect if two tables in a parent-child relationship comply with the Primary/Foreign Key constraint for the key columns.

(28) Semantic profiling with rules can be used to detect if compliance exists for integrity constraints of data across sources. For example, a rule can check between data tables from a sales territory management system and tables from the human resources system indicating that only seasoned sales employees, which work at least for 3 years in sales, should have been assigned to customer accounts in the platinum customer segment.
During data migration from a source system to a target system, a domain analysis for source records after transcoding can discover if the transcoding has been configured correctly so that the transcoded values indeed are all known by the target system.

(29) Lexical analysis is able to discover if all tokens in a text are recognized as patterns in a specific format of a domain like addresses, products, etc. Lets assume an attribute street with the following string: 274 St. John St. Lexical analysis in the parsing step would identify 4 tokens: ‘274’, ‘St.’, ‘John’ and ‘St.’. In the lexical investigation, it classifies the tokens as follows: ‘274’ is identified to be a number, ‘St.’ is identified as a possible street type, ‘John’ is identified as an alphanumeric string possible being a unique name and ‘St.’ is again identified as a possible street type. In the final step applying context sensitive interpretation of the lexical investigation, the conclusion would be that ‘274’ is the house number, ‘St. John’ would be the street name and the last ‘St.’ must be the street type indicator. The output of the lexical analysis would be in addition to the input column street three columns indicating house number, street name and street type and the string would have not produced any unhandled pattern (a token which the lexical analysis would flag as not understood based on available dictionaries and rules). With the additional three columns applying address standardization is now able to produce much better results.

Classification of Assessment Methods and Context-Dependent DQ Problems

Table 4 presents the results of the classification for the context-dependent data perspective DQ problems with the taxonomy elements across the top and the DQ problems shown vertically. However, in contrast to the context-independent classification, no gaps were identified this time. Each item in the table is discussed in the next section in further detail.

Data Perspective (Context-dependent) – Assessment Methods						
DQ Problems	***SAST***	***SAMT***	***MAST***	***SR***	***MR***	***MDS***
Violation of domain constraints	(28) Domain Analysis		(29) Domain Analysis, Data validation			
Violation of organization's business rules		(30) Semantic Profiling	(31) Semantic Profiling	(32) Semantic Profiling	(33) Semantic Profiling	(34) Semantic Profiling
Violation of company and government regulations	(33a) Lexical analysis	(33b) Column Analysis	(33c) Semantic profiling, Data validation	(33d) Matching algorithms	(34) Semantic profiling	(33e) Data validation
Violation of constraints provided by the database administrator	(35) Column Analysis, Domain Analysis	(36) Column Analysis	(37) Semantic Profiling	(38) Semantic Profiling	(39) PK/FK analysis, Cross-Domain Analysis	(40) Cross-domain Analysis

Table 4. Mapping of assessment methods and context dependent DQ problems

Discussion of the Context-dependent Classification of DQ Methods

(28) Context-dependent constraints are many-fold. For example, a company might sell in North-America a certain product with the permissible domain value range for the product colours of red, green and blue. However, if the same product is sold in Europe, the permissible range of domain values for the colour of the product might only be yellow, white and black.

(29) Using domain analysis the measured percentages of product ingredients during production can be assessed against the permissible ranges of each ingredient, An example for data validation would be to validate standardized address information against a postal address dictionary.

(29a) Packaged enterprise applications often allow to group customers by account group such as "enterprise customer", "one-time customer", etc. For each account group, there is typically a different numbering scheme representing the different context of the account group. The numbering scheme by account group can implement a number of different characteristics. For example, for one account group, the valid identifiers which can be used must be between 100.000 and 200.000 with increments of 1 starting at 100.000 whereas for another account group the range of valid identifiers is between 600.000 and 900.000 with increments of 5. Thus, to decide if the customer identifiers are valid the values need to be assessed in the context of the account group.

(29b) to **(32)**: For these cases, there are many project specific semantic rules possible. The difference to previous mentioned examples is that there are now multiple rules for the same basic data quality problem where the actual rule that is used depends on the given context.

(33a-e) For these parts of the taxonomy, examples are used to explain the use of methods in the Violation of company and government regulations row of Table 4. The Office of Foreign Assets Control (OFAC, see http://www.treasury.gov/about/organizational-structure/offices/Pages/Office-of-Foreign-Assets-Control.aspx) of the Department of the US Treasury in the US publishes and maintains a list of foreign countries, organizations and persons (e.g. terrorists) with whom US citizens are not allowed to do business with. Compliance with OFAC is required across all industries – otherwise possible fines might be placed on violators. To detect OFAC compliance can be done by performing data validation (33e) on customers, citizen and account records to verify none is on the OFAC list. The OFAC list would be another data source in addition to the in-house systems storing customer, citizen and account information. Another example of a legislation requiring data validation (33e) against another data source is The Do-Not Call Implementation Act of 2003 (see http://www.ftc.gov/bcp/edu/microsites/donotcall/index.html) in the US affecting customer and citizen data. The main purpose of this legislation is to allow US citizens to opt-out from telemarketing. Now in order to get good results for data validation in both examples in (33e), lexical analysis (33a) including support for cultural name variations in spelling might be applied to see if any first name is misspelled requiring possible correction. For compliance with the Do-Not Call Implementation Act of 2003 the Social Security Number (SSN) which is supposedly unique analyzed with Column Analysis (33b) and data validation (33c) might be applied on standardized addresses to see if the addresses actually exist. Then matching (33d) is applied to identify and reconcile if necessary duplicate customer records before doing the validation against the Do-No Call reference list.

(34) The Bank Secrecy Act of 1970 in the US (see http://www.fincen.gov/statutes_regs/bsa/) is among the first legislations with the intent to detect criminal activities in the financial and retail industry such as money laundering. This legislation for example requires companies to detect cash transactions of 10.000 US or more. To detect such events, semantic rules on transaction records need to be applied searching for transaction type='CASH' and AMOUNT>=10.000. Similarly, rules are used to detect criminal misuse of stolen credit card information credit card companies try to detect fraudulent credit card transactions. For example, if a credit card has been used for a payment in Pittsburgh, USA at 2 pm on June 14th, its impossible that the same credit card can be used only two hours later in Sydney, Australia since the owner has no means to get there in any way in just two hours. Thus, the second transaction in Sydney would be mark as suspicious.

(35) A database administrator might know due to having access to the data model that the values in a column should be in the domain of 1 to 100. This can be easily checked using domain analysis. Column analysis can be used to verify if a format for the values is actually complied with.

(36) Column analysis can be used to verify if the data in a column complies with a uniqueness constraint.

(37) and **(38)**: Depending on the business context, semantic profiling might be used to verify constraints set by a database administrator

(39) Improving data accuracy by improving the physical data design with index structures might lead a database administrator to search for good PK/FK candidates using PK/FK analysis. Reference data compliance across multiple tables can be measured with a domain analysis, where the domain is given by a lookup table. Country code, unit of measure, etc. are typically reference data tables used by business objects such as customer, supplier or product.

(40) A new database administrator assigned to a couple of database systems with the task to reduce TCO

with no documentation available might use cross-domain analysis to identify data redundancy across tables in the same or multiple databases. Removing data redundancies reduces storage requirements and thus reduces TCO.

CONCLUSION AND FUTURE WORK

The technological progress in information technology over past decades has provided organizations with the capability to collect and process vast amounts of data, which has been used to automate and innovate business processes and to make more informed operational and strategic decisions. However, the amount of data available has brought new challenges as data can only be as good as its quality allows. Automated DQ methods provide efficient ways to assure that the quality of data in large company-wide information systems is fit for the purpose of its use. This paper has analyzed DQ methods that are available to automate DQ assessment. We have shown where and how DQ assessment methods can be helpful and are applicable by providing a comprehensive classification that maps DQ assessment methods to the taxonomy of DQ problems. We have illustrated and discussed the use of the DQ assessment methods by giving detailed practical usage examples for each mapping in our classification. Moreover, the aim of this research was to use this classification to identify gaps where no existing methods exist for particular DQ problems and, thus, where new methods are required.

Several gaps were identified overall for two DQ problems: missing data and existence of synonyms and homonyms. The first gap relates to the missing data DQ problem, where for example a table is missing a row. In this case the whole relation is needed to determine if any rows are missing (all rows need to be checked) as well as external knowledge of the entity that the missing row(s) should represent. Instances of this problem occur when, for example, an organization records an item of stock as a row in a table (the full table is the inventory of stock). Items of stock may be physically present in the warehouse but do not appear in the table as a row. Currently, there are no automated methods to detect the challenging problem of missing data in this and similar contexts. The other gaps relate to the detection of synonyms and homonyms throughout multiple rows, multiple attributes, multiple relations and multiple data sources. For homonyms, this problem often occurs when inconsistent coding schemes are used for domain values in different databases, such as assigning the code ‘2’ and ‘4’ to ‘married’ and ‘single’ attribute value options in one database and a different coding scheme in other databases. Clearly, any data that is moved between these databases changes its value inadvertently. New solutions are required to tackle this interesting, but also difficult, problem.

One limitation of the classification is that it only considers whether a DQ method addresses a DQ problem (for each taxonomy element), and this may not always be absolute in the sense that some DQ methods may be more comprehensive than others—certain DQ methods will have limitations in their use such as execution performance and these have not been captured in the final classification. Furthermore, for the grey coloured cells in the classification, that indicate when an example DQ problem could not be found, it may be the case that only the developers of the classification were not able to identify a DQ problem. It is hoped that exposure of the classification to a wider expert audience will help identify any omissions such as these.

This paper has focused on classifying DQ assessment methods. The identified gaps for DQ assessment methods offer new challenges for DQ researchers and might provoke the development of innovative new methods. Future work could extend the classification with DQ improvement methods. Furthermore, as semi-structured data like XML files and unstructured data, e.g. power point presentations, are becoming increasingly important, an analogous classification for these data types is also needed. Lastly, this paper addressed DQ problems specifically from a data perspective and a similar classification designed from a user perspective could provide new valuable insights in the future.

ACKNOWLEDGEMENTS

This research has been partly funded by EPSRC project "Information Quality in Asset Management", reference number EP/G038171/1.

REFERENCES

[1] Agichtein, E., and Ganti, V., "Mining reference tables for automatic text segmentation," Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp.20-29.

[2] Barateiro, J., and Galhardas, H., "A survey of data quality tools," *Datenbank Spectrum*, 14, 2005, pp.15–21.

[3] Bauckmann, J., Leser, U., and Naumann, F., "Efficiently Computing Inclusion Dependencies for Schema Discovery," *In Second International Workshop on Database Interoperability, ICDE 06*, 2006.

[4] Borkar, V., Deshmukh, K., and Sarawagi, S., "Automatic segmentation of text into structured records," ACM SIGMOD Record, 2001, pp.175-186.

[5] Byrne, B., Fokoue, A., Kalyanpur, A., and Srinivas, K., "Scalable Matching of Industry Models – A Case Study." in *Proceedings of the Fourth International Workshop on Ontology Matching (OM 2009)*, Washington D.C., USA, 2009.

[6] Chiang, F., and Miller, R.J., "Discovering data quality rules," *Proceedings of the VLDB Endowment*, 1 (1), 2008, pp.1166–1177.

[7] Eppler, M., *Managing information quality: increasing the value of information in knowledge-intensive products and processes*. Springer-Verlag New York Inc, 2006.

[8] Friedman, T., and Bitterer, A., "Magic quadrant for data quality tools," Gartner RAS Core Research Note G00167657, 2007.

[9] Ge, M., and Helfert, M., "A Review of Information Quality Research," Proceedings of the 12th International Conference on Information Quality, 2007.

[10] Goasdoué, V., Nugier, S., Duquennoy, D., and Laboisse, B., "An Evaluation Framework for Data Quality Tools," Proceedings of the International Conference on Information Quality, 2007.

[11] Gu, L., Baxter, R., Vickers, D., and Rainsford, C., "Record linkage: Current practice and future directions," *CSIRO Mathematical and Information Sciences Technical Report*, vol. 3, p. 83, 2003.

[12] Hernández, M.A., and Stolfo, S.J., "The merge/purge problem for large databases," Proceedings of the 1995 ACM SIGMOD international conference on Management of data - SIGMOD '95, 1995, pp.127-138.

[13] Huang, K.T., Lee, Y.W., and Wang, R.Y., *Quality Information and Knowledge Management*. 1st ed., Prentice Hall, 1999.

[14] Lenz, H.J., and Shoshani, A., "Summarizability in OLAP and statistical data bases," Proceedings of the Ninth International Conference on Scientific and Statistical Database Management, 1997, pp.132-143.

[15] Li, F., Hadjieleftheriou, M., Kollios, G., and Reyzin, L., "Authenticated Index Structures for Aggregation Queries in Outsourced Databases" 2006.

[16] Maydanchik, A., *Data Quality Assessment*. Technics Publications LLC, 2007.

[17] Oliveira, P., Rodrigues, F., and Henriques, P., "A formal definition of data quality problems," Proceedings of the 10th International Conference on Information Quality, 2005, pp.13–26.

[18] Olson, J.E., *Data quality: the accuracy dimension*. Morgan Kaufmann, 2003.

[19] Patman, F., and Shaefer, L., "Is Soundex good enough for you? On the hidden risks of Soundex-based name searching," Language Analysis Systems, Inc., Herndon, 2001.

[20] Peukert, E., Eberius, J., and Rahm, E., "AMC - A framework for modelling and comparing matching systems as matching processes," 2011 IEEE 27th International Conference on Data Engineering, 2011, pp.1304-1307.

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

[22] Rahm, E., and Bernstein, P.A., "A Survey of Approaches to Automatic Schema Matching," *VLDB journal*, 10, 2001, p.2001.

[23] Taft, R.L., "Name search techniques," Bureau of Systems Development, New York State Identification and Intelligence System,, 1970, .

[24] Woodall, P., and Parlikad, A., "A Hybrid Approach to Assessing Data Quality," Proceedings of the 2010 International Conference on Information Quality, 2010.