Universität Stuttgart
Germany

IPVS

Institute of Parallel and Distributed Systems
Department of Applications of Parallel and Distributed Systems

Universität Stuttgart
IPVS
Universitätsstraße 38
D-70569 Stuttgart

Master Thesis Nr. 0990-0004

Measurement of the quality of structured and unstructured data accumulating in the product life cycle in a data quality dashboard

Shalini Chellathurai Saroja

Course of Study:	Master of Science Information Technology
Examiner:	Prof. Dr.-Ing. habil. Bernhard Mitschang
Supervisor:	M.Sc. Cornelia Kiefer
Commenced:	29 Aug 2016
Completed:	10 Feb 2017
CR-Classification:	H.2.8, H.3.5, I.2.7.

ABSTRACT

This thesis provides an overview on existing data quality metrics for structured and unstructured data as well as on the existing data quality dashboards for measuring the quality of structured and unstructured data. Open research questions for interpreting the data quality are discussed. The metrics *percentage of null values, percentage of duplicate values* and *percentage of non-domain values* were selected and implemented as REST based web services. Furthermore, a web application was developed to enable (1) upload of the data file for which data quality shall be assessed from two standard formats JSON and CSV and (2) flexible integration of various data quality metrics. The latter is enabled by using an interface. To illustrate the functionality of this interface, the metric *percentage of spelling mistakes* provided by the supervisor of the thesis is integrated with the web application. The data quality is indicated as percentage in the range from 0 to 100 as well as encoded with colors for the whole dataset and for each column. Donut chart or pie chart visualizations are implemented for the chosen data quality metrics. The implemented web application and metrics were evaluated with the example datasets for data accumulating in the product life cycle as provided by the supervisor. Finally, the dashboard is compared with existing data quality dashboards and the results are tabulated.

ACKNOWLEDGEMENTS

I thank Prof. Dr.-Ing. habil. Bernhard Mitschang for providing me an opportunity to work on my thesis in the Department of Applications of Parallel and Distributed Systems.

I express my deepest gratitude to my supervisor M.Sc. Cornelia Kiefer for guiding, motivating and supporting me throughout the entire duration of the thesis. I am extremely fortunate to work with her.

I thank my husband Pradeep Parameshwaran and my family for encouraging and supporting me throughout the thesis.

TABLE OF CONTENTS

ABSTRACT i
ACKNOWLEDGEMENTS iii
TABLE OF CONTENTS v
LIST OF FIGURES ix
LIST OF TABLES xi
LIST OF SCREENSHOTS xiii
ABBREVIATIONS xv
1 INTRODUCTION 1
1.1 Research Question 1
1.2 Thesis Organization 2
2 BACKGROUND 3
2.1 Data 3
2.2 Data Quality 4
2.3 Data Quality Dimensions 4
2.3.1 Accuracy 4
2.3.2 Completeness 5
2.3.3 Validity 5
2.3.4 Consistency 5
2.3.5 Currency 6
2.3.6 Interpretability 6
2.3.7 Relevancy 6
2.4 Product Life Cycle 6
2.5 Data Visualization 6
2.6 RESTful Web Service 8
3 RELATED WORK 9
3.1 Existing Data Quality Dashboards 9
3.1.1 IBM Watson Analytics 9
3.1.2 Data Analyzer by Uniserv 10
3.1.3 Data Quality Scorecard by Uniserv 11
3.1.4 Data Quality Dashboard and Reporting by Informatica 11
3.1.5 Data Quality Analysis Dashboard Package by Salesforce 12
3.1.6 Data Quality Dashboards by Talend 13
3.1.7 DQ Dashboard by Attacama 13
3.1.8 Data Quality Dashboard by InsightSquared 13
3.1.9 Data Governance Center 4.5 by Collibra 14

3.2 Comparison of Data Quality Dashboards 16
3.3 Prototypes of Data Quality Dashboards 17
3.4 Open Research Issues 18
4 DATA QUALITY METRICS 19
4.1 Overview on Data Quality Methodologies 19
4.2 Assessing the Quality of Unstructured Data 20
4.3 Completeness 22
4.3.1 Percentage of null values 22
4.3.2 Percentage of default values 23
4.3.3 Percentage of duplicate values 23
4.4 Validity 24
4.4.1 Percentage of non-domain values 24
4.4.2 Percentage of non-range values 24
4.4.3 Percentage of outliers 25
4.5 Consistency 25
4.5.1 Percentage of inconsistent format in a field 25
4.6 Interpretability 26
4.6.1 Percentage of spelling mistakes 26
4.6.2 Fit of training data 26
4.7 Relevancy 26
4.7.1 Percentage of non-relevant data 26
5 CONCEPT 29
5.1 Architecture 29
5.1.1 High Level Generic Architecture 29
5.1.2 High Level Specific Architecture 29
5.1.3 Low Level Architecture 31
6 IMPLEMENTATION 33
6.1 Libraries 33
6.1.1 AngularJS 33
6.1.2 Data Driven Documents (D3) 33
6.1.3 NVD3 33
6.1.4 Angular-nvD3 34
6.1.5 Bootstrap 34
6.1.6 Apache Tomcat 34
6.1.7 CherryPy 34
6.1.8 Jersey 34

6.1.9 JSON-Simple 34
6.1.10 Duplicate Detection Toolkit 34
6.1.11 PyEnchant 35
6.2 Implementation Details of the Web Front End 35
6.2.1 Upload Module 35
6.2.2 Metric Modules 36
6.2.3 Metric Interface 36
6.2.4 User-defined Calculation of Overall DQ 37
6.2.5 Visualizations 37
6.3 Instructions to Integrate New Metrics 38
6.4 Web Services 38
6.4.1 Null Value Web Service 38
6.4.2 Duplicate Value Web Service 39
6.4.3 Non-Domain Value Web Service 39
6.4.4 Spelling Mistakes Web Service 40
6.4.5 Convert to JSON file Web Service 40
6.5 User Interface 40
7 DEMONSTRATION AND EVALUATION 43
7.1 Datasets 43
7.2 Analysis of the Quality of the Datasets with the Dashboard 43
7.2.1 Analysis of the NHTSA Consumer Complaints Dataset 44
7.2.2 Analysis of DuDe Restaurant Dataset 48
7.2.3 Analysis of Twitter Dataset 48
7.2.4 Analysis of News Dataset 48
8 CONCLUSION AND FUTURE WORK 51
9 APPENDIX 53
REFERENCES 57

LIST OF FIGURES

Figure 1: Data in the Product Life Cycle [9] 7
Figure 2:Watson Analytics - Data Quality Score for each column [25] 10
Figure 3:Informatica-Threshold of conformance [29] 12
Figure 4:Informatica-Historical level of conformance [29] 12
Figure 5:Collibra metric configuration [37] 14
Figure 6:Data Governance Center 4.5 - Validation Script [38] 14
Figure 7: Data Governance Center 4.5 - Data Quality Dashboard [39] 15
Figure 8: Data Goverenance Center 4.5 - Details Pane [39] 15
Figure 9:Dimensional Assessment of IQ across Information Providers [17] 18
Figure 10:Phases of TDQM Methodology [1] 20
Figure 11: Indicators for measuring the quality of unstructured text data [2] 21
Figure 12:High Level Generic Architecture Diagram 30
Figure 13:High Level Specific Architecture 30
Figure 14:Low Level Architecture Diagram 31
Figure 15: Response from DQ Metric Web Service 36

LIST OF TABLES

Table 1: Comparison of Data Quality Dashboards 16
Table 2: Methodologies for Data Quality Assessment and Improvement [1,16] 19
Table 3: Overview Table for Data Quality Metrics 22

LIST OF SCREENSHOTS

Screenshot 1: User Interface to Compute Null Value Metric 41
Screenshot 2: User Interface for Preferential Data Quality 42
Screenshot 3:Overall Data Quality of NHTSA Dataset 44
Screenshot 4:Percentage of Null Values in Critical Fields of NHTSA Dataset 45
Screenshot 5: Percentage of Duplicate Values in Critical Fields of NHTSA Dataset 45
Screenshot 6: Non-Domain Values in Critical Fields of NHTSA Dataset 46
Screenshot 7: Percentage of Spelling Mistakes in Critical Fields of NHTSA 47
Screenshot 8: Summary of Data Quality of Critical Fields in NHTSA Dataset 47
Screenshot 9:Duplicate Values in DuDe Restaurant Dataset 48
Screenshot 10: Overall Data Quality of Twitter Dataset 49
Screenshot 11: Overall Data Quality of News Dataset 49

ABBREVIATIONS

DQ	Data Quality
NLTK	Natural Language Processing Tool Kit
TDQM	Total Data Quality Management
IQ	Information Quality
IP	Information Product
HDQM	Heterogeneous Data Quality Management
CDQ	Comprehensive Data Quality
ISO	International Standards Organization
DBMS	Database Management Systems
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
URI	Uniform Resource Identifier
REST	Representational State Transfer
JSON	JavaScript Object Notation
API	Application Program Interface
NOAD	New Oxford American Dictionary
PLC	Product Life Cycle
UI	User Interface
PLCA	Product Life Cycle Analytics
DOM	Document Object Model
AJAX	Asynchronous Javascript And Xml
HTML	HyperText Markup Language
SVG	Scalable Vector Graphics
D3	Data Driven Documents
CSS	Cascading Style Sheets
URL	Uniform Resource Locator

1 INTRODUCTION

Data plays a critical role in all business and government applications [1]. Low data quality leads to wrong or missing decisions, strategies and operations [2]. It slows down innovation processes and low data quality costs much money in organizations [3]. Sixty percent of enterprises suffer from data quality issues [4]. A survey reports that 70% of respondents have had their business affected by low data quality [5]. Low data quality could be an indicator of low process quality [6].

In organizations, workers, managers and customers produce structured and unstructured data [2]. More than 50% of data inside organisations are estimated to be unstructured and the remaining are structured [7]. Organisations need the information extracted from these data to make decisions and to be competitive [8]. Structured data such as machine data, product data and sales data and unstructured data such as emails, failure reports and customer complaints are produced during the product life cycle of a product and these data could be used to derive interesting hidden facts [9].

However, it is important to understand the quality of data before using the data for analytics to avoid data quality issues. To achieve this, we developed the PLCDQ dashboard which measures the quality of both structured and unstructured data accumulating in the product life cycle.

The PLCDQ dashboard provides an environment to measure the data quality of the PLC data using metrics. We implement and evaluate concrete indicators for measuring the quality of structured data such as percentage of missing values, duplicate values and non-domain values. We further integrate and validate the indicator percentage of spelling mistakes which is provided by the supervisor of the thesis, with the dashboard. Furthermore, we have designed our dashboard to be able to integrate other metric implementations.

1.1 Research Question

The research questions involved during the course of this thesis are listed in this section.

1. The main research question is to develop a dashboard for measuring Data Quality (DQ) of Product Life Cycle (PLC) data based on literature review and related work on dashboards and metrics.
2. Based on a comparison of existing data quality dashboards and literature review on data quality dimensions and metrics
    a. What are the most important and relevant dimensions for computing the DQ of structured and unstructured PLC data?
    b. What are the most important metrics for calculating the DQ of structured PLC data?
    c. What are good visualizations for presenting the DQ measurements?
3. How to provide the flexibility to integrate new metrics in the dashboard?
4. Identify useful features for DQ measurement of PLC data and implement a dashboard that includes these.

5. Show how DQ measurement works in this dashboard for concrete examples of PLC data.

1.2 Thesis Organization

The remaining parts of the thesis are organized as follows:

Chapter 2 provides information about data, data quality and the product life cycle. It describes various data quality dimensions and mentions different data quality methodologies and data visualizations.

Chapter 3 presents the existing data quality dashboards and the prototypical dashboards that are related to our work and provides a comparison of the PLCDQ dashboard with the existing data quality dashboards. The open research issues of data quality are also mentioned here.

Chapter 4 provides a brief description of various data quality metrics grouped under the corresponding data quality dimensions. The formula and steps to implement the metrics are illustrated.

Chapter 5 provides the concepts and the architecture of the PLCDQ dashboard.

Chapter 6 provides the implementation details of the web application. The libraries used for the implementation, modules in the web front end of the application and the web services are described. The user interface of the dashboard is also shown here.

Chapter 7 provides the demonstration and evaluation of the PLCDQ dashboard with the NHTSA, DuDe restaurant, Twitter and News datasets. The analysis of the quality of datasets with the data quality metrics of the dashboard are illustrated.

Chapter 8 provides the summary of the thesis and an overview for possible future works.

2 BACKGROUND

The literature review on topics related to the thesis is presented in this chapter. This chapter describes structured, semi-structured and unstructured data in section 2.1. The Section 2.2 provides a brief description on data quality. Various data quality dimensions are described in section 2.3. The section 2.4 contains a brief description of the product life cycle. Different data visualizations are mentioned in section 2.5. A brief description of RESTful web service is provided in the section 2.6.

2.1 Data

The scope of this section is to describe data and its classifications on structural representation which are structured, semi-structured and unstructured data.

The term data is derived from the Latin word dare which means to give. The New Oxford American Dictionary (NOAD) defines data as "facts and statistics collected together for reference and analysis" [10]. The International Standards Organization (ISO) defines data as a "re-interpretable representation of information in a formalized manner suitable for communication, interpretation or processing" [11]. Data is very important in several domains such as mathematics, statistics, medical science and information science. This makes it difficult to agree on a common definition for data suitable for all the domains, in which data is used. Authors like Blumenthal S.C., Fry J.P. and Sibley E.H define data as a set of facts [12, 13], while authors like Davis C.H. and Rush J.E. define data as the result of measurement or observation [14] and other definitions of data could be found in [5]. New Oxford American Dictionary defines data in relation to computing as the "quantities, characters or symbols, on which operations are performed by a computer, being stored and transmitted in the form of electrical signals and recorded on magnetic, optical or mechanical recording media" [10]. We will be restricting our discussion on data to the scope of information science. The definition of data which we found most suitable to our work is the one by the author Laura Sebastian-Coleman which is "Data are abstract representations of selected characteristics of real-world objects, events and concepts, expressed and understood through explicitly definable conventions related to their meaning, collection and storage" [15].

Different types of data such as perceptual data, linguistic data, elementary data, aggregated data and information as a product are described in. We restrict the scope of our work to three classifications of data based on the linguistic type, which are structured data, semi-structured data and unstructured data. [16]

- **Structured data** are those data whose schema is well defined and explicit. Schema denotes the formats, types, constraints and relationships of data [6]. Structured data represents real world objects that could be stored, retrieved and elaborated by database management systems (DBMS)[16]. Measurements, names, locations, numbers and dates stored in tabular databases are some examples of structured data.
- **Semi-structured data** are those data whose structure is defined, but it is not explicit. Data represented in XML or JSON formats are some examples of semi-structured data [6]. Semi-structured data is often used for exchanging information via internet [2]. Excel and e-mail are some examples of semi-structured data.

- **Unstructured data** are those data without a data model. Texts, speech, videos and pictures are some of the unstructured data sources. Text analytics, image recognition and speech recognition are used to extract knowledge from unstructured data [2]. Word documents, twitter data and wikis are some examples of unstructured data.

2.2 Data Quality

The New Oxford American Dictionary defines quality as "the standard of something as measured against other things of similar kind" and as "a distinctive attribute or characteristic possessed by someone or something" [10, 10]. The definition of data quality (DQ) which we found most suitable to our work is the one by the author Laura Sebastian-Coleman which is "the degree to which the data meets the expectations of data consumers, based on their intended uses of data". The practical problem associated is that most data consumers do not document the expectations of data or lack knowledge in defining the expectations of data [15].

Measurement of DQ could be classified into two broad scopes, which are objective measurements and subjective data assessments. Objective measurements measure the task independent characteristics of data. This measurement does not rely on the intended use of data, so that the measurement could be taken even when the intended use of data is not known. On the other hand, subjective data assessment strictly requires the intended use of data for DQ measurement. We will restrict the scope of this thesis to objective measures, since subjective data assessment is usually very specific to a scenario. Objective measures do not reject that DQ is defined by data consumers, instead it measures the characteristics which are basic for any data to be useful. Objective measures use data quality dimensions for measuring these characteristics. [15]

2.3 Data Quality Dimensions

Data quality dimensions are quality properties of data that could be measured and through which quality could be quantified [6, 15]. While fifteen dimensions have been identified in [17] for measuring the data quality, we restrict the scope of our discussion to the five popular data quality dimensions accuracy, completeness, validity, consistency, currency, interpretability and relevancy which are cited in many papers such as [2, 5, 17, 18, 18]. Different approaches to define data quality dimensions are mentioned along with the definitions in [16]. In this section, we provide a short description on these dimensions. The metrics used to measure the data quality, based on these dimensions will be discussed in chapter 4.

2.3.1 Accuracy

"Accuracy is defined as the closeness between a data value v and a data value v`, considered as the correct representation of the real world phenomenon that the data value v aims to represent" [16]. Accuracy could be measured for structured, semi-structured and unstructured data [6].Accuracy could be related with the dimensions precision, reliability and correctness [5, 16].

Let us consider an example for accuracy in which the data value v` that represents the real world is John and the data value v for which accuracy should be measured is Jhn. Here the data value v is inaccurate as it is not same as the data value v`. Accuracy could be classified into

temporal accuracy and structural accuracy based on the changes in the real-world phenomenon. The data value of temporal accuracy is updated with changes in the real world while the data value of structural accuracy remains stable as structural accuracy specifies the accuracy of data for a specific time frame. Structural accuracy of data is further classified into syntactic accuracy and semantic accuracy. Syntactic accuracy checks if a data value belongs to the corresponding domain of the data value. Here, we will not be comparing the data value v with the true data value v`, instead we will be comparing the data value v with the data values in the domain D. While, semantic accuracy checks if the data value v is the same as the data value v`. For example, let us consider the domain D as person, data value v` as John and data value v as Jack. Here v is syntactically accurate as it belongs to D and semantically inaccurate as it is not same as v`. [16]

2.3.2 Completeness

"Completeness is defined as the extent to which data are of sufficient breadth, depth and scope for the task at hand". Completeness could be calculated for structured relational data and web data [16]. Completeness could be related to the dimension duplication [5]. Breadth of data means that the dataset should contain all the desired attributes, depth of the data means that the dataset should contain the desired amount of data and finally the dataset should contain the attributes populated to a desired extent [15].

Let us consider an example for completeness with a person relation table with attributes name, birth date and email, which contains four records. The table is said to be complete if the values of all the attributes are present in the four records and is said to be incomplete, if any of the values of the attributes are missing. Completeness could be classified into three types, which are schema completeness, column completeness and population completeness. Schema completeness refers to the completeness of the concepts and properties of the schema. Column completeness refers to the completeness of a specific property or column. Population completeness evaluates completeness with reference to a reference population. [16]

2.3.3 Validity

Validity is defined as the "degree to which data conforms to a set of business rules, sometimes expressed as a standard or represented within a defined data domain". Let us consider a relation with attribute gender. The values of the attribute are said to be valid, if it is male or female, even though these values need not essentially be correct. Measurement of validity does not involve the comparison of data against real world objects. This characteristic of validity clearly differentiates it from the dimensions accuracy and correctness that require the comparison of data against real world objects. [15]

2.3.4 Consistency

Consistency is defined as the "degree to which data conforms to an equivalent set of data, usually a set under similar conditions or a set produced by the same process over time" [15]. Consistency is a necessary requirement for the data to be correct, but the converse is not true [5]. Consistency of a dataset could be checked against a set of standards or rules, a set of other data in a database, a set of other data in other systems and other data from a different instance of the same process. Consistency measures may reveal logical patterns of the real-world entity which the data represents [15]. Consistency could be related with the dimension integrity [5].

2.3.5 Currency

Currency is defined as the "temporal difference between the date in which data are used and the date in which the data are updated" [6].

Data is considered as current or up-to-date at time t, if it is correct at the time t. Data is considered as out-of-date, if it is incorrect at the time t, even though it was correct at a past instance of the time. For example, the address of a person in a relation person is not current, if it contains the past address of the person. Currency could be related with the dimensions timeliness and age. [5]

2.3.6 Interpretability

Interpretability is defined as the degree of similarity between the data in the dataset and the data expected by the current data consumer. For example, let us consider a statistical pre-processor that segments data into sentences as a data consumer that expects Chinese texts. The data is of low quality if English texts are passed to this data consumer because the expectation of the consumer and the input data differ. The dimension interpretability is important for unstructured data because many data consumers are used to interpret unstructured data automatically. [2]

2.3.7 Relevancy

Relevancy is defined as the degree of similarity between the data in the dataset and the optimal data for the task in hand. For example, let us consider an employee of a workshop searching for a solution to solve a problem with a machine in a knowledge base. The data is of low quality if he finds the price of the machine because the optimal data is the solution for the problem with the machine. Interpretable data that is not relevant to the data consumer is of low DQ. [2]

2.4 Product Life Cycle

Product Life Cycle Analytics (PLCA) is a platform and reference architecture for the holistic integration and analysis of unstructured and structured data from multiple data sources around the Product Life Cycle (PLC) [9]. Figure 1 provides an overview of the structured and unstructured data produced during the different phases of the PLC.

The six phases of the PLC are concept and product planning, design and development, production planning, production, use and support and reuse and recycling. The structured data are usually numbers stored in traditional databases as rows and columns. The unstructured data are contents in texts, pdfs, image, audio or video files. The planning and production phases of the PLC produce high volumes of structured data. The design and usage phases produce high volumes of unstructured data. [9]

2.5 Data Visualization

This section provides an overview of data visualizations and describes the pie chart and donut chart used for data visualizations in the dashboard.

A representation of data in pictorial or graphical format is termed as data visualization. It enables users to grasp difficult concepts quickly, identify new patterns and relationships,

pinpoint emerging trends and communicate the information to others. It is important to determine a best possible visual to represent the available data in an effective manner. [19]

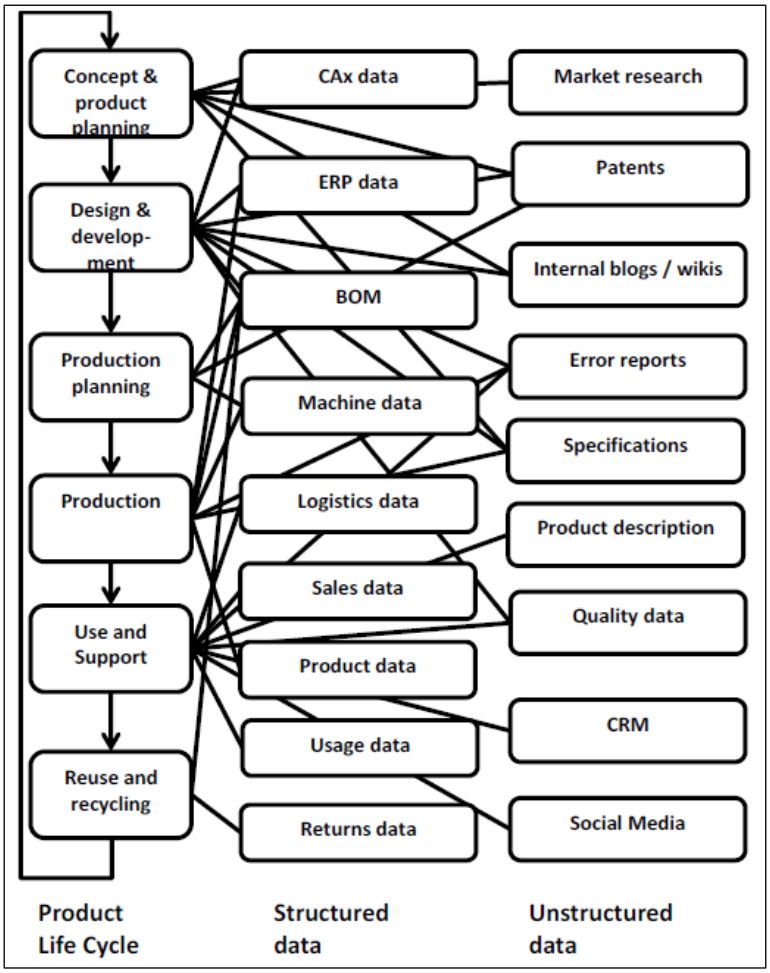


Figure 1: Data in the Product Life Cycle [9]

Different types of charts could be used for visualizing different types of data. Charts such as bar chart, pie chart, donut chart, line graph and bullet graph could be used to represent comparisons between values. Charts such as arc diagram, network diagram, venn diagram, chord diagram and connection map could be used to indicate relationships. Tree diagram, treemap, sunburst diagram and circle parking represents hierarchy. Connection map, dot map, flow map, chloropleth map and bubble map could be used to represent geographical regions. Charts such as histogram, density plot, dot matrix chart and tally chart represent distributions of data. [20]

Pie Chart

Pie charts are used commonly in presentations to display proportions and percentages between categories. It provides a quick overview on the proportional distribution of data. However, pie charts occupy more space and are not great for accurate comparisons of data. [20]

Donut Chart

Donut charts are like pie charts with an area cut out in the centre. Donut charts are used for comparison between categories. The user is focused on reading the length of the arc in donut chart as opposed to comparing the proportion of slices in pie chart. [20]

2.6 RESTful Web Service

The World Wide Web Consortium (W3C) defines web service as "a software application identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered as XML artifacts. A web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols" [21]. Representational State Transfer (REST) defines a set of architectural principles for defining web services. Each resource in REST has a unique Uniform Resource Identifier (URI) and all resources have a uniform interface for sending states between client and server. REST uses standard HTTP methods such as GET, PUT, POST and DELETE. In REST the interaction between client and server is stateless, that is each interaction should contain adequate information to be understood. REST uses layered system, that is a component could only view and interact with the immediate layer in the system. REST could be used to transfer XML and JavaScript Object Notation (JSON) messages [22].

3 RELATED WORK

This chapter provides a brief description of existing data quality dashboards in section 3.1 and a comparison of the existing data quality dashboards with the PLCDQ dashboard in section 3.2 and Prototypical data quality assessment tools are discussed in section 3.3. The last section 3.4 provides open research issues based on literature review on data quality dashboards.

3.1 Existing Data Quality Dashboards

This section provides an overview of the existing commercial data quality dashboards IBM Watson Analytics, Data Analyzer by Uniserv, Data Quality Scorecard by Uniserv, Data Quality Dashboard and Reporting by Informatica, Data Quality Analysis Package by Salesforce, Data Quality Dashboard by Talend, DQ Dashboard by Attacama, Data Quality Dashboard by InsightSquared and Data Governance Center 4.5 by Collibra. These dashboards are related to our work because they measure DQ we implement a data quality dashboard.

3.1.1 IBM Watson Analytics

IBM Watson Analytics is a data analysis solution in cloud which guides data discovery and predictive analysis with automatic visualizations and enables dashboard creation. With IBM Watson Analytics, the user could upload data, analyse the data quality of the uploaded data, ask a question and get insights from data, derive outcomes and create dashboards to illustrate the outcomes in a better way. [23, 24]

IBM Watson Analytics is a browser based dashboard that accepts structured data in the form of CSV files and Microsoft Excel files with .xls and .xlsx formats. Files with column headers are preferred. The number of columns in the header row is assumed to be the total number of columns present. For example, if the header row contains 5 column headers but there are 6 columns of data, the 6^{th} column is ignored. The professional edition of IBM Watson Analytics accepts data with maximum 10,000,000 rows and 500 columns. [25]

On analysis of data quality with IBM Watson Analytics, the user is provided with the following details [25]

- Overall average data quality score (based on average of the data quality score for every column in the dataset)
- Data quality score for each column (determined by metrics such as missing values and outliers)
- Percentage of missing data in each column
- Distribution graphs for numeric data values

Figure 2 displays the data quality score, percentage of missing values and distribution graphs for numeric values in each column. Watson analytics computes data quality scores of the raw input data. The data quality score is categorized as low, medium and high quality. A low data quality score indicates that the data may not be suitable for analytics. The overall data quality score is an average of the quality score for every column in the data set. [25]

The quality score is determined by the following indicators [25]

- Missing values
- Constant values
- Imbalance

- Influential Categories
- Outliers
- Skewness

The formulas used for calculating the above metrics is not mentioned in the documentation.

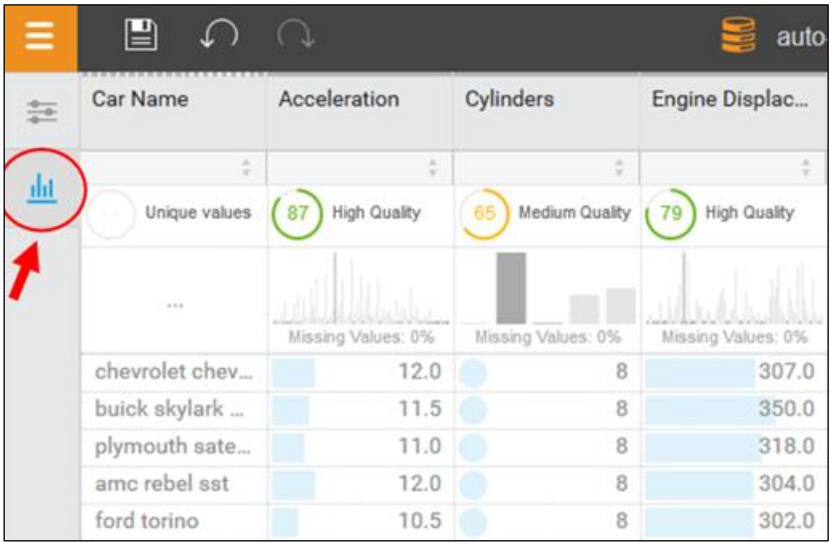


Figure 2:Watson Analytics - Data Quality Score for each column [25]

A bar chart is used for visualizing the data quality. After analysing the data quality of the uploaded data, Watson Analytics automatically excludes some fields with low data quality, if not explicitly stated otherwise by the user, before starting analytics. [25]

The reasons for excluding a field may be one of the following [25]

- Too many categories in a field
- Constant or near constant fields
- Missing values greater than 25%

IBM Watson Analytics does not support the user in cleansing and transforming the data. The user could improve the data quality manually if required before proceeding with data analytics.

3.1.2 Data Analyzer by Uniserv

Data Analyzer by Uniserv analyses and profiles data. It uses structured data for analysis. It uses column-oriented database technology, which supports the analysis of large datasets. The data analyser uses interactive analysis to get reliable status on the data quality of large amounts of data. The user could also define their own rules and metrics to get additional information on the data. With Data Analyzer from Uniserv, problematic data could be identified and further actions with their priorities could be derived. This data analyzer identifies missing information, outliers, format mismatch, inconsistent attributes and violation of rules from the data. The analyzer provides statistical information for each attribute of data. The provided statistical information includes percentage of null values, proportion of unique values and domain values in that attribute with its frequency of occurrence. Data Analyzer presents the data quality information in three different views called summary view, values view and data view. The summary view contains data quality information on each column. The values view contains

different values which occur in a column and its frequency of occurrence. The data view contains full records of data. Drill down functionality could be applied on these three views on any region of data. This helps to identify massive problems and dependencies of an attribute with other attributes. Drill down functionality could also be applied to user defined rules to derive additional information on data. [26]

3.1.3 Data Quality Scorecard by Uniserv

Data Quality Scorecard is another product from Uniserv. It is a browser based dashboard. It measures the quality of structured data accumulated in business processes of an organization such as campaign, service, customer and compliance management and provides data quality scores. The Data Quality Scorecard uses user-defined business rules with individual rule weighting to analyse the data quality. It provides data quality scores at the field level and at the record level. Data quality scores could also be produced by aggregating a group of fields, eg. by checking uniqueness on name and address fields. The Data Quality Scorecard does not provide any visalizations. [27]

3.1.4 Data Quality Dashboard and Reporting by Informatica

Data Quality Dashboard and Reporting of Informatica is a management tool that captures a virtual snapshot on the quality of data. It is a browser based dashboard that profiles, cleanses and certifies data across the entire organization. The dashboard supports structured data regardless of size from different platforms like cloud, Hadoop and embedded data. [28]

Data Quality Dashboard of Informatica profiles the data using a set of algorithms. These algorithms are used for statistical analysis, for analysing data quality within a data set and for discovering relationships between values within and across data sets. The data profiling tool in this dashboard provides frequency distribution of different values for each attribute, which provide insights on the type and use of the attribute. The result from data profiling is used for defining data quality rules. Data profiling tools, data transformation and data cleansing tools could be directly integrated with this software. The data quality rules of Data Quality Dashboard and Reporting of Informatica are of two types. They are validation rules and cleansing or correction rules. Validation rules verify if the data satisfies a certain condition, eg. all values in a column should be greater than 20. Cleansing or correction rules checks for a condition and modify the data automatically if the condition is not met, eg. telephone number should be in a format separated into its area code, exchange and line components. [29]

Data Quality Dashboard and Reporting of Informatica develops data quality rules based on data quality dimensions. The data quality dimensions such as accuracy, consistency and completeness are considered. The data quality dimension accuracy is ensured by verifying the values in a dataset with a source of correct information. The data quality dimension consistency is verified in different contexts which are record level consistency, cross-record consistency, temporal consistency and reasonableness. The data quality dimension completeness is verified using rules. [29]

The Data Quality Dashboard and Reporting of Informatica presents the result of data quality in three forms. They are thresholds for conformance, historical levels of conformance and as a data quality scorecard. [29]

Item	Passed %	Target %	40% 100%
Weighted Average	**83.9**	**93.0**	
Company Name Conformity	41.3	90.0	
TL - Customer Number Conformity	99.9	98.0	
TL - Company Name Conformity	74.6	85.0	
TL - Contact Name Conformity	99.9	99.0	
exposure_amount_conformity	95.1	96.0	
PD_conformity	97.0	96.0	
EAD_conformity	79.2	95.0	

Figure 3:Informatica-Threshold of conformance [29]

Figure 3 represents the view threshold for conformance in the Data Quality Dashboard and Reporting of Informatica. This view is presented as a chart diagram. This view classifies quality of data into three threshold levels, which are acceptable, questionable and unusable, based on how the data conforms to the defined data quality rules. [29]

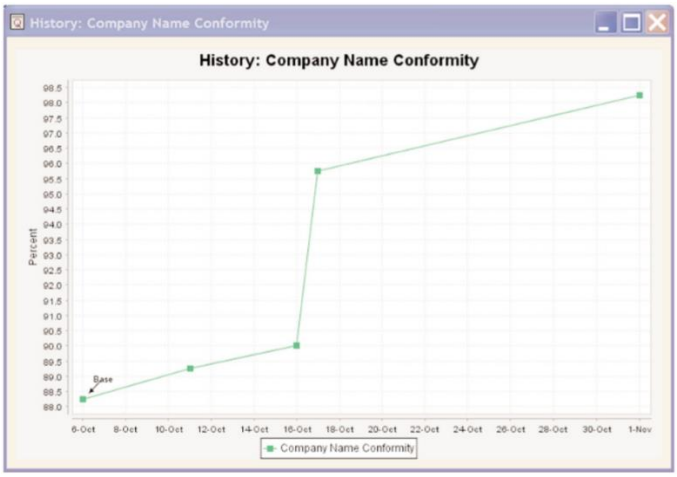


Figure 4:Informatica-Historical level of conformance [29]

Figure 4 represents the view historical level of conformance in the Data Quality Dashboard and Reporting of Informatica. This view is presented as a line graph. This view presents historical view on data quality of when and how much the quality of data is improved. [29]

The data quality scorecard in the Data Quality Dashboard of Informatica visualizes the collected scoring on the conformance of data to the defined data quality rules based on the data quality dimensions which provide an overview on the data quality. The status of data quality could also be viewed in drilled down form to specify areas of improvement and to identify trends. The visuals could be in the form of bar chart, pie chart and graphs. [29]

3.1.5 Data Quality Analysis Dashboard Package by Salesforce

The Data Quality Analysis Dashboard Package of Salesforce is a free of charge application by Salesforce Labs on AppExchange [30]. It is a browser based dashboard that provides data quality scores for structured data [31].

The Data Quality Analysis Dashboard Package of Salesforce uses custom formula fields to record data quality and completeness. The data quality score is based on number of fields populated in a record. The user could customize the application by choosing the fields which

are relevant for calculating data quality score, in case it is not important to populate all fields in a record. [30]

The Data Quality Analysis Dashboard Package of Salesforce uses bar and pie charts for visualization [31].

3.1.6 Data Quality Dashboards by Talend

Talend is an open source data management company that offers end-to-end data management tools. The Data Quality Dashboard of Talend monitors and reports the data quality of structured data available across the entire organization. The Dashboard profiles, cleanses and standardizes data. Data profiling provides evaluation on current state of data quality as well as measurement of data quality over time. Data cleansing and standardizing is achieved by using inbuilt tools which repair and cleanse data. The data quality standards for data cleansing could be set using data which could be referenced. The data quality standards include sets of standards for values, regular expressions, data shape and size. The Data Quality Dashboard of Talend provides data quality in a customizable web-based portal and in the form of data quality reports. The results are visualized as bar charts and pie charts. The data quality portal could be personalised according to the role of the user within an organization. This includes personalised alerts, data views and links. The dashboard provides access to users based on their roles, that is users like managers and team members could view only the dashboards which are required for their job. [32]

3.1.7 DQ Dashboard by Attacama

The DQ Dashboard by Attacama monitors data quality and provides statistical visualizations of the quality of data. This dashboard monitors structured data independent of the data domain. The DQ Dashboard provides multiple levels of customizable dashboards such as a master dashboard that provides a quick management overview of DQ, an admin-defined dashboard that provides all key metrics and a user-defined dashboard that provides metrics and charts specific to each user. The user could also configure actions which could be triggered based on certain conditions, eg. when the data quality falls below a certain threshold. The actions may include email notifications for certain users and dashboard visualization. The DQ Dashboard is a web-based dashboard which could be shared, exported, printed and could be inserted into a presentation. This Dashboard represents data quality in the form of pie charts, bar charts and time series charts. This dashboard supports root cause analysis of a problem. The methods for root cause analysis of the problem are not specified in the documentation. Particular combination of data dimensions causing the problem could be identified, also original data causing the problem could be drilled through to learn the context of the problem. [33]

3.1.8 Data Quality Dashboard by InsightSquared

The Data Quality Dashboard by InsightSquared is a browser based dashboard that monitors and cleanses data. The data quality is monitored and the errors are presented along with its priority. The data cleansing is done by filling the data gaps with values obtained from intelligent historical metrics. The format and domain of the data is not specified in the

documentation. The Data Quality Dashboard by InsightSquared presents data quality results in the form of bar charts and tables. [34]

3.1.9 Data Governance Center 4.5 by Collibra

Collibra provides business-focused applications for automating data management processes [35]. Their product Data Governance Center 4.5 is a web based platform that includes a data quality dashboard. This data quality dashboard accepts structured data in CSV, JSON, XML and Excel file formulae formats [36].

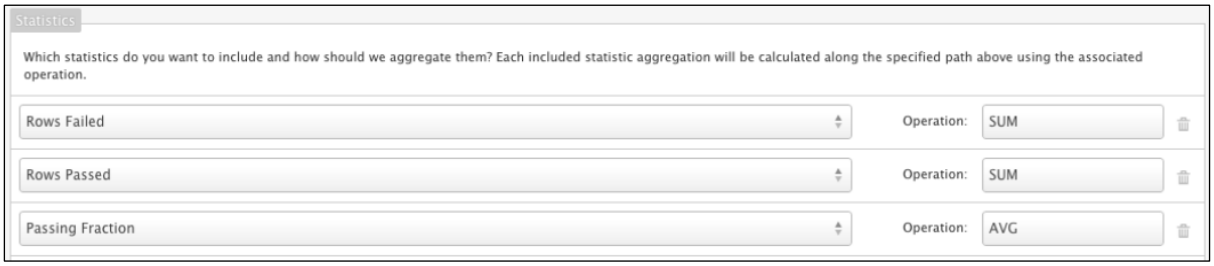


Figure 5:Collibra metric configuration [37]

The Data quality dashboard calculates data quality on assets based on defined metrics. The administrator could configure the metrics. The user interface to configure the metrics in Data Governance Center 4.5 of Collibra is shown in Figure 5. The Statistics tab defines the values which should be shown in the dashboard and the operations for aggregating those values. [37]

Validation rules are used in Data Governance Center 4.5 of Collibra to validate assets. Each validation rule should have a validation script. The validation script has a groovy[1]-based syntax customized by Collibra for ease of use. The validation script contains three constructs. They are given, when and then. The construct given is used for defining variables which could be used in the rest of the script. The construct when is used to define constraints for using validation. The construct then is the actual validation logic. [38]

```
rule {
    given {
        definitions = attributes['Definition']
    }
    when {
        isEqual(type.id,'00000000-0000-0000-0000-000000011001')
    }
    then {
        isNotEmpty(definitions, message: "The asset ${name} in domain ${vocabulary.name}
        must have at least one definition")
    }
}
```

Figure 6:Data Governance Center 4.5 - Validation Script [38]

The validation rule in Figure 6 takes values from Definition attributes and stores them in the variable definitions as a list. Then it checks for the assets with asset type id equals to '00000000-

[1] http://www.groovy-lang.org/

0000-0000-0000-000000011001' to apply the rule. The rule validates to true if at least one definition attribute is present, else an error message is displayed. [38]

The Data Governance Center 4.5 of Collibra presents data quality using pie charts as shown in Figure 7. Each pie chart shows three aspects of the specified data quality score. The three aspects are passing data quality score in percentage, colour coded indication of data quality score and arrow indication of trend of the score compared to the previous measurement. [39]

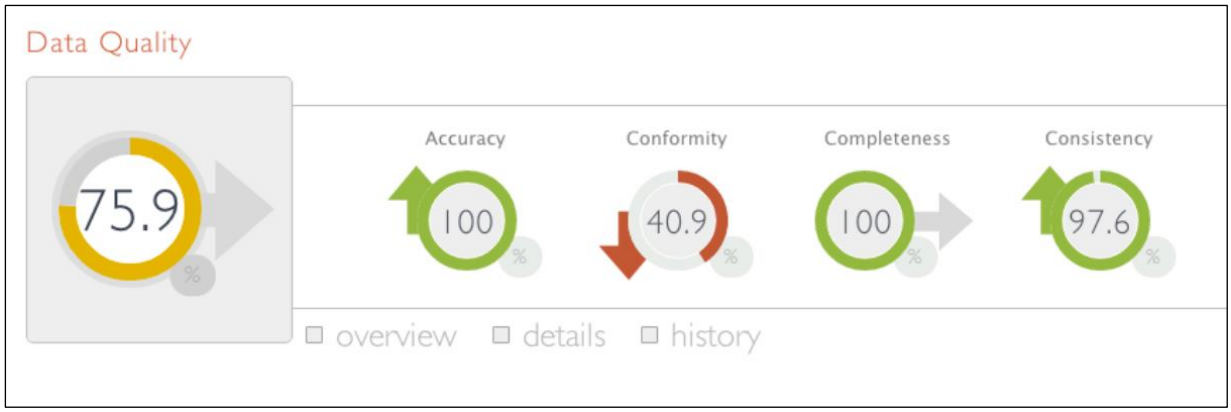


Figure 7: Data Governance Center 4.5 - Data Quality Dashboard [39]

The first pie chart in Figure 7 represents the overall quality score of the asset. The other pie charts represent the sub-scores for certain dimensions like accuracy, conformity, completeness and consistency. These dimensions should be configured in the metrics to provide automatic pie chart representation of the dimensions. [39]

The dashboard contains three other views, which are overview, details and history. History pane shows the improvements of data quality over a time period of one month. The overview pane provides more information on each level in the aggregation path for selected general score or dimension. [39]

Data Asset	Passing Rows	Failing Rows	Quality Score	Result
Policy	21752.0	26823.0	75.86 %	✗
Insurance Policy	21752.0	26823.0	75.86 %	✗
Channel Validation	3829.0	23253.0	15.00 %	✓
Maximum Cover Validation	257.0	243.0	97.57 %	✓
Policy Number Prefix Validation	0.0	0.0	100.00 %	✓
Cover Validation	7666.0	3322.0	66.78 %	✗
Broker Validation	10000.0	5.0	99.95 %	✗

Figure 8: Data Goverenance Center 4.5 - Details Pane [39]

Figure 8 represents the details view of the data quality dashboard of Data Governance Center 4.5 by Collibra. The details pane shows more information about all assets in a hierarchical table

format. The table contains the name, number of passing rows, number of failing rows, aggregated quality score and result of each asset. [39]

3.2 Comparison of Data Quality Dashboards

This section provides a comparison of the existing data quality dashboards mentioned in section 3.1 with the developed data quality dashboard. A tabular form of this comparison is presented in Table 1.The Data Analyzer and Data Quality Dashboard by Uniserv are merged in the table as they both are provided by Uniserv. In Table 1, U represents Uniserv, I represents Informatica, S represents Salesforce, T represents Talend, A represents Attacama, C represents Collibra, P represents PLCDQD, + represents supports, - represents does not support and ? represents not known.

	IBM	U	I	S	T	A	C	P
General								
PLC Data	-	-	-	-	-	-	-	+
Open Source	-	-	-	-	-	-	-	+
Support Large Datasets	+	+	+	?	?	?	?	-
Data Cleansing	-	+	+	?	+	?	?	-
Personalisation	-	-	-	-	+	+	-	-
User Defined Rules	-	+	?	-	+	+	+	-
Overall Data Quality Score	+	-	-	-	-	+	+	+
Browser Based Dashboard	+	+	+	+	+	+	+	+
Data Types								
Structured Data	+	+	+	+	+	+	+	+
Unstructured Data	-	-	-	-	-	-	-	+
Dimensions								
Completeness	+	+	+	+	?	-	+	+
Uniqueness	+	+	+	-	-	-	-	+
Relevancy	-	-	-	-	-	-	-	+
Interpretability	-	-	-	-	-	-	-	+
Accuracy	-	-	+	-	-	-	+	-
Consistency	-	+	+	-	-	-	+	+
Visualizations								
Bar Chart	+	-	+	+	+	+	-	-
Pie Chart	+	-	-	+	+	+	-	+
Donut Chart	+	-	-	-	-	-	+	+
Time Series Chart	-	-	+	-	-	+	+	-
Tables	+	+	+	+	-	-	+	+

Table 1: Comparison of Data Quality Dashboards

All the existing data quality dashboards support measurement of quality of structured data but fail to support unstructured data. The significance of our dashboard is that it supports data quality measurement of both structured and unstructured text data from CSV and JSON files. The next important significance of our dashboard is that it is designed for and evaluated with PLC data, while most of other dashboards are designed to support business or organizational data. Our dashboard is the only open source data quality dashboard available to the extent of our knowledge. Our dashboard computes the overall data quality of the data. Only few

dashboards such as IBM Watson analytics, DQ Dashboard by Attacama and Data Quality Dashboard by Talend computes the overall data quality of data while others do not compute it. The data quality dimensions such as completeness and uniqueness are measured in most of the dashboards while dimensions such as relevancy and interpretability are measured only in our dashboard. Our dashboard uses pie chart, donut charts and tables for visualizing the quality of data.

3.3 Prototypes of Data Quality Dashboards

This section provides a brief description of the Software tool for TDQM Methodology and Data Quality Dashboard for Reliability Data.

Software Tool for TDQM Methodology

The software tool illustrated in the paper [17] is used in the defining phase of the TDQM methodology described in chapter 4.1. This tool is used for defining the IQ requirements from the perspective of information providers like information suppliers, manufacturers, consumers and IP managers [17]. The data is collected from all the information providers through surveys and are stored in a survey database. The tool performs a query that maps the metrics obtained from the surveys to the corresponding IQ dimensions. The tool provides a graphical representation of the assessment of IQ from the point of view of information providers using the survey data. This tool does not calculate the DQ automatically, instead it uses the result of surveys to assess IQ of the data. Figure 9 shows a sample graphical output of the tool using the survey data collected from manufacturer and consumer. Here the IQ is assessed based on the listed dimensions from the point of view of manufacturer and consumer. From the graph we could interpret that the manufacturer and the consumer have same opinion on the dimension objectivity, whereas there is a difference in opinion between them on the dimension completeness. In PLCDQ dashboard, the dimensions relevancy, completeness and interpretability are measured, however the dimensions are not accessed from the point of view of information providers. [17]

A Data Quality Dashboard for Reliability Data

The paper "A Data Quality Dashboard for Reliability Data" proposes a data quality dashboard to identify the data quality problems in PLC data. Product manufacturers and equipment maintenance organizations use equipment failure data to understand the failure behaviour of their machinery. This paper groups the data quality problems encountered by the reliability data into six categories by examining the reliability-related data from five cases from different contexts. The six categories of data quality problems are sample size, sample selection, data cleanness, free of error, data completeness, level of detail and trustworthiness. A prototypical data quality dashboard is developed with seven data quality metrics based on these categories. This prototype provides the resulting metric values, an assessment on the quality of data and a list of actions to improve the data quality. Traffic-light representation is used in the prototype to visualize the data quality. This paper describes the formula and rules used in the prototype to calculate the seven metrics in detail. In PLCDQ dashboard dimensions such as completeness and validity are measured for PLC data using data quality metrics and are visualized in the form of donut and pie charts. [18]

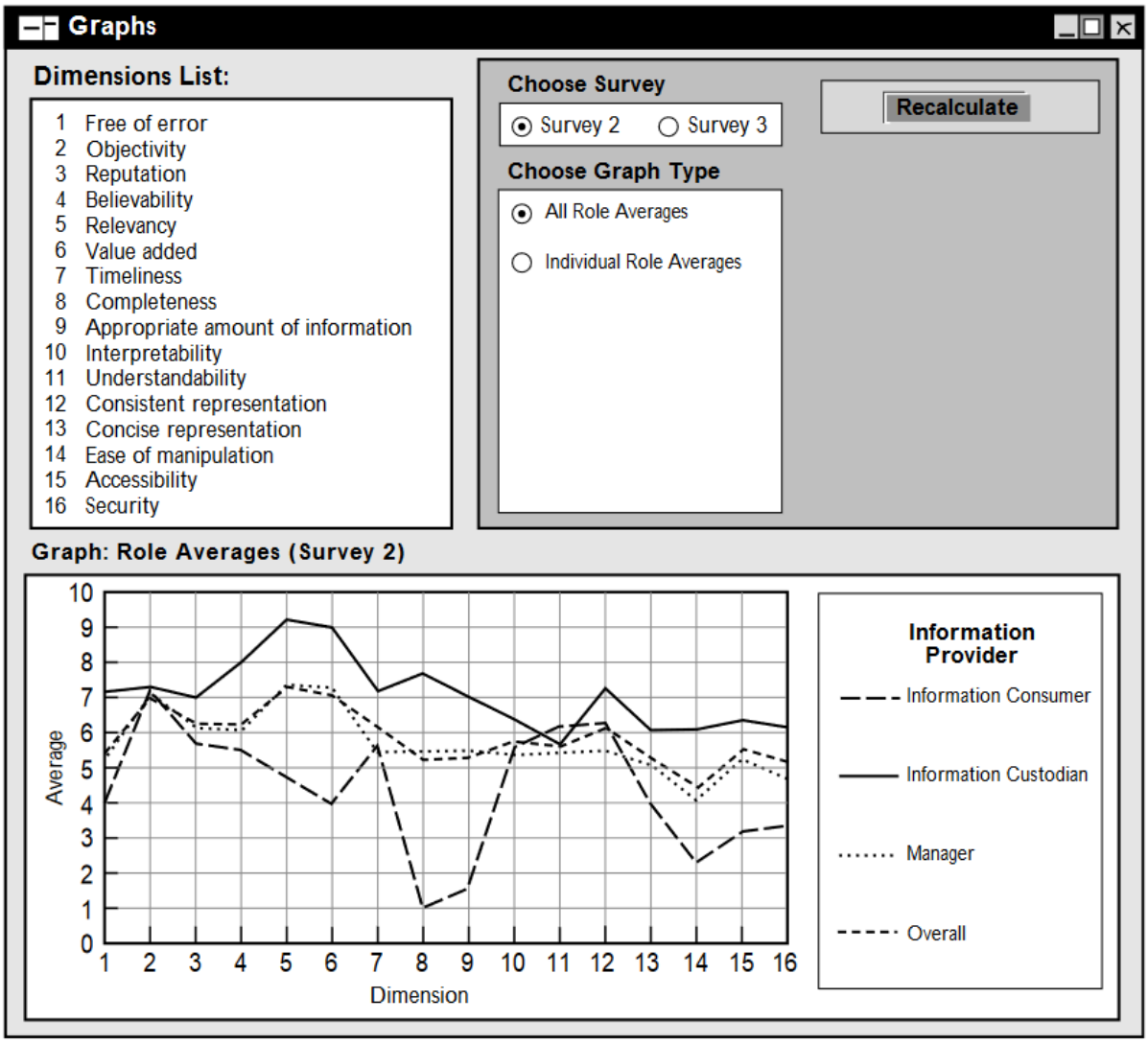


Figure 9:Dimensional Assessment of IQ across Information Providers [17]

Additionally, there are tools for data cleansing and data transformation such as OpenRefine, Drake and DataWrangler.

3.4 Open Research Issues

The open research issues based on related work and data quality metrics chapters are listed as follows

1. All dashboards support structured data but do not compute the data quality of unstructured data which is big part of PLC data.
2. Open source dashboards to compute the data quality of structured and unstructured data are not available.
3. Most of the existing dashboards calculate the data quality dimensions such as completeness and uniqueness but do not compute dimensions such as relevancy and interpretability which are usually related to unstructured data.
4. Dashboards to compute the data quality of PLC data are not available.

4 DATA QUALITY METRICS

This chapter is a result of literature review. It provides an overview of data quality metrics with focus on PLC data. The goal of this chapter is to select three DQ metrics to implement in the dashboard. Different data quality methodologies are mentioned and the TDQM methodology is described in section 4.1. The section 4.2 provides a brief description of the indicators to measure the quality of unstructured text data.

4.1 Overview on Data Quality Methodologies

Data quality methodologies provide guidelines to choose the most effective data quality measurement and improvement process [16]. Measuring data quality is usually a basic phase in the DQ methodologies. There are different data quality methodologies as shown in *Table 2* which are discussed in detail in [1,6]. In the table the data S stands for structured data, SS stands for semi-structured data, U stands for unstructured data and ISS stands for implicitly considered semi-structured data which means the methodology does not mention the type of data but phases and steps can be applied to it [1]. The data type supported by a methodology is a crucial discriminating factor for the case of PLC data.

Methodology Acronym	Extended Name	Data	Reference
TDQM	Total Data Quality Management	S, SS, U	[17]
DWQ	The Datawarehouse Quality Methodology	S	[40]
TIQM	Total Information Quality Management	S, ISS, U	[41]
AIMQ	A methodology for information quality assessment	S, ISS	[42]
CIHI	Canadian Institute for Health Information Methodology	S, SS	[43]
DQA	Data Quality Assessment	S	[44]
IQM	Information Quality Measurement	S, SS	[45]
ISTAT	ISTAT Methodology	S, SS, U	[46]
AMEQ	Activity-based Measuring and Evaluating of product information Quality methodology	S, ISS	[47]
COLDQ	Cost-effect Of Low Data Quality	S, ISS	[48]
DaQuinCIS	Data Quality in Cooperative Information Systems	S, SS	[49]
QAFD	Methodology for the Quality Assessment of Financial Data	S	[50]
CDQ	Comprehensive methodology for Data Quality management	S, SS	[51]
HDQM	Heterogeneous Data Quality Methodology	S, SS	[6]

Table 2: Methodologies for Data Quality Assessment and Improvement [1,16]

All these methodologies use some common phases for assessing and improving the DQ. The common phases used for DQ assessment are analysis, IQ requirements analysis, identification of critical areas, process modeling and measurement of quality. The analysis phase provides knowledge on data and its architecture, IQ requirements analysis phase identifies quality issues and sets new quality targets through surveys, identification of critical areas identifies the most important data, process modeling creates a model of the processes that produces and updates information and the measurement phase selects the DQ dimensions and defines the corresponding DQ metrics. [16]

We exemplary discuss in detail the TDQM methodology as it is a general-purpose methodology, supports a wide range of DQ dimensions and supports structured, semi-structured and unstructured data[1, 16].

A Product Perspective on Total Data Quality Management

The TDQM methodology assumes information as a product produced from an information manufacturing system, similar to a physical product like a car produced from a manufacturing system. This methodology identifies four roles of information providers, which are information suppliers, information manufacturers, information consumers and IP managers. The TDQM methodology uses the TDQM cycle, in which the phases are performed in an iterative manner as shown in Figure 10. The phases in the TDQM cycle are defining, measuring, analysing and improving IP. The defining phase identifies the characteristics, the IQ and the information manufacturing system of the Information Product(IP). This phase provides a quality entity-relationship model that defines the IP and its IQ requirements and an information manufacturing system that describes the interactions among the four roles of providers mentioned above. The measuring phase develops IQ metrics for measuring the IQ along different dimensions. The IQ metrics could be basic measures like data accuracy, timeliness, completeness and consistency or complex business rules like total risk exposure of a client should not exceed a certain limit or information-manufacturing-oriented IQ metrics based on measures like security and credibility. The analysing phase investigates the root cause of the current IQ problems. Analysis could be performed by introducing dummy accounts into the information manufacturing system, Statistical Process Control, pattern recognition and Pareto chart analysis. The improving IP identifies key areas of improvement such as aligning information flow and work flow with the corresponding information manufacturing system and realigning the key characteristics of IP with the business needs. [17]

The DQ dimensions supported by TDQM are completeness, believability, consistent representation, timeliness, interpretability, relevance, accessibility, appropriateness, ease of manipulation, value added, free of error, objectivity, reputation, timeliness and understandability. The DQ metrics such as percentage of syntactically accurate values, percentage of null values, percentage of consistent values, time of last update and time length for which data remain valid could be measured using TDQM. [1]

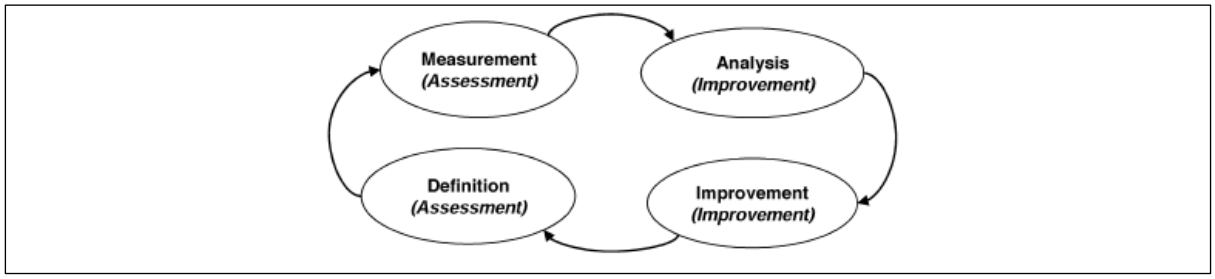


Figure 10:Phases of TDQM Methodology [1]

4.2 Assessing the Quality of Unstructured Data

In [2] the author describes the importance to ensure high quality unstructured data, provides three data quality dimensions that are relevant to the mining processes on unstructured data and provides indicators to automatically measure the quality of unstructured text data used in analytic pipelines. Figure 11 represents the dimensions and the corresponding indicators to measure the quality of unstructured text data in analytic pipelines. The paper also provides

hints on how to implement the indicators. The future work mentioned in this paper is to choose the most suitable implementation for the indicators and to validate them with experiments. [2]

Dimension	Indicator
Interpretability	Fit of training data
	Confidence
	Noisy data
Relevancy	Frequent keywords
	Specificity
Accuracy	Precision
	Accuracy
	Quality of gold annotations

Figure 11: Indicators for measuring the quality of unstructured text data [2]

Measurement of dimensions such as length, width and height of an object allows us to understand the characteristics of an object. Similarly, measurement of data quality dimensions of data such as completeness, validity and relevancy of data allows us to understand the data quality of the data. The data quality metrics are used to measure the data quality dimensions of data. Data quality metrics describe where the measurements are taken, what data are included, the measurement device and the scale on which the results are reported. [16]

The scope of our thesis is to calculate the data quality of structured and unstructured text data accumulating in the Product Life Cycle (PLC) of a product. All products are designed and manufactured using PLC data. Data quality issues in PLC data will lead to defects in product and loss of money [52]. The data quality dimensions completeness, validity, integrity, interpretability, relevancy and consistency described in chapter 2.3 could be applied to both structured and unstructured data and these dimensions also provide data quality issues in PLC data [2, 52]. So, we restrict the scope of our thesis to focus on the dimensions completeness, validity, integrity, interpretability, relevancy and consistency. Most of the PLC data are present in files [52]. So as an initial step of calculating the data quality of PLC data we focus on data from files in CSV and JSON file formats. The data in these files contain data in the format of rows and columns. Again, as an initial step we restrict the scope of our data quality metrics to compute measurements from a single dataset from a single file.

Table 3 provides an overview of the metrics described in this chapter. This table serves as the basis for our decision on the metrics to implement in the PLCDQ dashboard. It also maps each metric with its dimension, data type supported by the metric and the section in which the metric is explained in detail. In the column *Data Types Supported* in Table 3, S represents structured data and U represents unstructured text data.

Dimensions	Metrics	Data types supported	Used in PLCDQD	Detailed description given in chapter
Completeness	Percentage of null values	S	yes	4.3.1
	Percentage of default values	S	no	4.3.2
	Percentage of duplicate values	S	yes	4.3.3
Validity	Percentage of non-domain values	S	yes	4.4.1
	Percentage of non-range values	S	no	4.4.2
	Percentage of outliers	S	no	4.4.3
Consistency	Percentage of inconsistent format in a field	S	no	4.5.1
Interpretability	Percentage of spelling mistakes	U	yes	4.6.1
	Fit of training data	U	no	4.6.2
Relevancy	Percentage of non-relevant data	U	no	4.7.1

Table 3: Overview Table for Data Quality Metrics

The data quality metrics mentioned in table 3 are grouped under the corresponding data quality dimensions such as completeness, validity, consistency, interpretability and relevancy described in chapter 2.3. The description of each data quality metric contains an overview of the metric and the formula steps to compute the metric. The metrics for structured data could be applied to PLC data such as bill of materials, sales data, product data and machine data. The metrics for unstructured data could be used to measure the quality of PLC data such as textual description of failures of machines as free text fields in csv files. The result obtained from all metrics are in the interval (0,100) where 0 indicates low data quality and 100 indicates high data quality.

4.3 Completeness

This section provides a brief description of the metrics percentage of null values in section 4.3.1, percentage of default values in section 4.3.2 and percentage of duplicate values in section 4.3.3 which are used to measure the data quality dimension completeness [15].

4.3.1 Percentage of null values

The metric percentage of null values ensures the identification of values in a dataset which are not populated. Missing data degrades the quality of data in a dataset and increases the cost of obtaining the data. The prerequisite for computing the metric percentage of null values is the metadata that contains the number of columns and the number of rows in the dataset. The metric percentage of null values supports structured data. [15]

The formula to compute the percentage of null values deduced from [1, 15] is shown below

$$\text{Percentage of null values} = \frac{Number\ of\ null\ values\ in\ the\ dataset}{Total\ number\ of\ values\ in\ the\ dataset} * 100$$

The metric percentage of null values could be further classified into percentage of null values in non-nullable fields and percentage of null values in a record which are described below.

Percentage of null values in non-nullable fields

The metric percentage of null values in non-nullable fields ensures the identification of null values in mandatory columns in a dataset. In a dataset, some columns are mandatory while others are optional. The presence of null values in mandatory columns signifies an incomplete dataset. Incomplete datasets degrade the quality of data, which leads to faulty analysis or incorrect conclusions from the data in the dataset. Thus, rework on analysis of data is required, which in turn increases the cost of data analysis. The prerequisites for computing the metric is the metadata that contains two specific details on the dataset. The first detail is the column headers of the non-nullable fields for which this metric should be calculated. The second detail is the different representations of null values in a dataset. Some representations of null values are NULL, NA and values left completely blank. The metric percentage of null values in non-nullable fields operates on data in the field level. [15]

Percentage of null values in a record

The metric percentage of null values in a record ensures the identification of records that do not satisfy the defined expectation of the number of non-nullable values in a record in the dataset. The prerequisite for computing the metric percentage of null values in a record is the metadata that contains two specific details. The first detail is the expected record length which means the expected number of fields to be populated in a record of the dataset. The other detail is the logic to measure the record length in the dataset. The percentage of null values in a record operates on data at the record level. [15]

The formula to compute the percentage of null values in a record deduced from [1, 15] is shown below

$$\text{Percentage of null values in a record} = \frac{Number\ of\ null\ values\ in\ a\ record}{Expected\ number\ of\ values\ in\ a\ record} * 100$$

4.3.2 Percentage of default values

A Default value is a specific value assigned to a field to signify the non-availability of data or incorrect data such as out of range values for that field. A field may contain a consistent proportion of default values, however significant changes in the proportion levels may indicate missing values in the dataset. The metric percentage of default values ensures the identification of missing values in a dataset. This metric could be applied to critical fields in which few default values are expected and to fields in which missing or default values could impact the process of data analysis. The prerequisite for the metric percentage of default values is metadata that contains two specific details. The first detail is the column header for which percentage of default values should be calculated and the other detail is the default value specified for that respective column. [15]

The formula to compute the percentage of default values in a field deduced from [15] is shown below

$$\text{Percentage of default values} = \frac{Number\ of\ default\ values\ in\ the\ field}{Number\ of\ rows\ in\ the\ dataset} * 100$$

4.3.3 Percentage of duplicate values

The metric percentage of duplicate values ensures identification of duplicate values. Duplicate records could cause confusion for the users. Data could be misunderstood and the number of records represented in the dataset could be incorrect [15]. For example, consider a table that

lists the contact information of people living in a city which contains two records with two different addresses for a person Bob in which one is duplicate. From this data, one may misinterpret that there are two persons with the same name as Bob in these address which leads to incorrect count of people living in that city.

The logic behind the identification of duplicate record and the logic behind the identification of unique record should be available in the dataset. The logic behind the identification of duplicate records could be of different criteria. One criterion is that a record is termed as duplicate if all the field value of a record matches exactly with the all the field values of another record. Another criterion is that a record is termed as duplicate if some specified field values of a record match exactly with the same specified field values of another record. The prerequisite to calculate the metric percentage of duplicate values is metadata that defines the logic behind the identification of unique record in the dataset. [15]

The formula to compute the percentage of duplicate values in a field or a combination of fields deduced from [15] is shown below

$$\text{Percentage of duplicate values} = \frac{Number\ of\ duplicate\ values\ in\ the\ field}{Number\ of\ rows\ in\ the\ dataset} * 100$$

4.4 Validity

This section provides a brief description of the metrics percentage of non-domain values in section 4.4.1, percentage of non-range values in section 4.4.2 and percentage of outliers in section 4.4.3 which are used to measure the data quality dimension validity [15].

4.4.1 Percentage of non-domain values

The metric percentage of non-domain values ensures the identification of non-domain values in a field that contains values from the defined domain values. Domain values are valid values defined in reference files or reference tables. Domain values provide basic expectations for data in a specified field. The metric percentage of non-domain values supports structured data. This metric is associated with the data quality dimension validity. The prerequisite for the metric percentage of non-domain values is metadata that provides two specific details. The first detail is the column header for which percentage of non-domain values is to be computed. The second detail is the set of domain values for that specified column. [15]

The formula to compute the percentage of non-domain values in a field deduced from [15] is shown below

$$\text{Percentage of non domain values} = \frac{Number\ of\ non\ domain\ values\ in\ the\ field}{Number\ of\ rows\ in\ the\ dataset} * 100$$

4.4.2 Percentage of non-range values

The metric percentage of non-range values ensures the identification of non-range values in a field that contains values from a defined range of values. Range of values specifies minimum and maximum values acceptable in that field. A range of values provide basic expectations for data in a specified field to contain values within the minimum and maximum ranges of values. Non-range values are the values which do not fall between the minimum and maximum ranges of values. The prerequisite for the metric percentage of non-range values is metadata that

provides two specific details. The first detail is the column header for which percentage of non-range values is to be computed. The second detail is the minimum and maximum ranges of values for that specified column. [15]

The formula to compute the percentage of non-range values in a field deduced from [15] is shown below

$$\text{Percentage of non range values} = \frac{Number\ of\ non\ range\ values\ in\ the\ field}{Number\ of\ rows\ in\ the\ dataset} * 100$$

4.4.3 Percentage of outliers

The metric percentage of outliers ensures the identification of outliers in a field. Outliers are data values that differ from the rest of the values in the field. For example, consider a dataset containing the height of boys in a kindergarten class with most values ranging from 39 to 41 inches, the data of a boy with a height of 50 inches is considered as an outlier because it differs from the rest of the values in the dataset. [53]

An outlier may indicate incorrect data or correct but exceptional data. The essential metadata to compute the percentage of outliers is the column header of the field for which the metric should be computed. [54]

The formula to compute the percentage of outliers in a field deduced from [1, 54] is shown below

$$\text{Percentage of outliers} = \frac{Number\ of\ outliers\ in\ the\ field}{Number\ of\ rows\ in\ the\ dataset} * 100$$

The number of outliers in the field is determined by using libraries such as SciPy[2] or any other outlier detection modules. [1, 55]

4.5 Consistency

This section provides a brief description of the metric percentage of inconsistent format in a field which measures the data quality dimension consistency [15].

4.5.1 Percentage of inconsistent format in a field

The metric percentage of inconsistent format in a field ensures the identification of inconsistent formatting of data in a field. Inconsistent data are difficult to use. Establishing standards for formatting and defaulting of data removes minor inconsistencies in data. Consistent representation of precision of numeric data in terms tenths or hundredths is an example for formatting in a field. The prerequisite for computing the metric percentage of inconsistent format in a field is metadata that provides two specific details. The first detail is the column header for which percentage of inconsistent format in a field should be calculated and the other detail is the standard for formatting and defaulting the field. [15]

The formula to compute the percentage of inconsistent values in a field deduced from [1, 15] is shown below

[2] https://www.scipy.org/

$$\text{Percentage of inconsistent values} = \frac{Number\ of\ inconsistent\ values\ in\ the\ field}{Number\ of\ rows\ in\ the\ field} * 100$$

4.6 Interpretability

This section provides a brief description of the metrics percentage of spelling mistakes in section 4.6.1 and fit of training data in section 4.6.2 which are used to measure the data quality dimension interpretability [2].

4.6.1 Percentage of spelling mistakes

The metric percentage of spelling mistakes ensures the identification of spelling mistakes in a dataset. The spelling mistakes, grammar mistakes and abbreviations are categorized as noisy data. The metric percentage of spelling mistakes supports unstructured text data. The prerequisite for computing the metric percentage of spelling mistakes is the metadata that provides column header for which the percentage of spelling mistakes should be calculated. [2]

The library PyEnchant[3] described in chapter 6.1.11 or any other spelling correction modules may be used to compute the number of spelling mistakes. Then, the total number of words in the field is computed. [2]

4.6.2 Fit of training data

The metric fit of training data computes the similarity between two texts [56]. This metric supports unstructured text data and is related to the dimension interpretability [2]. The metadata such as the field for which percentage of similar data should be calculated and the reference document are essential to calculate the fit of training data [56].

Libraries such as DKPro Similarity[4] library may be used to compute text similarity measures such as cosine similarity between two texts. The obtained text similarity could be normalized between 0 and 1 where 0 indicates no similarity and 1 indicates high similarity. [2]

4.7 Relevancy

This section provides a brief description of the metric percentage of non-relevant data which is used to measure the data quality dimension relevancy [2].

4.7.1 Percentage of non-relevant data

The metric percentage of non-relevant data identifies the data which are not relevant in a dataset. The relevance of text data may be computed by using the existing approaches to compute relevance in information retrieval systems such as boolean retrieval described in [57]. The prerequisite for computing the metric percentage of non-relevant data is the metadata that provides two specific details. The first detail is the column header for which the percentage of non-relevant data should be calculated. The second detail is the set of keywords that are relevant for the specified column. [2]

[3] http://pythonhosted.org/pyenchant/

[4] https://dkpro.github.io/dkpro-similarity/

The formula to compute the percentage of non-relevant data in a field deduced from [2] is shown below

$$\text{Percentage of non relevant data} = 100 - Percentage\ of\ relevant\ data$$

$$\text{Percentage of relevant data } = Text\ similarity\ between\ fk\ and\ rk * 100$$

where fk = number of frequent keywords in the field
rk = number of relevant keywords

The steps to compute the metric percentage of non-relevant data in a field is illustrated below. First, the frequent keywords in the field are identified. Libraries such as Natural Language Processing Tool Kit[5] (NLTK) may be used to identify frequent keywords. Then, the frequent keywords in the field are compared with the keywords that are relevant for the field, to determine the text similarity between them. Libraries such as DKPro Similarity may be used to compare the frequent keywords in the field with the relevant keywords and compute text similarity. The obtained text similarity in the interval (0,1) is multiplied with 100 to obtain the percentage of relevant data in the field. This value is subtracted from 100 to calculate the percentage of non-relevant data in the field. [2]

[5] http://www.nltk.org/

5 CONCEPT

This chapter is based on the result from the literature study on existing data quality dashboards and metrics. A brief description on the high-level and low-level architecture of the dashboard is provided in section 5.1.

The objective of the thesis is to implement a dashboard that measures the data quality of structured and unstructured data accumulating in the Product Life Cycle (PLC). To achieve this, we develop the PLCDQ dashboard that accepts PLC data defined in chapter 2.4, measures the quality of PLC data and displays the data quality results. The dashboard accepts PLC data from the user in two standard file formats JSON and CSV. The dashboard uses the Data Quality (DQ) metrics *percentage of null values, percentage of duplicate values and percentage of non-domain values* to measure the quality of PLC data. These three metrics are implemented as web services. The dashboard also uses the metric *percentage of spelling mistakes* provided by the supervisor of this thesis to measure the quality of unstructured PLC data. The dashboard provides a flexible integration of new metrics to measure the quality of PLC data. In the Metric Interface described in chapter 6.2.3 a flexible integration of new metrics is implemented and as an example the metric *percentage of spelling mistakes* is integrated with the dashboard to show the flexible integration of new metrics in the dashboard. The dashboard displays the data quality results in tables, pie charts or donut charts and JSON files.

5.1 Architecture

This section provides a brief description of the high level generic architecture in the subsection 5.1.1, high level specific architecture in the subsection 5.1.2 and low level architecture in the subsection 5.1.3.

5.1.1 High Level Generic Architecture

The high level generic architecture of the PLCDQ dashboard is shown in Figure 12. The dashboard contains a web front end to accept input from the user, to send requests and to receive responses from the web services and to display the results. REST APIs are used to send requests from the web front end and receive responses from the web services. *Upload service, Convert service* and n number of *DQ Metric services* are implemented as REST web services. All the REST web services are deployed in an Application server.

The *Upload service* receives the request from the web front end, saves the PLC data file in the file repository and sends the response to the web front end. The *Convert service* receives the request from the web front end, converts the uploaded file to JSON file format to maintain a standard file format for DQ metrics to retrieve data, saves the JSON file in the file repository and sends the response to the web front end. All *DQ Metric services* receive the request from the web front end, retrieve data from the JSON file, measure the DQ of the data and send the response to the web front end. The uploaded file and the converted file are saved in the file repository as it is the most straight forward approach to save the file. All *DQ Metric services* retrieve data from the JSON file in the file repository.

5.1.2 High Level Specific Architecture

The high level specific architecture of the PLCDQ dashboard is presented in Figure 13. The AngularJS framework defined in chapter 6.1.1 is used to implement the web front end of the dashboard. REST APIs (defined in chapter 2.6) are used to send requests from the AngularJS and receive responses from the web services. Jersey REST web services framework (defined in chapter 6.1.8) is used to develop the *Upload service* to upload the data file, the *Convert service* to convert the uploaded file to JSON file format and the three *DQ Metric services* to

compute DQ of the data file as Jersey web services. These Jersey REST web services are deployed in Apache Tomcat Application Server (described in chapter 6.1.6).

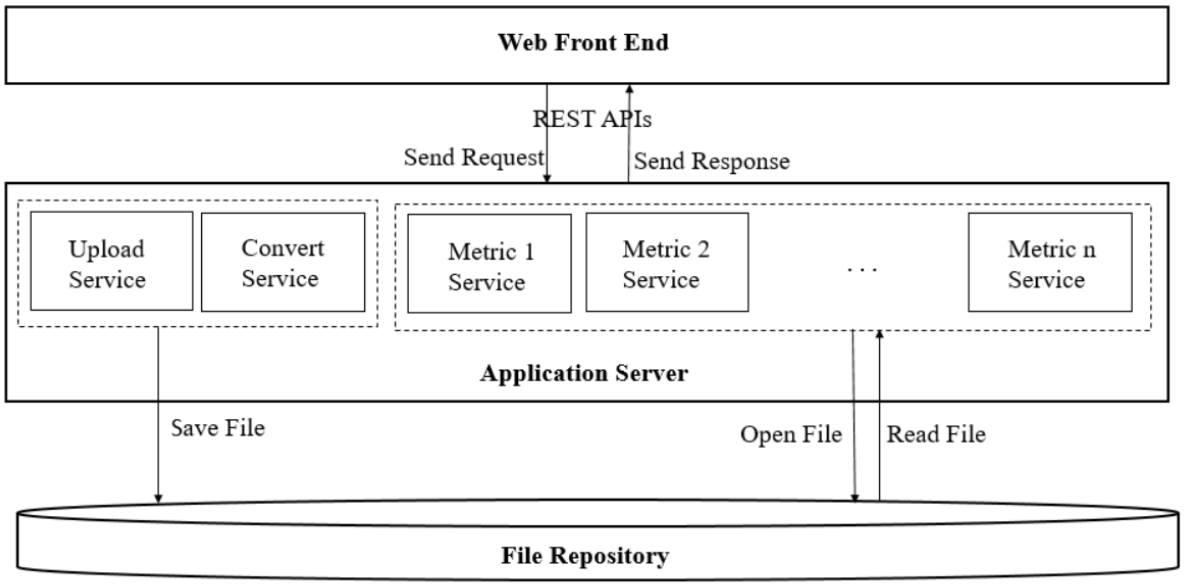


Figure 12:High Level Generic Architecture Diagram

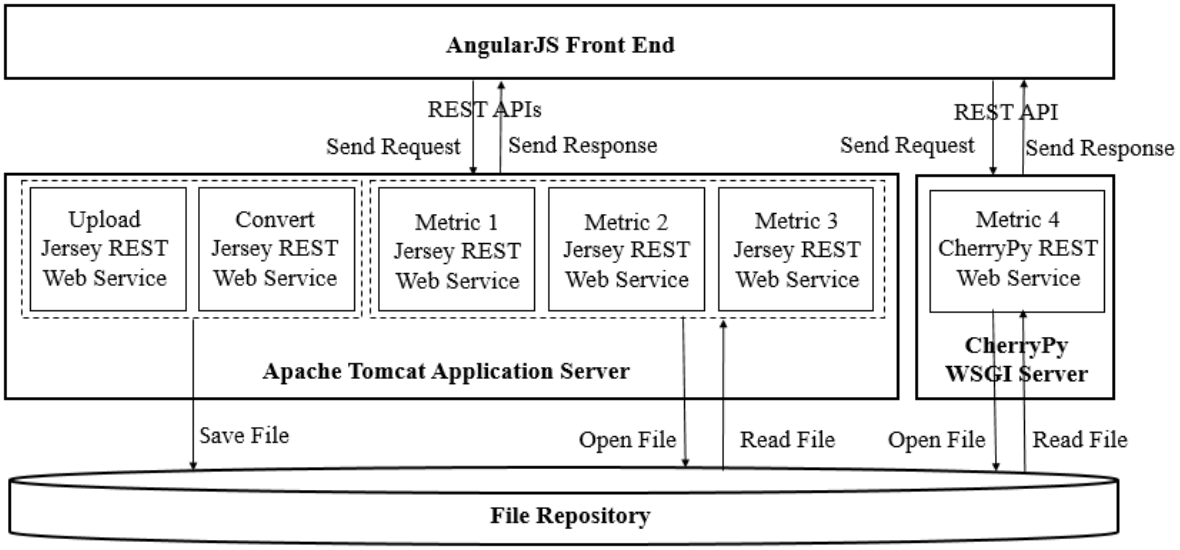


Figure 13:High Level Specific Architecture

The CherryPy framework defined in chapter 6.1.11 is used to implement the fourth *DQ metric service* as a REST web service. The CherryPy REST web service is deployed in CherryPy WSGI Server. The uploaded file and the converted JSON file are saved in the file repository. All the *DQ metric services* retrieve the PLC data from the JSON file saved in the file repository.

5.1.3 Low Level Architecture

The low-level architecture of the PLCDQ dashboard is illustrated in the Figure 14. The AngularJS front end contains the *Upload module, Null Value module, Duplicate Value module, Non-Domain Value module* and *Spelling Mistakes module*. The *Upload module* sends REST requests to the *Upload Jersey REST web service,* receives responses from the same, then sends REST requests to the *Convert Jersey REST web service* and receives responses from the same. The *Null Value module, Duplicate Value module, Non-Domain Value module* and *Spelling Mistakes module* invoke the *Metric Interface* (described in chapter 6.2.3) which acts as an interface to send requests and receive responses from the *Null Value REST web service, Duplicate Value REST web service, Non-Domain Value REST web service* and *Spelling Mistakes REST web service*.

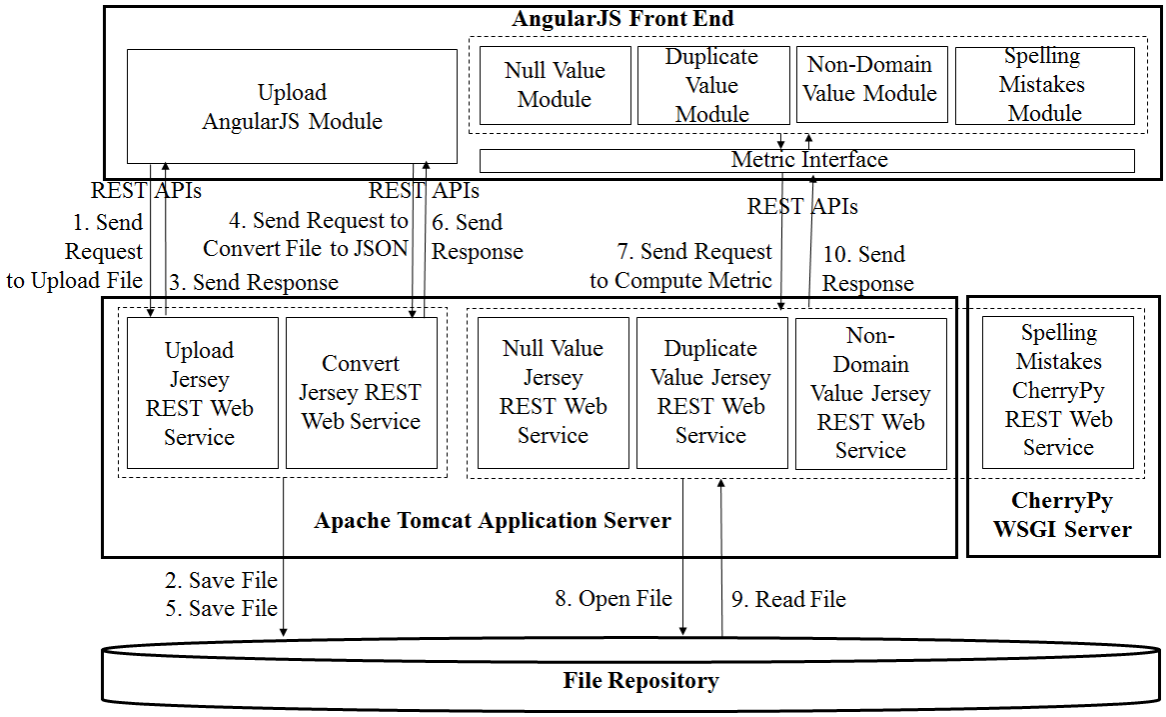


Figure 14:Low Level Architecture Diagram

The *Upload Jersey REST web service* receives the REST request from the *Upload module* in the AngularJS front end, saves the PLC data file in the file repository and sends the REST response to the *Upload module* in the AngularJS front end. The *Convert Jersey REST web service* receives the REST request from the *Convert module* in the AngularJS front end, converts the uploaded file to JSON file format, saves the JSON file in the file repository and sends the REST response to the *Convert module* in the AngularJS front end. The *Null Value Jersey REST web service, Duplicate Value Jersey REST web service, Non-Domain Value Jersey REST web service* and *Spelling Mistakes CherryPy REST web service* receives the REST request with the request parameters from the *Metric Interface* in AngularJS, retrieve data from a JSON file in the file repository, compute the respective metric and send the REST response to the *Metric Interface* in the AngularJS. The uploaded file and the converted JSON file are saved in the file repository which is a folder in the Apache Tomcat Server. All the DQ metric web services retrieve PLC data from the JSON file saved in the file repository.

6 IMPLEMENTATION

This chapter provides a detailed illustration on the implementation details of the PLCDQ dashboard. The libraries used for the implementation are described in section 6.1. The modules used to implement the web front end of the application are described in detail in the section 6.2. The implementation details of the web services are described in section 6.4. The final section 6.5 of this chapter presents the user interfaces of the PLCDQ dashboard.

6.1 Libraries

This section provides a brief description of the libraries used for implementation of the PLCDQ dashboard. The AngularJS library described in the section 5.1.1 is used for the development of web front end of the application. The libraries Data Driven Documents (D3) defined in the section 5.1.2, NVD3 defined in the section 5.1.3 and Angular-NVD3 defined in the section 5.1.4 are used to visualize the DQ in the form of pie and donut charts. The library Bootstrap defined in section 5.1.5 is used to design the user interface of the dashboard. The libraries CherryPy defined in section 5.1.7, Jersey defined in section 5.1.8 and JSON-Simple defined in section 5.1.9 are used in the implementation of the web services. The libraries Duplicate Detection Toolkit described in the section 5.1.10 and PyEnchant defined in the section 5.1.11 are used in the web services to compute DQ metrics.

6.1.1 AngularJS

AngularJS[6] is an open source JavaScript structural framework for building dynamic web applications. It uses Hyper Text Markup Language (HTML) as a template and allows to extend HTML's syntax to represent components in an application. AngularJS supports the complete client-side development of a web application such as UI development, development of business logic and testing. It decouples the client and server side of the application to enable parallel development and reuse of modules in both sides. It handles Document Object Model (DOM) and Asynchronous Javascript And Xml (AJAX) codes such as callbacks to simplify application development. [58]

6.1.2 Data Driven Documents (D3)

Data Driven Documents[7] (D3) is a JavaScript library that visualizes data using web standards such as HTML, Scalable Vector Graphics (SVG) and canvas. It binds data to a DOM and applies data-driven transformations to the DOM. This feature supports the creation of an HTML table or an SVG chart such as a bar chart using the same data. D3 supports large datasets and is extremely fast with minimal overhead. It also supports dynamic behaviour for animation and interaction. D3 provides a diverse variety of visuals ranging from basic charts such as line, bar and pie charts to visuals such as maps and flight visualizations. [59]

6.1.3 NVD3

NVD3[8] is a JavaScript library that builds reusable charts and chart components for D3 defined in 5.1.2. NVD3 library is dependent on D3 library. NVD3 supports the latest version of browsers such as Firefox, Google Chrome and Safari. NVD3 provides at least thirteen visualizations that include basic charts such as line, bar and pie charts. [60]

[6] https://angularjs.org/
[7] https://d3js.org/
[8] http://nvd3.org/

6.1.4 Angular-nvD3

Angular-nvD3[9] is an AngularJS directive for NVD3 library defined in chapter 6.1.3. This directive is dependent on the libraries AngularJS defined in chapter 6.1.1, D3 defined in chapter 6.1.2 and NVD3. Angular-nvD3 provides at least twenty basic visualizations such as line, bar and pie charts. The visualizations could be customized using JSON APIs. [61]

6.1.5 Bootstrap

Bootstrap[10] is an open source HTML, Cascading Style Sheets (CSS) and JavaScript framework for designing web applications. It simplifies the process of front-end web development. Bootstrap supports devices of all shapes such as mobiles, tablets and desktops. Bootstrap is used in millions of amazing websites such as Vogue[11] and Newsweek[12]. [62]

6.1.6 Apache Tomcat

Apache Tomcat[13] is an open source software that contains Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. This software is used to support many large-scale, mission-critical web applications in industries and organizations across the world. [63]

6.1.7 CherryPy

CherryPy[14] is an object-oriented web framework for Python. It allows development of web applications with Python programming language. This strategy reduces the length of the code which in turn reduces the time taken to develop the code. [64]

6.1.8 Jersey

Jersey[15] is an open source framework for developing RESTful web services (defined in chapter 2.6) in Java. This framework supports JAX-RS APIs and provides its own API that extends the JAX-RS toolkit to simplify the development of RESTful services and clients. [65]

6.1.9 JSON-Simple

JSON-Simple[16] is a Java library for JSON texts. This library is used to encode and decode or parse JSON formatted texts. JSON-Simple uses a heap based parser. The library is simple and easy to use. [66]

6.1.10 Duplicate Detection Toolkit

Duplicate Detection (DuDe) Toolkit[17] is an open source library developed by Hasso-Plattner-Institut to provide a system to compare duplicate detection algorithms. Duplicate detection algorithms identify duplicate pairs of records that represent the same real world entity. DuDe provides several algorithms, similarity measures, and datasets with gold standards that may serve as a benchmark for the comparisons. The toolkit consists of components such as data sources, preprocessor, algorithms, similarity functions, postprocessor and output functions with clear interfaces that makes it easy to use and extend. Algorithms detect pairs of records

[9] https://krispo.github.io/angular-nvd3/#/
[10] http://getbootstrap.com/
[11] http://www.vogue.de/
[12] http://europe.newsweek.com/
[13] http://tomcat.apache.org/
[14] http://cherrypy.org/
[15] https://jersey.java.net/
[16] https://github.com/fangyidong/json-simple
[17] https://hpi.de/naumann/projects/data-quality-and-cleansing/dude-duplicate-detection.html

that may be classified as duplicates or non-duplicates. Similarity functions compute the similarity between the identified pairs of records. [67]

6.1.11 PyEnchant

PyEnchant[18] is an open source python library used to check spelling mistakes in texts. This library provides all the functionalities of the Enchant[19] library along with the flexibility of python. [68]

6.2 Implementation Details of the Web Front End

This section provides a detailed description of the modules implemented in the AngularJS Web Front End. The functionalities of the Upload Module are described in the subsection 5.3.1, the Metric Modules used to compute DQ for specific columns in the data file are defined in the subsection 5.3.2, the Metric Interface used to provide flexible integration of new metrics is presented in the subsection 5.3.3. The subsection 5.3.4 illustrates the calculation of overall DQ for specific columns with priorities for the DQ metrics. The subsection 5.3.5 defines the procedure to visualize the data in the dashboard. The last subsection 5.3.6 of this chapter illustrates the steps to integrate new metrics into the PLCDQ dashboard.

6.2.1 Upload Module

The Upload module in the AngularJS Front end accepts a data file in CSV or JSON formats. To maintain the consistency of file formats for the computation of all DQ metrics, this module converts the CSV file to JSON file format. The converted JSON file is saved in the file repository. The overall data quality is calculated from the JSON file and is visualized as defined in chapter 6.2.5. This section contains a detailed illustration of the Upload module.

The Upload module sends a HTTP POST request to the Upload Jersey REST web service as shown in the Code Example 1. This request contains the *data file* to upload as a request parameter. The HTTP response from the Upload web service contains a message that indicates the execution status of the service.

```
/**Module to send post request to the Upload web service
   Request Parameters: URL, filename
   Response : status message **/
myApp.service('fileUpload', ['$http', function ($http) {
    this.uploadFileToUrl = function(file, uploadUrl, callback){
        var fd = new FormData();//fd contains file data
        fd.append('file', file);
        //HTTP POST Request
        $http.post(uploadUrl, fd, {
            transformRequest: angular.identity
        })
        .success(function(response){
            callback(response); //response on successful execution of the upload web service
        })
        .error(function(response){
            alert("Error in Upload Module") //Error message on failure of the upload web service
        });
    }
}]);
```

Code Example 1: POST Request from Upload Module

[18] http://pythonhosted.org/pyenchant/
[19] http://www.abisource.com/projects/enchant/

On successful execution of the Upload web service, the Upload module in AngularJS front end sends a HTTP POST request to the Convert Jersey REST web service. This request contains the *uploaded file name* as a request parameter. The HTTP response from the Convert web service contains a message that indicates the execution status of the service.

Automatic Calculation of Overall Data Quality

The overall quality of the data is calculated from the JSON file after the successful execution of the Convert web service. The Null Value REST web service described in chapter 6.4.1 and the Duplicate Value REST web service described in chapter 6.4.2 are used to compute the overall DQ. The request parameters *Uniform Resource Locator (URL)* and *file name* are assigned with URL of the Null Value web service and the uploaded file name. The Metric Interface described in chapter 6.2.3 is then invoked to compute the Null Value metric and its response that contains the percentage of null values received. The request parameter *URL* is assigned with the URL of the Duplicate Value REST web service and the Metric Interface is invoked once again to compute the Duplicate Value metric and its response that contains the percentage of duplicate values received. The overall data quality is computed from the percentage of null values and the percentage of duplicate values. Equal weights are used for the metrics Null Value and Duplicate Value to calculate the overall data quality.

6.2.2 Metric Modules

The Null Value module, Duplicate Value module, Non-Domain Value module and Spelling Mistakes module in AngularJS front end are the Metric modules that are invoked from the user interface described in chapter 6.5. The metric modules assign the request parameters *URL*, *file name* and *column name* with the URL of the corresponding DQ metric web service, uploaded data file name and column names provided by the user via user interface. Then the modules instantiate the *Metric interface* defined in chapter 6.2.3 to compute the corresponding metric and receive its response. The response is a JSON formatted string that contains a JSON array with attributes such as *name*, *count* and *percentage* for each column and attributes *total count* and *total percentage* for the entire dataset as shown in Figure 15. The response values are used to visualize the data in the form of tables and charts as described in chapter 6.5 and for preferential calculation of overall DQ as described in chapter 6.2.4.


```
{
"column":
[
      {
            "nullValuePercent":6.2,
            "columnIndex":"faildate",
            "nullValueCount":62
      }
],
"totNullPercent":6.2,
"totNullCount":62
}
```


Figure 15: Response from DQ Metric Web Service

6.2.3 Metric Interface

The Metric Interface is used for the flexible integration of new metrics to the PLCDQ dashboard. The interface is an AngularJS service that sends HTTP GET request to the DQ metric web services and receives HTTP response as shown in Code Example 2. In AngularJS, services are used to organize and share code across the application [69]. The *URL*, *file name*

and *column names* are the parameters of the HTTP GET request. The parameter *URL* represents the URL of a DQ metric web service described in chapter 6.4, *file name* represents the uploaded file name and *column names* represent the column names for which the metric should be computed. The HTTP response from the web service is a JSON formatted string that contains a JSON array with *count* and *percentage* of data that satisfy and do not satisfy the metric as shown in Figure 15. The response from the web service is send back to the Metric module that instantiated the Metric Interface.

```
/** Metric Interface to send get request to REST web services
 *  Request Parameters: URL, File Name, Column Name
 *  Response: JSON formatted string
 */
myApp.service('metricinterface', function($http) {
    this.getData = function(callbackFunc) {
        //HTTP GET request to compute DQ metric
        $http({
            method: 'GET',
            url: serviceURL, //contains the URL of the web service
            params:{filename: fileName, colnames: columnNames}
        }).success(function(data){
            callbackFunc(data);//send response to the Metric Module on successful execution
        }).error(function(){
            alert("Error in web service");//error message on failure of web service
        });
     }
});
```

Code Example 2: Metric Interface

6.2.4 User-defined Calculation of Overall DQ

The Metric Modules Null Value, Duplicate Value, Non-Domain Value and Spelling Mistakes module described in the chapter 6.2.2 are instantiated by the user with preferred columns via user interfaces described in chapter 6.2.3 and the DQ results are obtained. The DQ results from these modules are used to compute the overall data quality. The weights for the metrics null value, duplicate value, non-domain value and spelling mistakes are obtained from the user via user interface as described in chapter 6.5 and the overall DQ is computed from the DQ results with the respective weights. The computed overall DQ is visualized in the form of donut chart as described in chapter 6.2.5.

6.2.5 Visualizations

Donut charts and pie charts described are used in the PLCDQ dashboard to visualize the quality of data. Donut charts are used to visualize the overall DQ described in chapters 6.2.1 and 6.2.4 and the DQ results of Duplicate Value module, Non-Domain module and Spelling Mistakes module, which are calculated for specific columns defined by the user via the user interfaces described in chapter 6.5. Pie charts are used to visualize the DQ results of Null Value module calculated for specific columns defined by the user via user interface.

The library Angular-nvD3 defined in chapter 6.1.4 is used to visualize data in the form of pie and donut charts. Angular-nvD3 is dependent on the libraries D3 defined in chapter 6.1.2 and NVD3 defined in chapter 6.1.3. The Code Example 3 is used to visualize data in the form of a donut chart. The Bootstrap library defined in chapter 6.1.5 is used to display the DQ results in tables.

```
<div>
    <p><nvd3 options="donutOptions" data="overallQuality"></nvd3></p>
</div>
```

Code Example 3: Code to Visualize Data

6.3 Instructions to Integrate New Metrics

The steps to integrate new metrics to the PLCDQ dashboard are illustrated in this section.

1. Implement a REST web service to compute the new metric. The web service should accept the parameters *file name* and *column names*, and provide a JSON formatted string as response. The parameter *file name* represents the uploaded file name and *column names* represent the column names for which the DQ should be computed. The response should contain a JSON array with *count* and *percentage* of the values that satisfy and do not satisfy the metric.

2. Create a metric module described in chapter 6.2.2 in the AngularJS front end that assigns request parameters *URL* and *column names* with the URL of the web service created in step 1 and the column names obtained from user via the user interface created in step 3, invokes the Metric Interface and obtains the response from the Metric Interface.

3. Create a user interface as shown in chapter 6.5 that accepts the parameter *column names* from the user, triggers the metric module defined in step 2 and visualizes the DQ results obtained in step 2 in the form of charts and tables as described in chapter 6.2.5.

6.4 Web Services

The section Web Services provides a brief description on the implementation details of the web services *Null Value* in the subsection 5.4.1, *Duplicate Value* in the subsection 5.4.2, Non-*Domain Value* in the subsection 5.4.3 and *Spelling Mistakes* in the subsection 5.4.4. The subsection 5.4.5 describes the functionalities of the *Convert to JSON file* web service in detail.

All the web services retrieve the request parameters file name and column names separated by comma from the HTTP request. The JSON-Simple parser defined in chapter 6.1.9 is used to parse the data from the JSON file in the file repository.

6.4.1 Null Value Web Service

Null value web service calculates the null value metric. Code Example 4 shows the code for the calculation of the null value metric. For each row, each specified column is verified for null value using default String operations in Java and is counted if present. The rows identified to contain null values are documented in a JSON file. The percentage of null values for each specified column, the total count and the percentage of null values is calculated. The column name, count and percentage of null values in each column, the total count and percentage of null values are send as JSON response.

```
//For loop to iterate through each record in the JSON file
for (Object row : rows)
{
    JSONObject rowData = (JSONObject) row;

    //For loop to iterate through each selected column
    for (int i=0;i<columnNames.length;i++)
    {
        String value = (String) rowData.get(columnNames[i]);
        if(value.isEmpty() || value.equalsIgnoreCase("NULL") || value.equals("N/E"))
        {

                //Counts the null values in each selected column
                nullValues[i]+=1;

                //Counts the total number of null values
                totalNullValues+=1;

                //Write row to JSON File
                fw.write(rowData.toJSONString());
                fw.write(",");
        }

    }
}
```

Code Example 4: Computation of Null Value Metric

6.4.2 Duplicate Value Web Service

Sorted Neighbourhood algorithm and Levenshtein Distance similarity function from the DuDe library described in chapter 6.1.10 are used to compute similarity. Code Example 5 shows the code for the calculation of the duplicate value metric.

```
    for (DuDeObjectPair pair : algorithm) {

        double similarity = similarityFunction.getSimilarity(pair);

        // Notifies the algorithm whether the pair is categorized as a duplicate or a non-duplicate
        if (similarity > similar)
        {
            algorithm.notifyOfLatestComparisonResult(DuplicateCountSNM.ComparisonResult.DUPLICATE);
            ++duplicateRecordCount;
        }
        else
        {
            algorithm.notifyOfLatestComparisonResult(DuplicateCountSNM.ComparisonResult.NON_DUPLICATE);
        }
    }
```

Code Example 5: Computation of Duplicate Records

The algorithm detects DuDe object pairs, which are pairs of records that may be classified as duplicates or non-duplicates. The similarity function computes the similarity between the identified pair of records for each DuDe object pair. [67]

If the similarity is more than 0.75 then the pair is considered as duplicate and is counted. The percentage of duplicate values is calculated. The count and percentage of duplicate values and non-duplicate values are send as JSON response. The rows that are identified to contain duplicate records are also documented in a JSON file.

6.4.3 Non-Domain Value Web Service

Non-Domain value web service stores the data values from the selected column in a list. The unique values from the list and its respective count are computed and are stored in a hash map. The domain values from the domain file are stored in a hash set.

The unique values in the hash map are compared with the values in the domain set. If the comparison fails, the value is considered as non-domain value and is counted and the percentage of non-domain values is calculated. The count and percentage of non-domain values and domain values are send as JSON response and the rows that are identified to contain non-domain values are documented in a JSON file.

6.4.4 Spelling Mistakes Web Service

The Spelling Mistakes web service is implemented from a metric provided by the supervisor of the thesis. It is implemented in python and accepts a text string as input, uses the PyEnchant library described in chapter 6.1.11 to identify spelling mistakes and provides the percentage of spelling mistakes as output. This metric was extended to retrieve data from a JSON file, calculate the percentage of spelling mistakes for a specified column in each row, the total percentage of spelling mistakes and non-spelling mistakes in a column. The spelling mistakes web service accepts the request parameters *file name* and *column name*. The percentage of spelling mistakes and non-spelling mistakes are send as JSON response.

6.4.5 Convert to JSON file Web Service

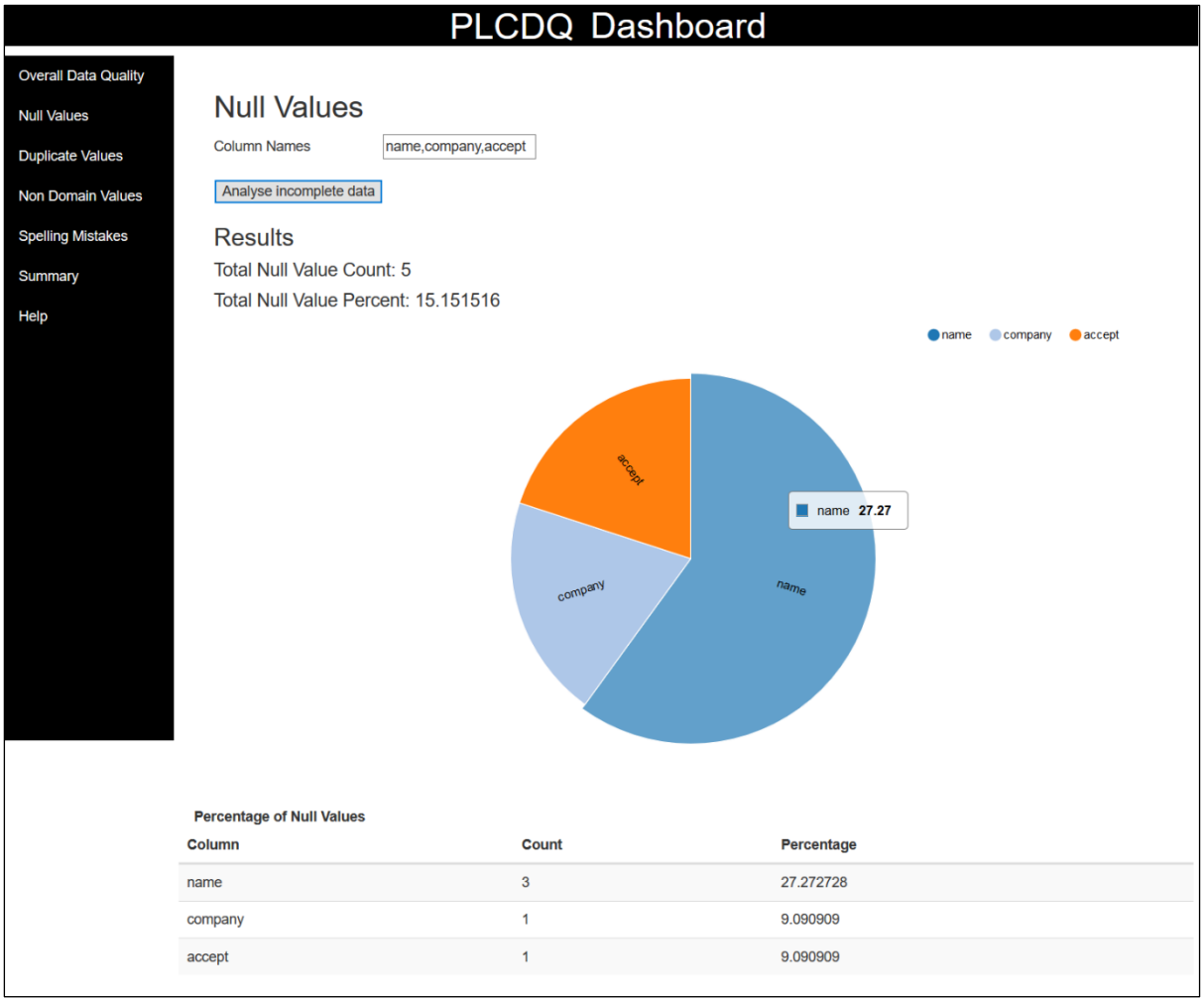
The Convert to JSON file web service is used to convert a dataset in CSV file format to JSON file format. The web service accepts the *file name* which represents the name of the file as an input parameter. If the file extension of the *file name* is in CSV format, the web service retrieves the data from the file and converts it to a byte array stream. Each row value in the dataset is retrieved from the byte array by using the default String operations in Java. Each column value is retrieved from the row value again by using the default String operations in Java. The extracted column values are appended to a String builder as a key-value pair with column header as the *key* and the column value as the *value*. A JSON file is created with the same *file name* and the contents in the String builder are written to the JSON file and the file is closed. The response of the web service is a success message in String format.

6.5 User Interface

This section provides the screenshots of the user interfaces in the PLCDQ dashboard for calculating user-defined data quality and the metric null value. The screenshots of the user interfaces to calculate the overall data quality and the metrics non-domain values and spelling mistakes are provided in appendix. In all the user interfaces, the legends of the visualizations appear on mouse over.

Null Values

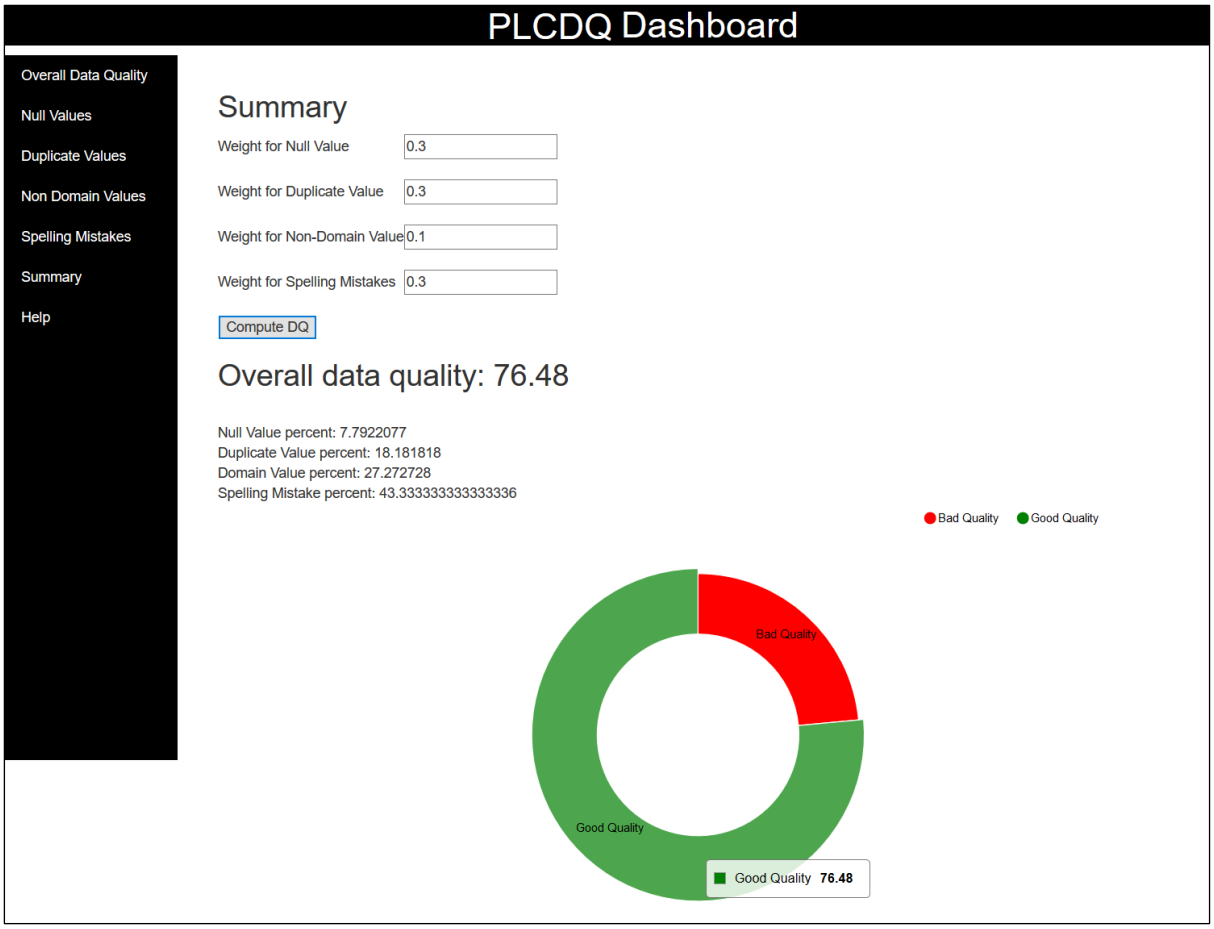
The user interface to calculate the Null Value metric for specific columns is shown in the Screenshot 1. The user needs to enter the desired columns separated by a comma in the *Column Names* text box and click the *Analyse incomplete data* button. This triggers the Null Value module described in the chapter 6.2.2 and visualizes the data quality of the null value metric in the form of a pie chart.



Screenshot 1: User Interface to Compute Null Value Metric

User-defined Data Quality

The user interface to compute the preferential data quality as described in chapter 6.2.4 in the dashboard is shown in Screenshot 2. User needs to enter the weights for the metrics in the text boxes *Weight for Null Value, Weight for Duplicate Value, Weight for Non-Domain Value* and *Weight for Spelling Mistakes* and click the *Analyse Spelling Mistakes* button. This triggers the module that calculates the overall data quality and visualizes it in a donut chart.



Screenshot 2: User Interface for Preferential Data Quality

7 DEMONSTRATION AND EVALUATION

This chapter demonstrates and evaluates the dashboard with four different datasets. The datasets used in this chapter are NHTSA consumer complaints dataset, DuDe restaurant dataset, a Twitter and a News dataset. A short description of these datasets is provided in section 7.1 and the demonstration on the evaluation of the dashboard with these datasets are provided in section 7.2.

7.1 Datasets

The four datasets used for evaluating the dashboard are NHTSA consumer complaints dataset which is an example of PLC dataset, DuDe restaurant dataset, a Twitter and a News dataset. This section contains a brief description of the datasets.

NHTSA Consumer Complaints Dataset

The NHTSA consumer complaints dataset provides traffic safety-related defect complaints received by NHTSA since 01 January, 1995. The National Highway Traffic Safety Administration (NHTSA) does research programs for the United States Department of Transportation to reduce vehicle crashes. The dataset is structured in rows and columns format with 49 fields that provide data such as the vehicle type, consumer details, accident details and information on vehicle dealer. The size of the dataset is approximately 800 MB. [70]

DuDe Restaurant Dataset

The DuDe restaurant dataset provides information such as the name, location and classification of restaurants from the Fodor's[20] and Zagat's[21] restaurant guides. The DuDe toolkit described in chapter 6.1.10 provides this dataset to verify the accuracy of the duplicate detection algorithms. This is a non-PLC dataset but contains the annotation of duplicates. The restaurant dataset contains 864 records of which 112 records are duplicates. The dataset is in csv format. The size of the restaurant dataset is approximately 69 KB. [67]

Twitter Dataset

The Twitter dataset contains tweets in the form of unstructured text data. The dataset was created as a part of developing a special part-of-speech tagger for Twitter data. This dataset contains 547 rows with a column *Tweet.* This column contains twitter messages from 01.01.2011 to 30.06.2012 with one tweet per day. The dataset is in csv format. The size of the dataset is 41 KB. [71]

News Dataset

The News dataset provides news in the form of unstructured text data. The dataset is from a contribution to the Conference on Computational Natural Language Learning in the year 2000. The dataset comprises of a single column *News* with 10948 rows. The column *News* belongs to conll corpus that contains data from parts of the Wall Street Journal corpus. The size of the dataset is 1,421 KB. [72]

7.2 Analysis of the Quality of the Datasets with the Dashboard

This section evaluates the quality of the datasets mentioned in section 7.1. The NHTSA consumer complaints dataset is used to analyse the overall data quality based on the total

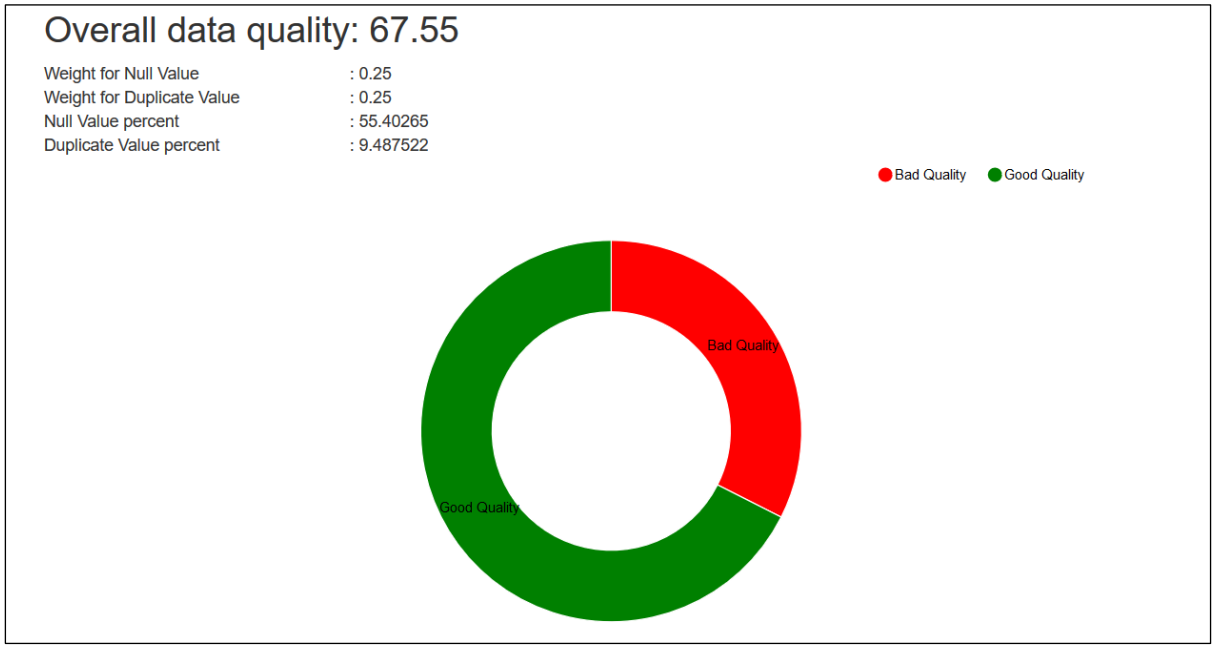
[20] http://www.fodors.com/

[21] https://www.zagat.com/

percentage of null values and duplicate values in all fields, percentage of null values and duplicate values for specified columns, percentage of non-domain values and spelling mistakes for a specified column and summary on data quality for the specified columns. The DuDe restaurant dataset is used to analyse the overall data quality and the accuracy of the computation of duplicate values in the dashboard. The Twitter and News Dataset are used to analyse the overall data quality and to illustrate the metric percentage of spelling mistakes.

7.2.1 Analysis of the NHTSA Consumer Complaints Dataset

A random sample containing 5% of NHTSA dataset in CSV format is used for the analysis because the NHTSA is a very large dataset of nearly 800 MB and so the PLCDQ dashboard could not support the whole dataset. We start our analysis in the *Overall Data Quality* page of the dashboard. The dataset is uploaded and the overall data quality of the NHTSA dataset is calculated. The metrics used to calculate the overall data quality are described in chapter 6.2.1. Screenshot 3 provides the overall data quality of the NHTSA dataset.

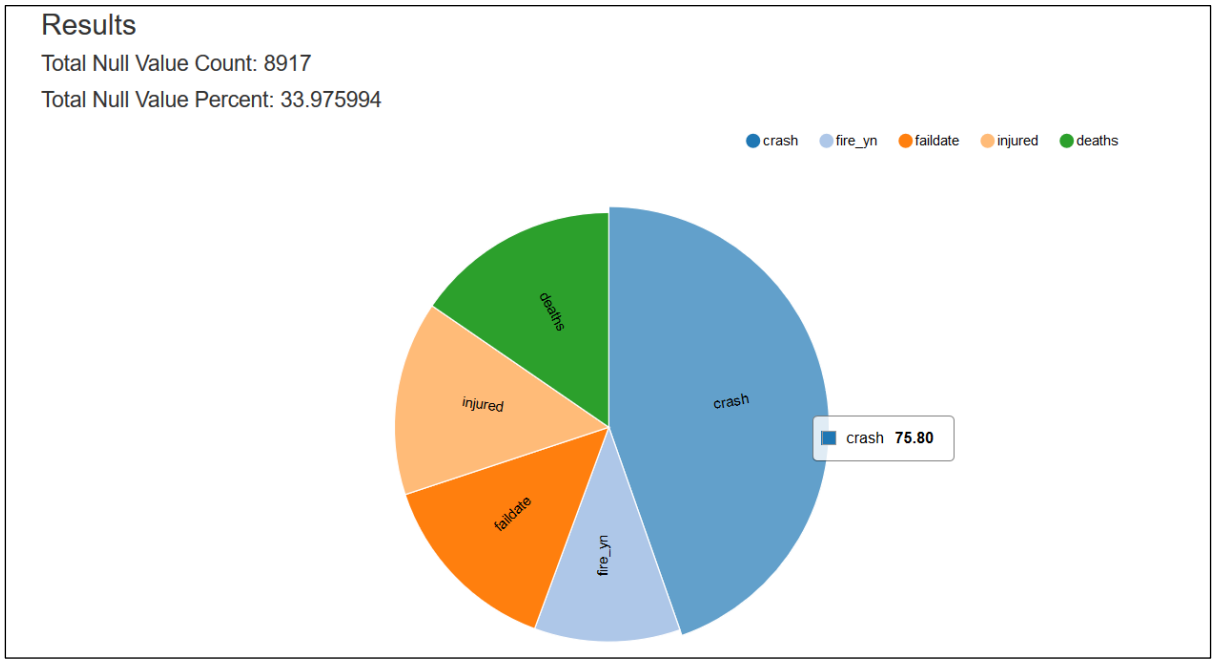


Screenshot 3:Overall Data Quality of NHTSA Dataset

The overall data quality of the dataset based on the metrics percentage of null values and percentage of duplicate values is 67.55 percent which is provided at the top of Screenshot 3. The overall data quality is calculated based on these two metrics because they could be applied for all columns. The percentage of null values taking all fields into account in the dataset is 55.40 percent, which implies that almost half of the dataset contains null values. With this information, we could not conclude that if the dataset could be used for further analysis, thus it may be necessary to calculate the percentage of null values for specific non-nullable fields in a next step. The percentage of duplicate values in the dataset is 9.48 percent, we may want to consider the percentage of duplicate data for the combination of specific fields which may not contain duplicate values before using the data for further analysis.

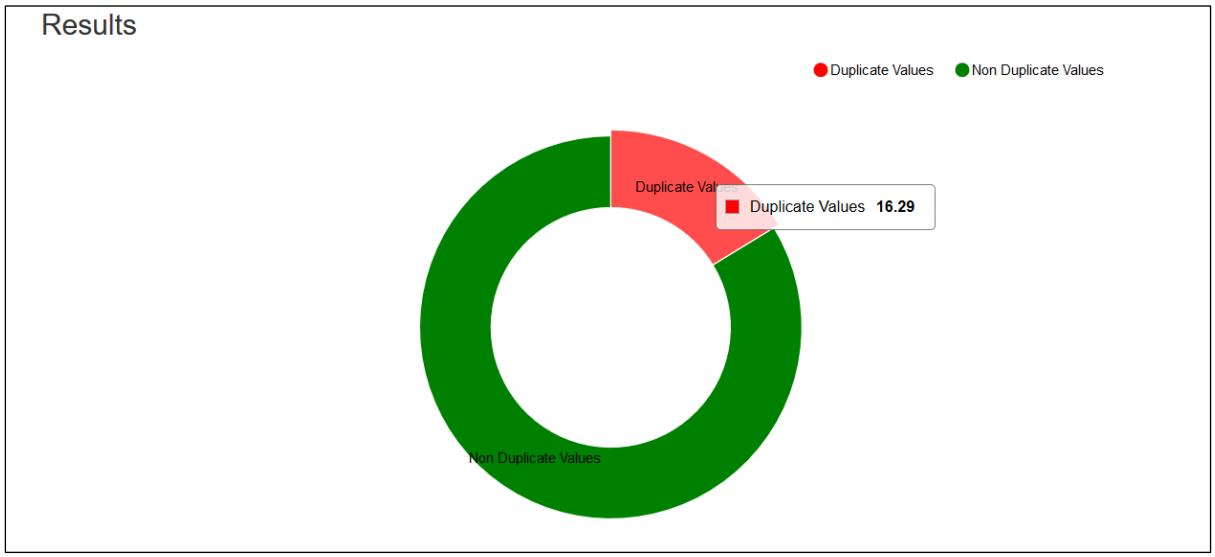
The percentage of null values in the critical fields could be calculated in the *Null Values* page in the dashboard. The fields *crash, faildate, fire_yn, injured* and *deaths* are considered for computing the percentage of null values. The *crash* field specifies if the vehicle was involved

in a crash, *faildate* field provides the date of the incident, *fire_yn* field specifies if the vehicle was involved in a fire, *injured* field indicates the number of persons involved in the accident and *deaths* field indicates the number of fatalities [70]. The Screenshot 4 shows the percentage of null values in these critical fields.



Screenshot 4:Percentage of Null Values in Critical Fields of NHTSA Dataset

In the dataset, 75.8 percent of the field *crash*, 18.68 percent of the field *fire*_yn, 24.29 percent of the field *faildate*, 24.88 percent of the field *injured* and 26.21 percent of the field *deaths* contain null values.

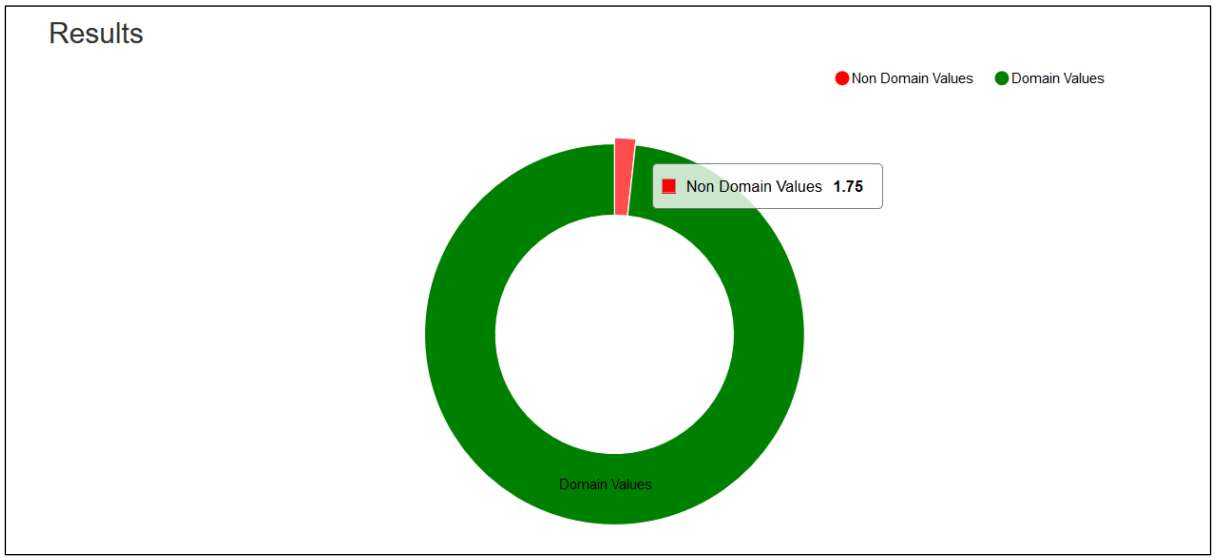


Screenshot 5: Percentage of Duplicate Values in Critical Fields of NHTSA Dataset

We continue the analysis by measuring the percentage of duplicate values in critical fields such as *faildate, state, city, mfr_name, make, model, compdesc* and *cdescr* which contain details of the vehicle, location, date and description of the accident. If these fields are duplicated, there is a high possibility that the records are duplicate. Screenshot 5 shows details on the percentage

of duplicate values based on these columns. The percentage of duplicate values in the critical fields is 16.29 percent.

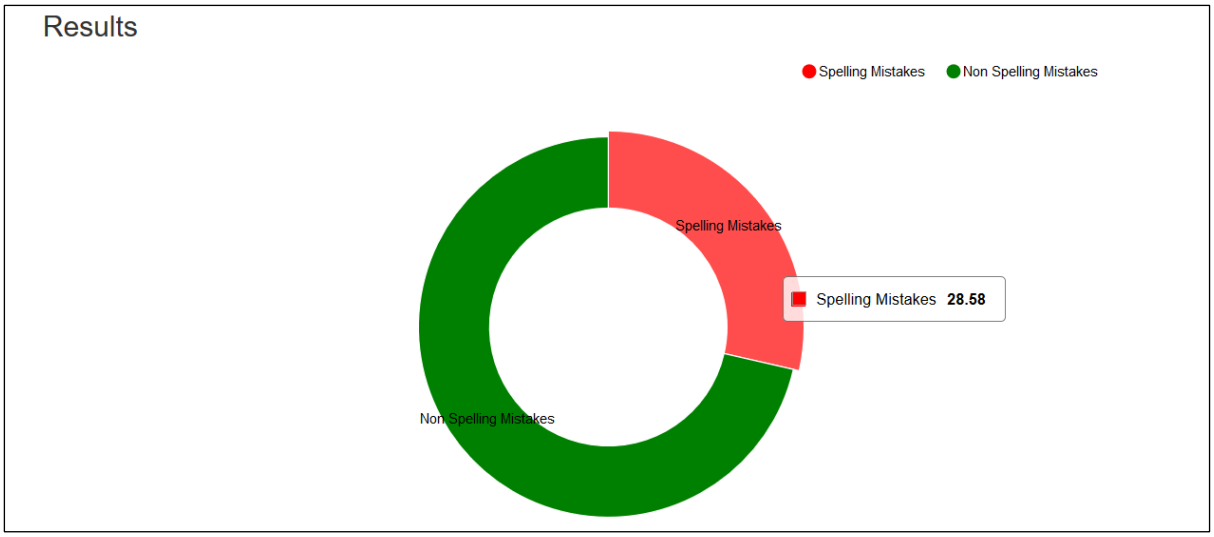
We proceed the analysis on *Non-Domain Values* in the user interface described in chapter 6.5. The domain values described in chapter 4.4.1 are uploaded as a text file. The domain values that we consider for the analysis are V, T and E, where V represents vehicle, T represents tires and E represents Equipment. In the NHTSA dataset, the field *prod_type* specifies the product type codes which has these domain values. We use the field *prod_type* to calculate the percentage of non-domain values. The Screenshot 6 is the result of the analysis of the Non-Domain Values in the dataset. The *prod_type* field contains 1.75 percentage of non-domain values. The rows affected are documented in a file from which more details on the data quality problems could be gained.



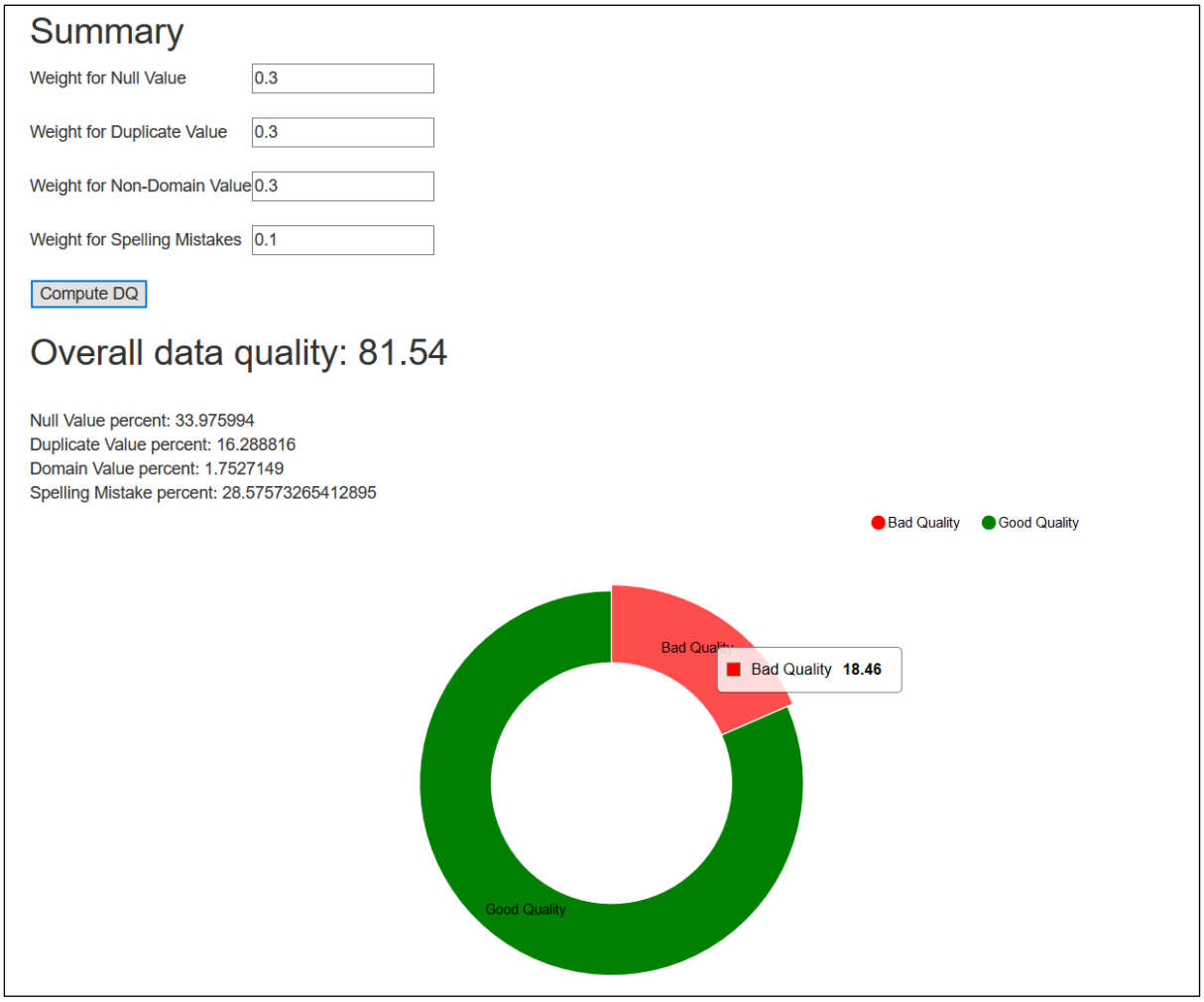
Screenshot 6: Non-Domain Values in Critical Fields of NHTSA Dataset

The *Spelling mistakes* page calculates the spelling mistakes of a column as described in chapter 6.4.4. The text fields such as *compdesc* and *cdescr* contain the description of components and accidents, that may contain spelling mistakes. The field *compdesc* is used for the analysis and the results are shown in Screenshot 7. This field contains 28.58 percentage of spelling mistakes.

We move to the *Summary* user interface described in chapter 6.5 for getting an overview on the data quality of critical fields. We assign equal weights of 0.3 to the metrics Null Values, Duplicate Values, Non-Domain Values and 0.1 to the Spelling Mistakes metric. Screenshot 8 shows the summary of data quality on critical fields in the NHTSA dataset. The overall data quality of critical fields is 81.54 percent.



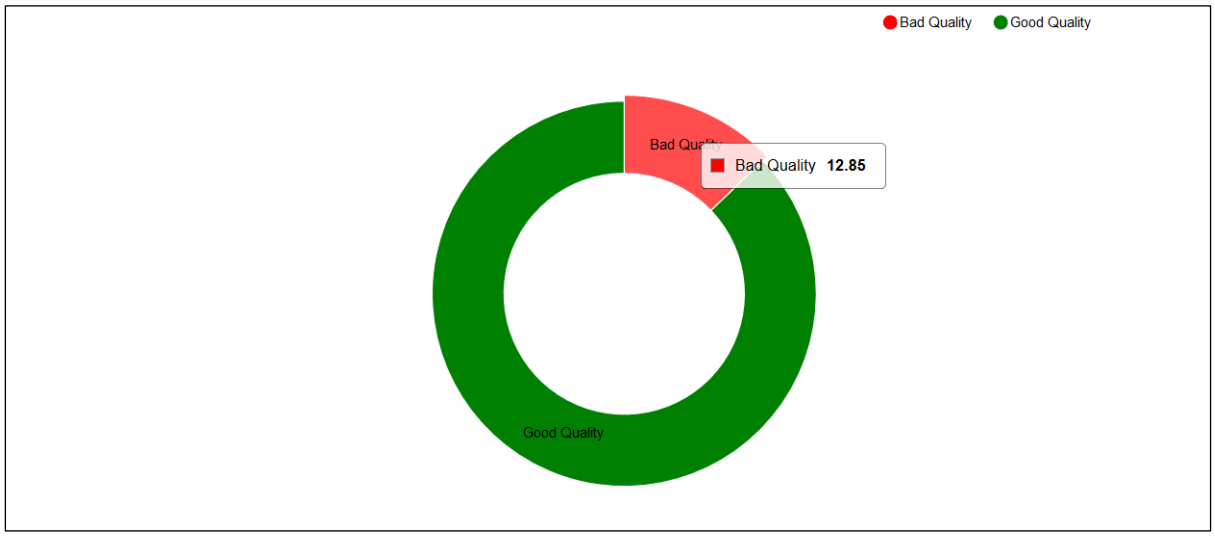
Screenshot 7: Percentage of Spelling Mistakes in Critical Fields of NHTSA



Screenshot 8: Summary of Data Quality of Critical Fields in NHTSA Dataset

7.2.2 Analysis of DuDe Restaurant Dataset

The complete DuDe restaurant dataset comprising of 864 rows and 7 columns is used for the experimentation. The critical fields to detect duplicate records in this dataset are *name* and *city* that provides the name and city of the restaurants. We assume that a city may not have more than one restaurant with the same name to calculate the duplicate records. The percentage of duplicate values in the critical fields *name* and *city* are calculated in the Duplicate Values page of the dashboard and is shown in Screenshot 9.



Screenshot 9:Duplicate Values in DuDe Restaurant Dataset

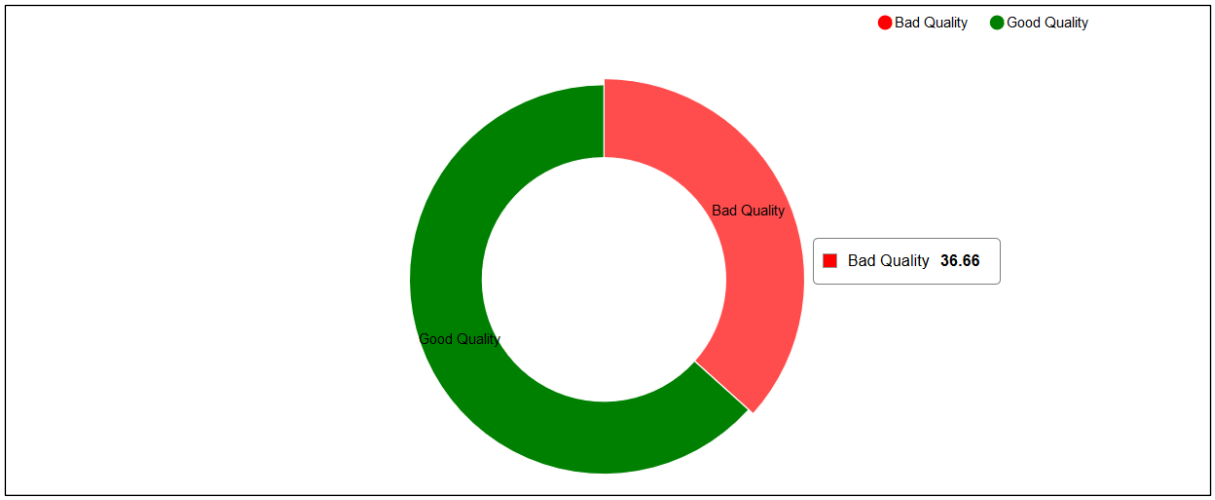
Duplicate values with combination of critical fields *name* and *city* computed by the dashboard contains 111 duplicates, that is 12.85 percentage of the dataset contains duplicate records. The DuDe restaurant dataset contains 112 duplicate records [67]. The precision of the sorted-neighbourhood algorithm that is used to detect duplicate values is nearly 86 percent [73].

7.2.3 Analysis of Twitter Dataset

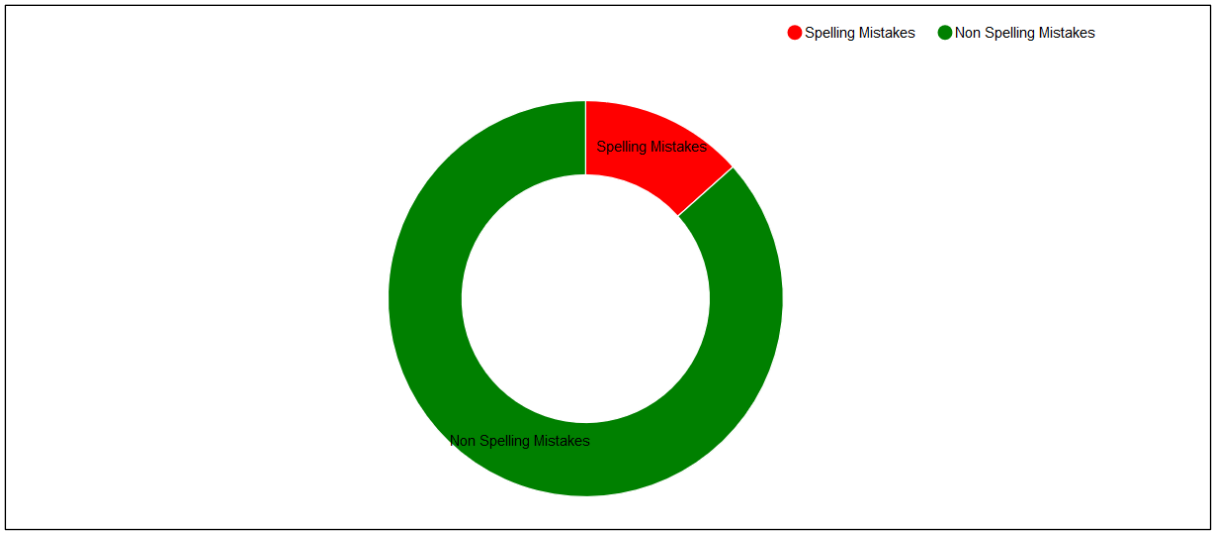
The complete Twitter dataset provided by the supervisor with 547 rows is used for the experimentation. The Twitter dataset contains twitter messages in the form of unstructured text data in the column *Tweet* which may contain spelling mistakes. The percentage of spelling mistakes in the column *Tweet* is computed in the *Spelling Mistakes* page of the dashboard and is shown in Screenshot 10. The column *Tweet* contains nearly 36 percentage of spelling mistakes, that implies one third of the dataset contains spelling mistakes.

7.2.4 Analysis of News Dataset

The complete News dataset provided by the supervisor of the thesis with 10948 rows is used for the experimentation. The field *News* contains news in the form of unstructured text data that may contain spelling mistakes. The percentage of spelling mistakes in the column *News* is computed in the *Spelling Mistakes* page of the dashboard and is shown in Screenshot 11. The percentage of spelling mistakes in the column *News* is 13.40 percent, this implies more than one tenth of the dataset contains spelling mistakes. Based on the analysis of Twitter and News datasets, it is clear that the Twitter dataset contains a higher percentage of spelling mistakes compared to the News dataset.



Screenshot 10: Overall Data Quality of Twitter Dataset



Screenshot 11: Overall Data Quality of News Dataset

8 CONCLUSION AND FUTURE WORK

Existing data quality metrics and data quality dashboards were analysed and documented in the thesis report. The PLCDQ dashboard is also compared with the existing data quality dashboards and the results are presented. On an overview on existing DQ dashboards and metrics, valuable (new) features of a DQ dashboard and useful DQ metrics were identified in Table 1, Table 3 and described in chapter 4.

The PLCDQ dashboard is implemented during the course of this thesis. The metrics *percentage of null values, percentage of duplicate values* and *percentage of non-domain values* for computing the quality of structured data were implemented as REST web services for the dashboard. An interface was developed to provide flexible integration of the metrics with the dashboard. The metric *percentage of spelling mistakes* provided by the supervisor of the thesis for computing the quality of unstructured text data is integrated with the PLCDQ dashboard using the developed interface.

The PLCDQ dashboard could accept datasets from the user in CSV and JSON format, compute the overall data quality of the dataset based on the metrics *percentage of null values* and *percentage of duplicate values* as default and visualize the quality of data in the form of donut charts. Quality of the dataset based on metrics *percentage of null values, percentage of duplicate values, percentage of non-domain values* and *percentage of spelling mistakes* could be determined for specific columns and could be visualized. A summary of the data quality calculations for the individual metrics could be obtained and visualized in the form of a donut chart.

The implemented PLCDQ dashboard is demonstrated and evaluated with the NHTSA, DuDe restaurant, Twitter and News datasets. The results obtained from the evaluations of these datasets shows that NHTSA dataset contains nearly 50 percent of null values and News dataset contains less number of spelling mistakes compared to Twitter dataset.

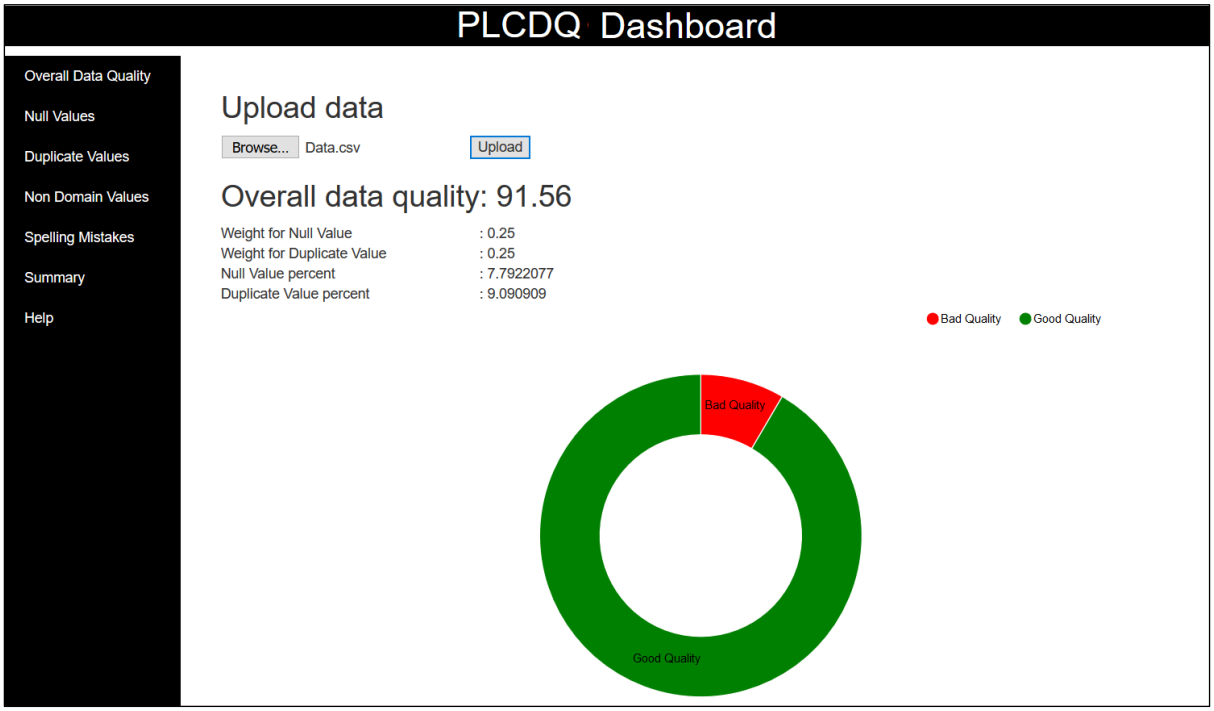
The future work of this thesis may be to improve the PLCDQ dashboard to accept data from large datasets like big data. To achieve this, the uploaded data files may be stored in databases. The performance of the dashboard should also be considered while dealing with large datasets. New data quality metrics could be integrated with the PLCDQ dashboard by using the implemented metric interface.

9 APPENDIX

The user interfaces of the PLCDQ dashboard are provided in this chapter.

Overall Data Quality

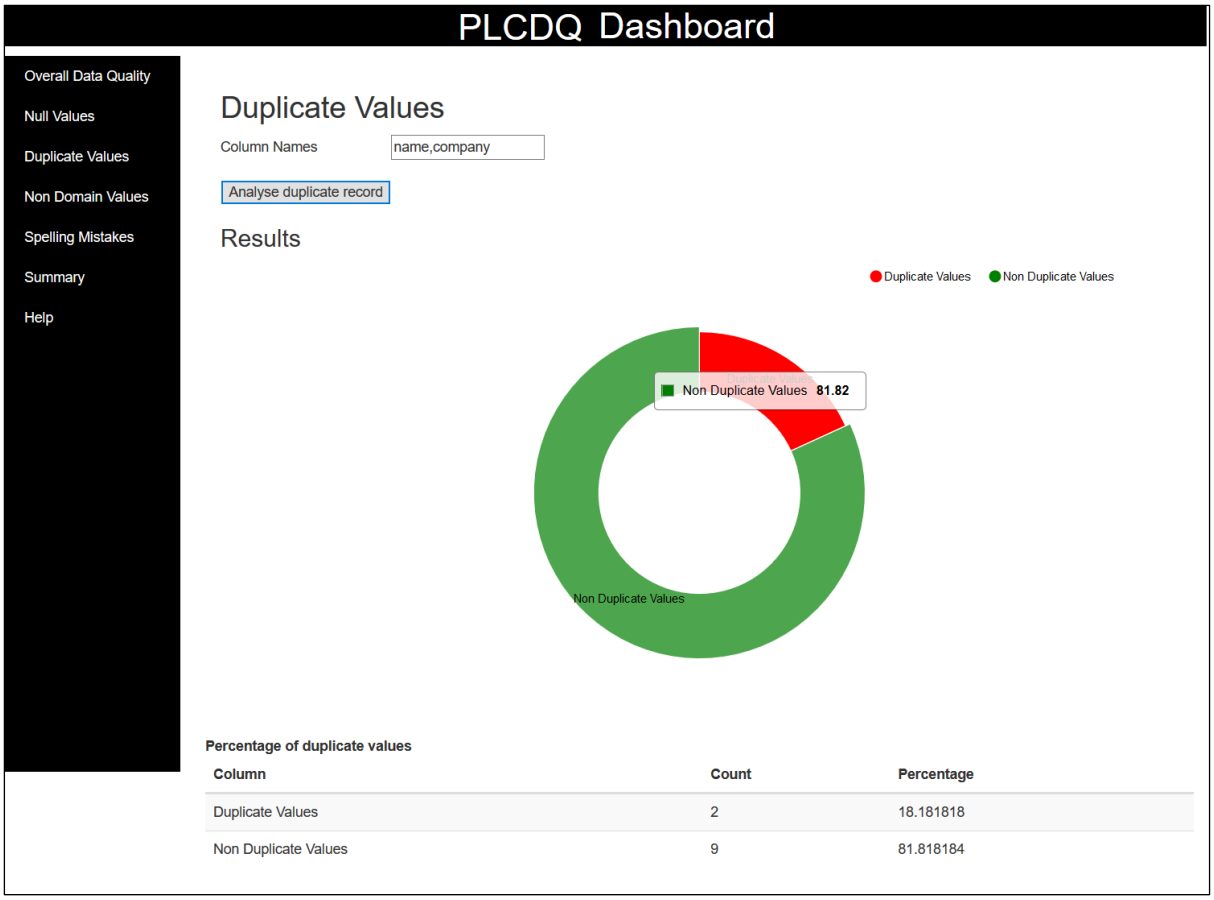
The user interface to compute the overall data quality in the PLCDQ dashboard is shown in the Screenshot . The user needs to choose a file and click the *Upload* button. The *Upload* triggers the Upload Module described in the chapter 6.2.1 and visualizes the overall data quality in a donut chart.



Screenshot 13: User Interface to Compute Overall Data Quality

Duplicate Values

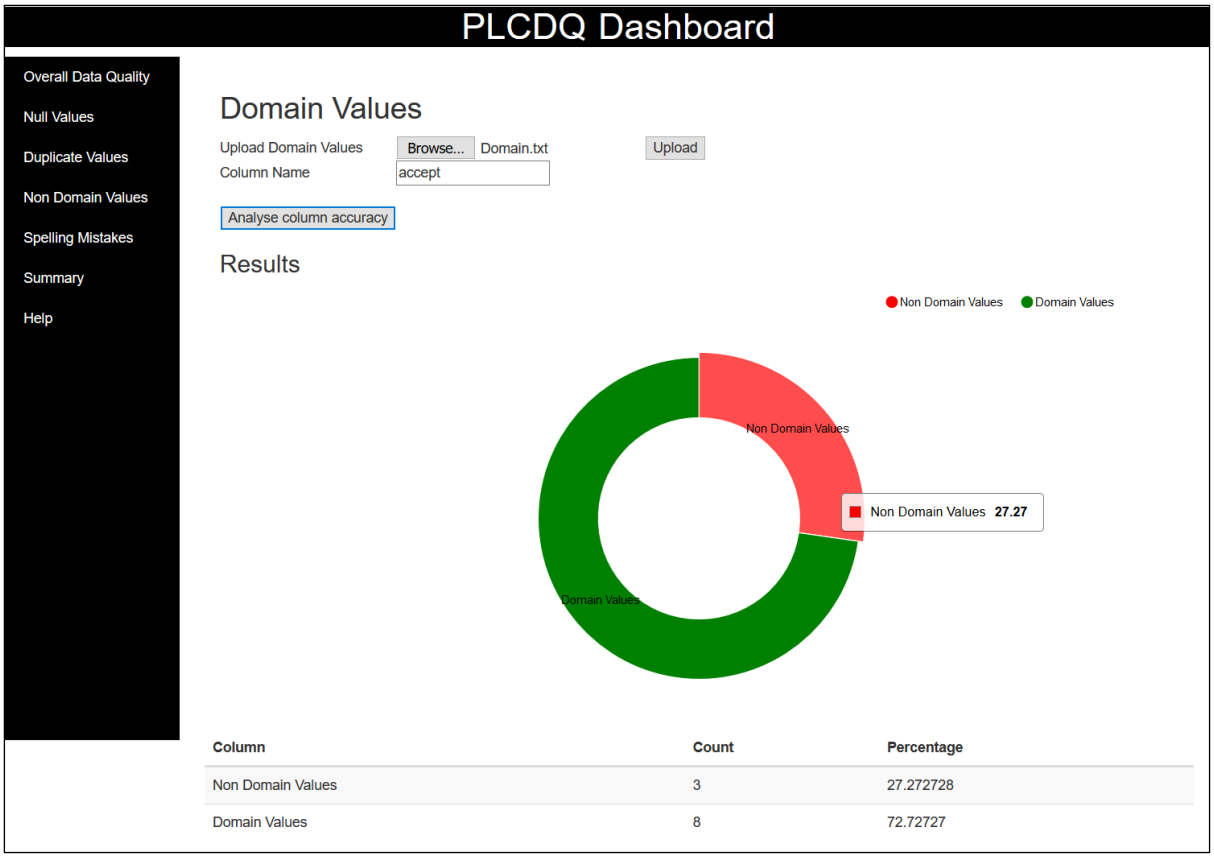
The user interface to compute the duplicate value metric in the dashboard is shown in the Screenshot . The user needs to enter the desired combination of columns separated by comma in the *Column Names* text box and click the *Analyse duplicate record* button. This triggers the Duplicate Value module described in chapter 6.2.2 to compute the percentage of duplicate values for the desired columns in the dataset and visualize it in donut chart.



Screenshot 14: User Interface to Compute Duplicate Records

Non-Domain Values

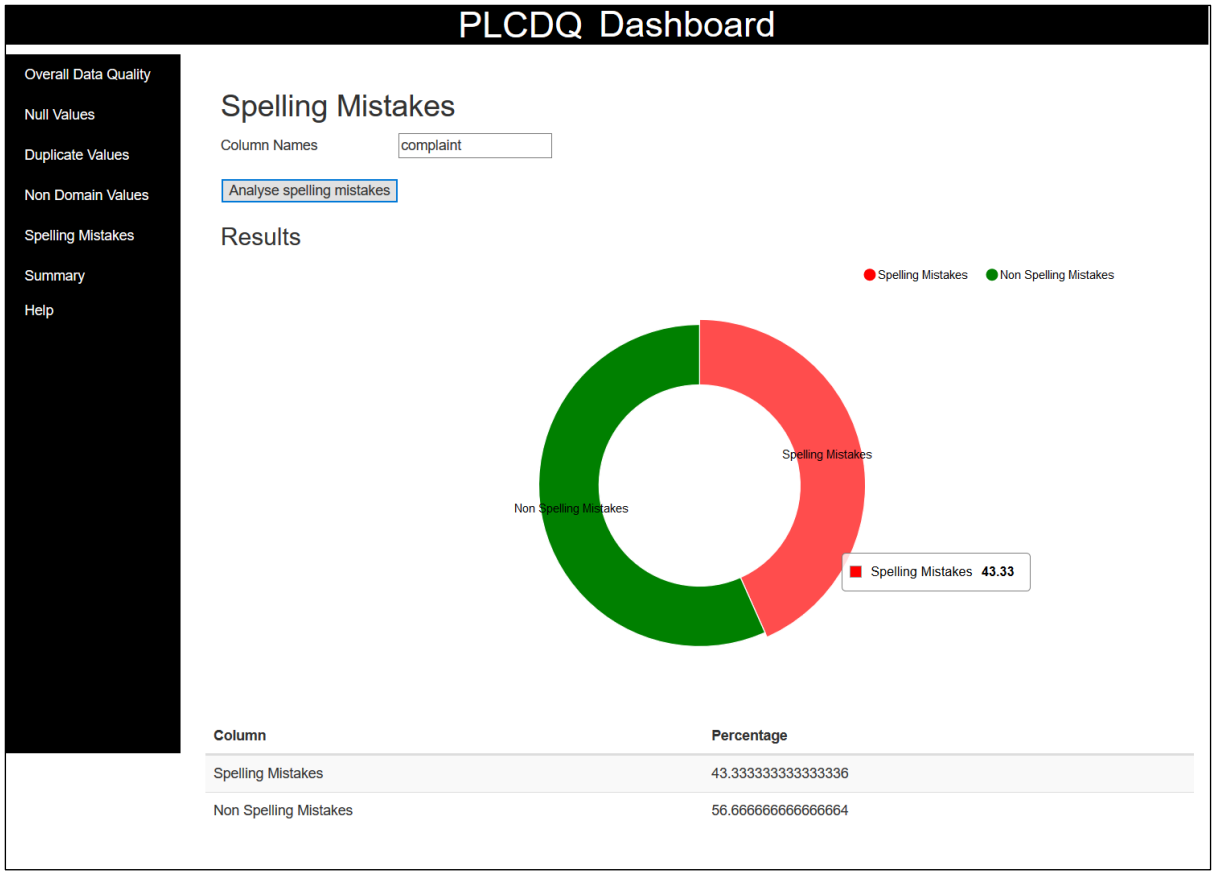
The Non-Domain Values page of the dashboard provided in the Screenshot calculates the non-domain value metric for a specific column. The user needs to choose the file that contains the domain values and click upload. Then, the user needs to enter a preferred column in the *Columns Names* text box and click the *Analyse column accuracy* button. This triggers the Non-Domain Values module described in the section 5.3.2 that computes the percentage of non-Domain values and visualizes the data quality in a donut chart.



Screenshot 15: User Interface for Non-Domain Values

Spelling Mistakes

The user interface to compute the metric spelling mistakes in the dashboard is shown in the Screenshot . The user needs to enter a preferred column in the *Column Names* text box and click the *Analyse Spelling Mistakes* button. This triggers the Spelling Mistakes module described in chapter 6.2.2 to compute the percentage of spelling mistakes in the column and visualize the data quality in a donut chart.



Screenshot 16:User Interface for Spelling Mistakes

REFERENCES

[1] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for Data Quality Assessment and Improvement," *ACM Comput. Surv.*, vol. 41, no. 3, 16:1-16:52, http://doi.acm.org/10.1145/1541880.1541883, 2009.

[2] Cornelia Kiefer, "Assessing the Quality of Unstructured Data: An Initial Overview," *In Proceedings of the LWDA 2016 Proceedings (LWDA), CEUR Workshop Proceedings*, no. 1613-0073, 62--73, http://ceur-ws.org/Vol-1670/#paper-25, 2016.

[3] Amir Parssian, Sumit Sarkar, Varghese S.Jacob, "Assessing Data Quality for Informaion Products: Impact of Selection, Projection, and Cartesian Product," *Institute for Operations Research and the Management Sciences (INFORMS),* http://dl.acm.org/citation.cfm?id=1014426, 2004.

[4] Jingyu Han,Kenjia Chen,Jiannig Wang, *Web Article Quality Ranking Based on Web Community Knowledge*: Springer, 2015.

[5] C. Fox, A. Levitin, and T. Redman, "The Notion of Data and Its Quality Dimensions," *Inf. Process. Manage.*, vol. 30, no. 1, pp. 9–19, http://dx.doi.org/10.1016/0306-4573(94 [Titel anhand dieser DOI in Citavi-Projekt übernehmen])90020-5, 1994.

[6] C. Batini, D. Barone, F. Cabitza, and S. Grega, "A Data Quality Methodology for Heterogeneous Data," *International Journal of Database Management Systems (IJDMS)*, vol. 3, no. 1, pp. 60–79, http://airccse.org/journal/ijdms/papers/3111ijdms05.pdf, 2011.

[7] Philip Russom, "Bi Search and Text Analytics: New Additions to the Bi Technology Stack," *Information & management,* http://www.bi-bestpractices.com/view-articles/5643, 2007.

[8] O. J. Karin Hartl, "Determining the Business Value of Business Intelligence with Data Mining Methods," *The fourth International Conference on Data Analytics,* https://www.thinkmind.org/index.php?view=article&articleid=data_analytics_2015_5_30_60153, 2015.

[9] Laura Kassner, "Product Life Cycle Analytics - Next Generation Data Analytics on Structured and Unstructured Data," *9th CIRP Conference on Intelligent Computation in Manufacturing Engineering,* http://www.sciencedirect.com/science/article/pii/S2212827115006514, 2015.

[10] Erin McKean, *New Oxford American Dictionary*: Oxford University Press, 2005.

[11] Dan Gillman,Frank Farance,Tae-Sul Seo,Ray Gates,Judith Newton,Keith Gordon,Guangzhi Sun, *ISO/IEC 11179-4 information technology–metadata registries.* [Online] Available: http://metadata-standards.org/11179/. Accessed on: Aug. 25 2012.

[12] Blumenthal S.C., *Management Information Systems*. NJ: Prentice-Hall: Englewood Cliffs, 1969.

[13] Fry, J.P & Sibley E.H, *Evolution of Data-base Management Systems*: ACM Computing Surveys, 1976.

[14] Davis C.H & Rush J.E, *Guide to Information Science*. CT: Greenwood Press: Westport, 1979.

[15] L. Sebastian-Coleman, *Measuring data quality for ongoing improvement: A data quality assessment framework*. Burlington: Elsevier Science, 2013.

[16] C. Batini and M. Scannapieco, *Data and Information Quality*. Cham: Springer International Publishing, 2016.

[17] Richard Y.Wang, "A Product Perspective on Total Data Quality Management," *Communications of the ACM,* http://dl.acm.org/citation.cfm?id=269022, 1998.

[18] Ralf Gitzel, Simone Turrin, Sylvia Maczey, "A Data Quality Dashboard for Reliability Data," *CBI '15 Proceedings of the 2015 IEEE 17th Conference on Business Informatics,* http://dl.acm.org/citation.cfm?id=2848831, 2015.

[19] SAS, *Data Visualization by SAS.* [Online] Available: http://www.sas.com/en_us/insights/big-data/data-visualization.html. Accessed on: Jan. 22 2017.

[20] The data visualization catalogue, *The data visualization catalogue.* [Online] Available: http://www.datavizcatalogue.com/search.html. Accessed on: Jan. 22 2017.

[21] Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju, *Web Services and their approach to Distributed Computing*: Springer Berlin Heidelberg, 2004.

[22] IBM, *RESTful Web Service by IBM*. [Online] Available: https://www.ibm.com/developerworks/library/wa-aj-multitier/. Accessed on: Nov. 02 2016.
[23] IBM, *Guided and automated analytics from the cloud by IBM*. [Online] Available: https://watson.analytics.ibmcloud.com/product. Accessed on: Sep. 14 2016.
[24] IBM, *Introduction to IBM Watson Analytics Data Loading and Data Quality*. [Online] Available: https://community.watsonanalytics.com/discussions/questions/548/introduction-to-data-loading-and-data-quality-docu.html. Accessed on: Sep. 30 2016.
[25] IBM, *How to Use Refine on Your Data by IBM*. [Online] Available: https://www.ibm.com/communities/analytics/watson-analytics-blog/learning-something-new-how-to-use-refine-on-your-data/. Accessed on: Sep. 14 2016.
[26] Uniserv, *Data Analyser Factsheet by Uniserv*. [Online] Available: www.uniserv.com/produkte/data-quality-service-hub/data-analyzer/. Accessed on: Sep. 30 2016.
[27] Uniserv, *Data Quality Scorecard by Uniserv*. [Online] Available: http://www.uniserv.com/services/data-quality-scorecard/. Accessed on: Sep. 30 2016.
[28] Informatica, *Data Quality by Informatica*. [Online] Available: https://www.informatica.com/de/products/data-quality/informatica-data-quality.html. Accessed on: Sep. 30 2016.
[29] Informatica, *Monitoring Data Quality using Metrics by Informatica*. [Online] Available: https://it.ojp.gov/documents/informatica_whitepaper_monitoring_dq_using_metrics.pdf. Accessed on: Sep. 30 2016.
[30] Salesforce, *Data Quality Analysis Dashboard Description by Salesforce*. [Online] Available: https://help.salesforce.com/apex/HTViewSolution?id=000214470&language=en_US. Accessed on: Sep. 30 2016.
[31] Salesforce, *Data Quality Analysis Dashboard by Salesforce*. [Online] Available: https://appexchange.salesforce.com/listingDetail?listingId=a0N300000016cshEAA. Accessed on: Sep. 30 2016.
[32] Talend, *Data Quality Dashboard by Talend*. [Online] Available: http://resources.idgenterprise.com/original/AST-0027904_Data_Quality_Dashboards_in_Support_of_Data_Governance.pdf. Accessed on: Sep. 30 2016.
[33] Attacama, *Data Quality Dashboard by Attacama*. [Online] Available: https://www.ataccama.com/files/sheets/dqd-140421-screen.pdf. Accessed on: Sep. 30 2016.
[34] InsightSquared, *Data Quality Dashboard by InsightSquared*. [Online] Available: http://www.insightsquared.com/features/data-quality/. Accessed on: Sep. 30 2016.
[35] Collibra, *Collibra*. [Online] Available: https://www.collibra.com/. Accessed on: Sep. 30 2016.
[36] Collibra, *Data Governance 4.5 Product Architecture by Collibra*. [Online] Available: https://compass.collibra.com/display/DOC/Product+Architecture. Accessed on: Sep. 29 2016.
[37] Collibra, *Metrics by Collibra*. [Online] Available: https://compass.collibra.com/display/DOC/Metric+Groups. Accessed on: Sep. 19 2016.
[38] Collibra, *Data Governance 4.5 Validation Rules by Collibra*. [Online] Available: https://compass.collibra.com/display/DOC/Validation+Rules. Accessed on: Sep. 29 2016.
[39] Collibra, *Data Governance Centre 4.5- Dashboard by Collibra*. [Online] Available: https://compass.collibra.com/display/DOC/Data+Quality+Dashboard. Accessed on: Sep. 30 2016.
[40] M. Jarke, M. A. Jeusfeld, C. Quix, and P. Vassiliadis, "Architecture and quality in data warehouses: An extended repository approach," *Information Systems*, vol. 24, no. 3, pp. 229–253, 1999.
[41] L. P. English, "Improving data warehouse and business information quality: methods for reducing costs and increasing profits," *John Wiley & Sons, Inc.*, 1999.
[42] Y. W. Lee, D. M. Strong, B. K. Kahn, and R. Y. Wang, "Aimq: a methodology for information quality assessment," *Information & management*, vol. 40, no. 2, pp. 133–146, 2002.

[43] H. Richards and N. White, *Ensuring the quality of health information: The canadian experience*: Springer Berlin Heidelberg, 2013.
[44] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data Quality Assessment," *Commun. ACM*, vol. 45, no. 4, pp. 211–218, http://doi.acm.org/10.1145/505248.506010, [Titel anhand dieser DOI in Citavi-Projekt übernehmen], 2002.
[45] M. J. Eppler and P. Muenzenmayer, "Measuring information quality in the web context: A survey of state-of-the-art instruments and an application methodology," *Proceedings of the Seventh International Conference on Information Quality (ICIQ-02)*, pp. 187–196, https://pdfs.semanticscholar.org/b7e0/4978992851255d26fd8a00b6673ea9f27f84.pdf, 2002.
[46] P. Falorsi, S. Pallara, A. Pavone, A. Alessandroni, E. Massella, and M. Scannapieco, "Improving the quality of toponymic data in the italian public administration," *Proceedings of the ICDT*, vol. 3, 2003.
[47] Y. Su and Z. Jin, "A methodology for information quality assessment in the designing and manufacturing process of mechanical products," *Proceedings of the Ninth International Conference on Information Quality (ICIQ-04)*, pp. 190–220, 2006.
[48] David Loshin, *Enterprise knowledge management: The data quality approach*: Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2001.
[49] M. Scannapieco, A. Virgillito, C. Marchetti, M. Mecella, and R. Baldoni, "The daquincis architecture: a platform for exchanging and improving data quality in cooperative information systems," *Information Systems*, vol. 29, no. 7, pp. 551–582, 2004.
[50] F. De Amicis and C. Batini, "A methodology for data quality assessment on financial data," *Studies in Communication Sciences*, vol. 4, no. 2, pp. 115–136, 2004.
[51] Carlo Batini and Monica Scannapieca, *Data Quality Concepts, Methodologies and Techniques*: Springer-Verlag New York, 2006.
[52] John Stark, *Product Lifecycle Management*: Springer International Publishing.
[53] Edwin M. Knorr, "Outliers and Data Mining: Finding Exceptions in Data," The University of British Columbia, 2002.
[54] C. Y. Eric Poulin, *Outlier Detection and Analysis.* [Online] Available: www.cse.yorku.ca/~jarek/courses/6412/lectures/Outliers.ppt. Accessed on: Jan. 25 2017.
[55] SciPy, *SciPy.* [Online] Available: https://www.scipy.org/.
[56] Daniel Bär,Iryna Gurevych,Ido Daga,Torsten Zesch, "A Composite Model for Computing Similarity Between Texts," Technischen Universität Darmstadt, 2013.
[57] C.D. Manning, P. Raghavan, H.Schütze, *Introduction to Information Retrieval*. New York: Cambridge University Press, 2008.
[58] AngularJS, *AngularJS Developer Guide.* [Online] Available: https://docs.angularjs.org/guide/introduction. Accessed on: Jan. 26 2017.
[59] D3, *Data Driven Documents.* [Online] Available: https://github.com/d3/d3/wiki. Accessed on: Jan. 26 2017.
[60] NVD3, *NVD3 Reusable Charts for D3.* [Online] Available: http://nvd3.org/index.html.
[61] Angular-nvD3, *Angular-nvD3 Directive.* [Online] Available: https://krispo.github.io/angular-nvd3/#/. Accessed on: Jan. 26 2017.
[62] Bootstrap, *Bootstrap.* [Online] Available: http://getbootstrap.com/. Accessed on: Jan. 26 2017.
[63] Apache Tomcat, *Apache Tomcat Document.* [Online] Available: http://tomcat.apache.org/. Accessed on: Jan. 26 2017.
[64] CherryPy, *CherryPy Documentation.* [Online] Available: http://docs.cherrypy.org/en/latest/. Accessed on: Jan. 26 2017.
[65] Jersey, *Jersey Documentation.* [Online] Available: https://jersey.java.net/. Accessed on: Jan. 27 2017.
[66] JSON-Simple, *Documentation for JSOn-Simple.* [Online] Available: https://code.google.com/archive/p/json-simple/. Accessed on: Jan. 27 2017.

[67] Dr. Felix Naumann, *DuDe Duplicate Detection Toolkit.* [Online] Available: https://hpi.de/naumann/projects/data-quality-and-cleansing/dude-duplicate-detection.html. Accessed on: Jan. 17 2017.
[68] PyEnchant, *PyEnchant Documentation.* [Online] Available: http://pythonhosted.org/pyenchant/. Accessed on: Jan. 29 2017.
[69] AngularJS, *Services in AngularJS.* [Online] Available: https://docs.angularjs.org/guide/services. Accessed on: Jan. 29 2017.
[70] NHTSA, *NHTSA Consumer Complaints Dataset.* [Online] Available: http://www-odi.nhtsa.dot.gov/downloads/. Accessed on: Jan. 12 2017.
[71] Kevin Gimbel,Nathan Schneider, Brendan O'Connor,Dipanjan Das,Daniel Mills,Jacob Eisenstein,Michael Heilman,Dani Yogatama,Jeffrey Flanigan,Noah A.Smith, "Part-of-speech tagging for Twitter: annotation, features and experiments," *In Proceedings of the 49th Annual for the Association for Computational Linguistics: Human Language Technologies,* http://dl.acm.org/citation.cfm?id=2002747, 2011.
[72] CoNLL, *CoNLL 2000 Dataset.* [Online] Available: http://www.cnts.ua.ac.be/conll2000/chunking/.
[73] F. N. Uwe Draisbach, "DuDe: The Duplicate Detection Toolkit," *Draisbach2010DudeTD,* http://www.vldb.org/archives/workshop/2010/proceedings/files/vldb_2010_workshop/QDB_2010/Paper5_Draisbach_Naumann.pdf, 2010.

Declaration

I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work, nor significant part of it was part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Signature

(Shalini Chellthurai Saroja)

(Stuttgart, 10-02-2017)