— Master's Thesis —

Assessment and Visualization of Metadata Quality for Open Government Data

Konrad Johannes Reiche

October 17, 2013

Reviewers
Prof. Dr. Ina Schieferdecker
Prof. Dr. Claudia Müller-Birn

Supervisor
Dr. Edzard Höfig

FREIE UNIVERSITÄT BERLIN

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
INSTITUTE OF COMPUTER SCIENCE

Declaration of Authorship

I hereby confirm that I have written this thesis on my own and that I have not used any other materials than the ones referred to. This thesis has not been submitted, either in part or whole, for a degree at this or any other university.

Konrad Johannes Reiche

October 17, 2013

Abstract

With the rise of the open data movement, government and public agencies start to open up their data for the public use. The technical tool for implementing this infrastructure are repositories. Repositories facilitate the collection, publishing and distribution of data in a centralized and possibly standardized way. Metadata is used to catalog and organize the provided data. The operationality and interoperability depends on the metadata quality.

Quantifying the metadata quality can help to measure the efficiency of a repository and discover low quality metadata records which prevent the user from finding what he/she is looking for. For this a range of metrics from the field of metadata quality assessment are researched and implemented. Current approaches should be adopted to the specifics of open government data repositories but also new approaches should be explored to fit the requirements.

In order to show the feasibility of these metrics a platform is implemented which demonstrates the automatic quality assessment of different repositories. A harvester component is used to gather metadata from different repositories. The metrics are discussed in detail, but also the platform's experimental results are analyzed for practical usage.

Acknowledgements

My sincere thanks goes to Dr. Edzard Höfig for his continuous advice, reviews and knowledge throughout the course of this work. His ideas always helped to spark the research into new directions. In the last two years I attended several of his lectures and I am glad for eventually writing my master's thesis under his guidance. I would also like to thank Prof. Ina Schieferdecker for making this thesis possible in the first place, her extensive knowledge in the research area of open data and quality assurance helped to set the keystones for this work.

Special thanks are due to Florian Marienfeld, who is the reason why I am at Fraunhofer FOKUS. For his expertise on the open data platform and metadata in general I am very grateful. It helped to identify issues in the field of metadata quality and pushed the research into the right direction.

I want to acknowledge Nikolay Tcholtchev, who reviewed parts of the thesis and gave insightful notes. I also want to thanks Thomas Tauber who printed the thesis in Vienna. Much appreciation goes to all my friends for their faith and patience.

My greatest gratitude goes to my girlfriend Mareike Schirmer. Without her constant love and support I would not have had the stable environment to make this work possible. Finally, I would like to thank my mother Britta Reiche and my grandparents Dorit Reiche and Dr. Heinz Reiche making my educational journey possible and supporting me throughout my study.

Konrad Reiche
Berlin, October 2013

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

"Information is power. But like all power, there are those who want to keep it for themselves. The world's entire scientific and cultural heritage, published over centuries in books and journals, is increasingly being digitized and locked up by a handful of private corporations."

—Aaron Swartz
Guerilla Open Access Manifesto (2008)

With rise of the open data movement, government and public agencies have started to open up their data for the public use including data which is collected and processed by the public sector. The Internet is a crucial tool for making government information available. Government information has become a keen interest of citizen, academics, and politicians. This is not limited to the published raw data itself, but the creation of applications, interfaces and visualization making the information dramatically more useful. On a technical level, metadata is used to index the data. Software repositories are used to make the metadata available. Too often a lack of quality in the content of metadata records leads to a loss in functionality of repositories. In order to harness repositories the research goal of this thesis is to find ways for assessing the metadata quality of open government data repositories.

Assessing the quality is only one step towards better and thus more usable metadata. For actual improvement the metadata providers have to be reached and informed about lack of quality. Visualization can be used as the language of data. By assessing the quality, visualizing the results and making the results available with a monitoring system a sustainable solution can be found to improve metadata quality over time.

Chapter 1 introduces the field of open government data and states the motivation and problem, and proposes a solution. In Chapter 2 the actual problem domain is discussed.

1.1 Open Data and Open Government

The topic of open data and open government is approached in different ways. One attempt of defining open data has been made by the Open Knowledge Foundation [1].

Definition 1 (Open Data)**.** Open data refers to a piece of content which is free to use, reuse and redistribute by anyone, only constrained by the needs to attribute the author, respectively the source and/or share it under a similar license.

An alternative approach is taken by the Sunlight Foundation. Instead of defining open data by one canonical definition, a set of open data policy guidelines is proposed. The guidelines answers the questions what data should be public, how to make data public and how to implement the policy. It includes a series of recommendations. For example, to set the default to open, mandate open formats for government data, remove restrictions on reuse of information or create processes to ensure data quality [2].

Like open source, open data is a vision and the idea to open all non-personal and non-commercial data in order to make it accessible for other uses. This is especially interesting with respect to data collected and processed by government organizations. For democratic reasons this would lead to more transparency, participation and innovation in society.

Initiatives that promote or reinforce the trend are described as open government projects. These initiatives provide reusable data as one of many efforts designed to increase overall governmental transparency. For example, President Obama's Open Government Directive, instructed executive branch agencies to publish information online in an open format. Open formats are platform independent, machine-readable, and made available to the public without restrictions that would hinder the reuse of these information. Among many promises of policymakers is the commitment to provide high-value information, including raw data, formats the public can easily locate, understand and use and formats that facilitate reuse [3].

These open government policies have blurred the distinction between the technological aspect of open data and the political aspect of open government: open government and open data can exist independent and without the other. A government can be an open government by being transparent without embracing new technology and a government can provide open data on politically neutral topics. If, for example, a government like those of the Hungarian cities of Budapest and Szeged provide online, machine-readable transit schedules then the data is both, open and governmental, but without providing accountability for the Hungarian government. Not only on these accounts is the popular term open government data is deeply ambiguous. When looking at the given definition of open data another factor becomes visible: licensing. Not all data that the government makes freely available is open government data.

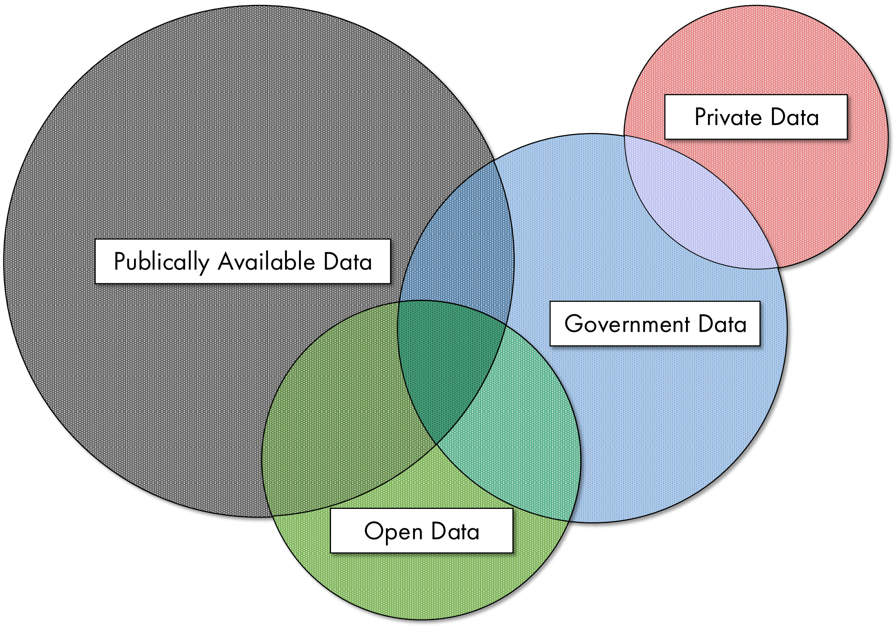


Figure 1.1: Venn diagram visualizing the different classes of data

1.1.1 Terminology

The terminology concept is illustrated in by a Venn diagram in Figure 1.1. There is *government data*, there is *publicly available government data* and there is *open government data*. Government data is the biggest subset containing data produced and processed by the government. Publicly available government data is data, which is made accessible. Public government data is free, but not necessarily open. Hence, open government data only refers to data which is made available free of charge in the interest of the general public for use, for spreading and for reuse without any limitations. This is opposed to private data because private is by definition not public. After all it would be unconscionable if the government would give out private data for everyone who asks for it. The analyzed data in this work will be referred to as government data, because open government repositories contain almost always data which is published under an open license, as well as a closed license. Actually public government data would be precise. Government data, however, is the shorter and more convenient term. Yu and Robinson use a framework to categorize open data even further by using the dimensions adaptable data, service delivery, inert data and public accountability. In this thesis, however, the problem will be limited to the choice of licenses.

1.1.2 Rise of Open Government Data

In April 2003 the Advisory Panel on Public Sector Information (APPSI) is established to advise on and encourage opportunities in the information industry for greater reuse of public sector information. In November 2003 the European Public Sector Information Directive (PSI Directive) is adopted. The PSI Directive is a directive by the European Union (EU) in order to encourage EU member states to make public sector information available for reuse. States had to implement it by July 2005. In June 2006 the MEPSIR (Measuring EU Public Sector Information Resources) Study estimated a mean potential value from the PSI Directive reuse across Europe at EUR 27 billion. The Apps for Democracy program is ran in October 2008, in order to encourage reuse of data from the DC data catalogue. In January 2009 President Barack Obama issued the Memo on Transparency and Open Government as one of his first acts in office. First Rewired State Hack the Government Day was held in March 2009 in the UK including 80 developers building applications of which many involved scraping government data [4].

In May 2009 data.gov (US) launched in the US with 47 datasets. In January 2010 data.gov.uk (United Kingdom) is officially launched. In February 2013 the GovData.de (Germany) prototype has been launched. In April 2013 data.gov offers 373.029 raw and geospatial datasets and at the time being, GovData.de offers 4,595 datasets and data.gov.uk offers 14,297 datasets.

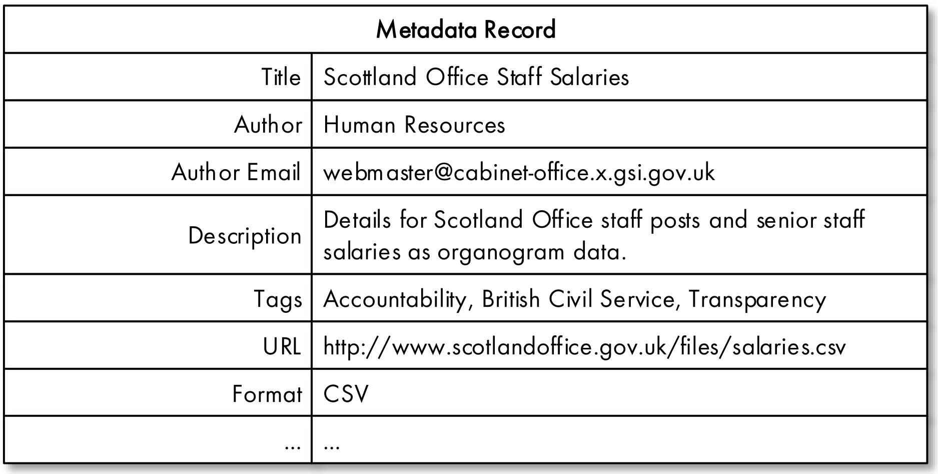
1.1.3 Definitions

Before describing the motivation, problem and solution approach some basic definitions will be given to set the stage. In Subsection 1.1.1 the ambiguity of the term open data was already discussed. The following definitions will help to understand what is meant when these terms or similar terms are used.

Definition 2 (Resources)**.** A resource is a component, a part of the published data. Data can be embodied by different formats. Each format, representing the data, or being a part of the data is a resource.

Example: A government organization decides to open up statistical data. The statistical data is represented by two datasets and a document. The two datasets comprise of a comma separated value file (CSV) and a Microsoft Excel spreadsheet (XLS). The document is a file in the PDF format. With respect to the information, the different formats can be redundant, but also complimentary.

Definition 3 (Metadata)**.** Metadata is a means of information gathering around the resources. The metadata describes the resources in different aspects. These aspects ought to to give an-



Metadata Record	
Title	Scottland Office Staff Salaries
Author	Human Resources
Author Email	webmaster@cabinet-office.x.gsi.gov.uk
Description	Details for Scotland Office staff posts and senior staff salaries as organogram data.
Tags	Accountability, British Civil Service, Transparency
URL	http://www.scotlandoffice.gov.uk/files/salaries.csv
Format	CSV
...	...

Figure 1.2: Example metadata record illustrating different fields and their values

swers about the who, what, when where and why of the resources. This includes who is the publisher of the data, what is the content, from when are the data, etc. The most inherent piece of information is a location description of these data. In practice, this would be a URL pointing to the actual resources.

Definition 4 (Metadata Record)**.** A metadata record is a software implementation of metadata. A record, also called tuples, structs or compound data, is a value that contains other values. Typically, the number of values and their sequence is fixed and they are indexed by names. The names are called fields, or members.

Definition 5 (Repository)**.** A repository is a software that is used for storing, cataloging and indexing data. This software is typically run in a server architecture on the web. Some repositories store the data, some repositories index the data through metadata records which is used to point to the actual resources and some repositories do both.

1.2 Motivation

Today, a government data repository is advertised by the number of metadata records made available. This figure is falsely used as a surrogate for the performance of a repository. The actual performance should be measured on the non-functional requirements, including: accessibility, discoverability, compliance, efficiency, effectiveness, interoperability and timeliness. It may be difficult to understand why quality can be an issue in the first place when examining a metadata record such as depicted in Figure 1.2. There are certain fields with certain values.

In reality, the problems are manifold. The content of a metadata record directly influences its quality and thus the non-functional requirements.

Values might be in a different format than specified by the schema leading to non-schema compliance. Non-schema compliance can be a problem for both, metadata consumer and metadata provider. Fields might be incomplete or missing altogethers. When a field has its value set the information provided can be insufficient. For example, when a field used for describing a resource lacks information due to being too short and indistinctive. The data can be outdated or false. Already simple problems like orthography can degrade the quality of a record. Yet the question remains, does this really influence the usability of a repository?

In fact, a large proportion of metadata having a low quality can render the whole repository useless. An essential component is the URL which links to the actual resources. Without maintenance links may go dead and a metadata record with a broken link cannot fulfil its task. Information of low quality degrade the discoverability dramatically. Metadata records being outdated lead to incredibility and untrustworthiness. In the worst case, a repository is not used anymore because of its reputation. Open data and government data has generated excitement in the communities of experts, donors, governments and developers. Everyone should be able and aware that he/she can take advantage of the data. Repositories need to address the quality question for supporting data literacy across the public.

1.3 Problem

Metadata quality is not standardized. For a number of important attributes there is no good ratio scale known. Quality belongs into this category. Quality is subjective and depends on individual requirements. Yet for a human it is possible to judge if a given metadata record is of use or not. Criteria need to be researched which make the quality definable. In order to approach this problem, three questions need to be answered:

1. What attributes influence the quality of metadata?
2. How can metadata quality attributes be quantified?
3. Does a quantification satisfy the metadata quality assessment?

The quantification needs to be formalized. The total number of metadata records suggests an automatic assessment. While metadata quality assurance can also be addressed by a proper process it is not a feasible method for an initial quality assessment. For a final assessment method it has to be considered that a series of quantification methods may not be a final solution. The inherent property of quality suggests that a series of quantification methods for quality criteria will not solve the problem completely. Hence, alternative ways of communicating the state of metadata quality need to be involved, too.

1.4 Approach

The proposed solution in this thesis is basing the quality assessment on metrics. Quality metrics are functions mapping metadata records to symbolic values. The symbolic value is ought to quantify a quality aspect of the metadata record. Thus, there is not one metric, but many metrics for a range of different properties. As stated above, with respect to quality symbolic values can be a means to an end, but not an end in itself. The quality metrics are implemented as component of a platform for assessing the quality of repositories.

A platform is chosen, because it enables the users to engage with the question of metadata quality themselves. The platform is ought to be a tool to investigate about the quality of metadata. Ultimately, quality metrics form the core and are used to assign ranks. Visualization plays a crucial role in communicating data and statistics. Visualization offers the possibility to approach specific results from different perspectives allowing the user to explore the outcome in different ways. Visualization is not the focus of this research, yet it will be approached practical in order to harness the metric results.

In the end, it should not be a tool to ultimately differentiate between high quality and low quality repositories, but to function as a beacon for quality issues. This way a process can be created to assert and improve the quality continuously.

Chapter 2

Background

"We don't have better algorithms. We just have more data."

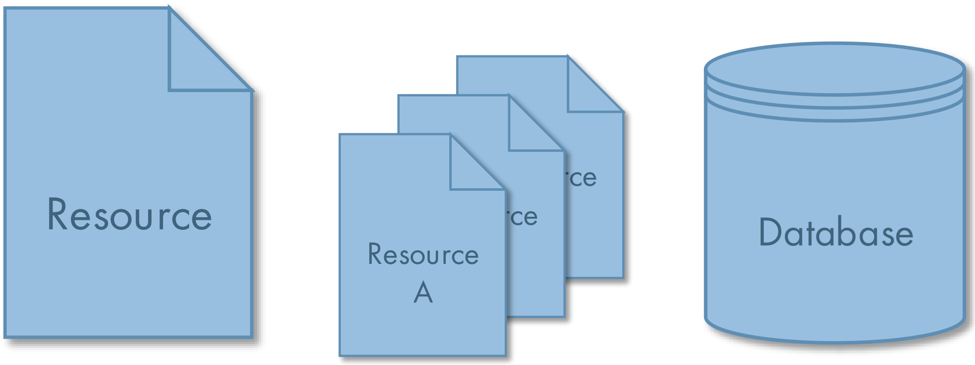
—Peter Norvig
Director of Research at Google (2006)

This chapter gives an overview over the basic knowledge required to understand the problem of metadata quality assessment. Firstly, the term metadata is introduced formally. Large parts are taken from the book Introduction to Metadata [5]. Secondly, repositories and their relationship to metadata is discussed. This is followed by approaching the domain of quality in general, and metadata quality in particular. Metrics are discussed separately in Chapter 3 since they are part of the solution domain.

2.1 Metadata

Metadata, literally meaning data about data has become the most used, yet underspecified, definition. The term is understood differently by various professional communities. Although they all design, create, describe, preserve, and use information systems and resources. Once, metadata was only a concern of information professionals engaging in cataloging, classification and indexing. An often cited example are libraries and their librarians using catalog cards to assess the content and location of a book.

Today, more and more resources are put online by the general public. Metadata is not any longer the solely domain of information professionals. While the term itself is much less familiar among providers and consumers of digital content, the same individuals grow used to the creation, exploitation and assessment of metadata in the age of user-generated web content. Even in schools and colleges students are taught to source their citations by searching for provenance and date information. This kind of resource will be referred to as information resource.



(a) Single item (b) Aggregate of many items (c) Record system

Figure 2.1: Different types of an information object

Definition 6 (Information Resource)**.** An information resource is an entity either in electronic or physical form or both which is capable of conveying or supporting intelligence or knowledge.

On a technical level an information resource is represented by an information object. Whereas the information object contains or embodies the information resource.

Definition 7 (Information Object)**.** An information object is a digital item or a digital group of items which can be addressed or manipulated as a discrete entity by a human being or an information system.

Given this definition, an information object can comprise of a single item, it can comprise as an aggregate of many items or it may be the entire database or any other record system. This is depicted in Figure 2.1. In addition, it narrows the scope of metadata usage. Due to the etymology the original meaning of the term will always apply in its broad sense. In the domain of information resources, however, a more sophisticated definition is suitable. Metadata consists of a set of information pieces about the information object. An information piece is a metadatum, respectively a statement. Thus, the following definition is proposed [5].

Definition 8 (Metadata)**.** The sum of statements that can be associated with any information object at any level of aggregation.

Nevertheless, it must be considered why information objects should be part of the discussion when talking about metadata. The generality of metadata is demonstrated by its definition. Information objects, however, make it possible to discuss metadata more concretely. The associated information object influences the metadata directly. This associative relationship induces the context. As a matter of fact, the context makes metadata more expressible. The context influences the content, but also the structure and the purpose (function) of the metadata. With this knowledge it should become clear to look in more detail at information objects, especially three distinct features: content, context, and structure [5].

- **Content.** The content is intrinsic to an information object which means that the content is inherent, respectively an essential part of the information object.
- **Context.** The context is extrinsic to an information object which means that the context is not an essential part of the information object. This includes the who, what, why, where and how aspects of an information object's creation.
- **Structure.** The structure can be either intrinsic, extrinsic or both to the information object. It is a formal set of associations. These associations can be within the information object or among different information objects, hence the intrinsic and extrinsic aspect.

From a historical point of view metadata has been used for centuries by librarians to index and categorize books. Until the mid-1990's the term metadata was primarily used by organizations involved with geospatial data, but also as data management and data maintenance. This was bound to a set of industry standards used by these organizations.

Today, metadata creation is not limited to humans, but involves processes including automated generation, too. Examples to automate the process of metadata creation include metadata transferring, metadata harvesting and web crawling. The connection of metadata with the web is inevitably. The W3C even defines metadata as machine understandable information for the web. HTML meta tags can be used to make semantic content machine-readable and thus even easier to find sites, index products offered in shopping sites, etc.

The consolidation of metadata across institutions, for example across online museum resource repositories, is a desirable goal but many approaches could only be met with limited success. Due to different notions of provenance, collectivity and structure as well as different institutional cultures the problem boils down to the fact, that there is no single metadata standard for describing all types of collections and materials.

It is important to note, that metadata is more than a mere set of descriptions that are used for the sole purpose of resource discovery. In digital information systems, like repositories, a broader range of activities include metadata. As a result a more inclusive conceptualization of metadata is required. Repositories create metadata for administration, accession, preservation and use of collections. All these different perspectives on metadata and the mission-specific focus from all the actors dealing with metadata rises the need for an in-depth discussion of metadata to make the concept more clear. This includes the functions, different structures and a categorization of metadata which will be elaborated in the following.

2.1.1 Functions

The functions of metadata can be classified into different groups. One function group includes the *creation*, *multiversioning*, *reuse*, and *recontextualisation* of information objects. When an information resource enters the digital information system this happens either by being directly created in a digital way or by being converted into a digital format. Different versions of the same resource might be created for reasons of preservation, research, exhibit, dissemination, but also product-development. This administrative and descriptive metadata is needed by the creator to fulfill this purpose.

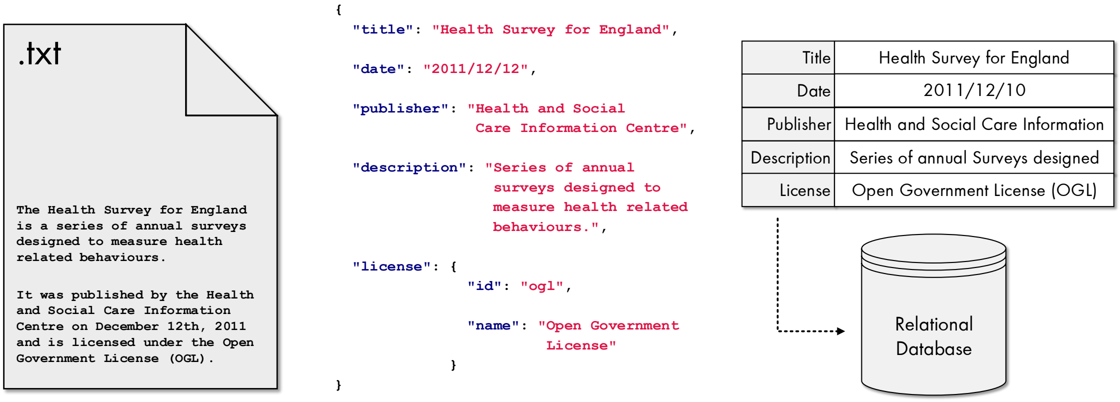
Another group is *organization* and *description*. This is a primary function of metadata which allows the ordering of information objects in a repository, but also further information object relating to the original information resource. This metadata can be given by the original creator, but also generated by the repository.

Validation is a very basic function of metadata. When users are searching for information resources they want to be assured that the given resources comply with their requirements. The metadata scrutinization is used to ascertain oneself on the one hand of the authoritativeness and on the other hand of the trustworthiness.

A major factor why descriptive metadata contributes to the quality of metadata is because it is essential to the searchability of information objects. Hence, *search* and *retrieval* are another function group of metadata. Metadata keeps track of location, way of retrieval, user transactions and the systems' effectiveness to find the object.

Utilization and *preservation* are part of the longterm functions of metadata. A characteristic of the digital realm is that data is typically not archived on hard drives to stay there for eternity. Information objects are subject to different kind of uses. In this process the objects, sometimes even the information resources, are reproduced, but also modified whereas modified also includes change of location. Metadata helps to persist user annotations, track rights and establish version control.

Disposition can only be achieved through metadata which is necessary to document the process. This is required to decide whether information object have become inactive or are not needed any longer in order to decide which should be discarded and which should be kept.



The Health Survey for England is a series of annual surveys designed to measure health related behaviours.

It was published by the Health and Social Care Information Centre on December 12th, 2011 and is licensed under the Open Government License (OGL).

```
{
  "title": "Health Survey for England",

  "date": "2011/12/12",

  "publisher": "Health and Social
                Care Information Centre",

  "description": "Series of annual
                  surveys designed to
                  measure health related
                  behaviours.",

  "license": {
               "id": "ogl",

               "name": "Open Government
                        License"
             }
}
```

Title	Health Survey for England
Date	2011/12/10
Publisher	Health and Social Care Information
Description	Series of annual Surveys designed
License	Open Government License (OGL)

(a) Unstructured

(b) Semi-structured

(c) Structured

Figure 2.2: Examples of different structure levels

2.1.2 Structure

As stated above the structure can be intrinsic, extrinsic or both to the information object. Whether metadata is structured, how much or not structured is decided by its use, context, but also technical circumstances. For instance factors decided by the implementors of repositories. Metadata is still data, too. Data can be either, unstructured, semi-structured or structured. The following elaboration of different metadata structure levels is depicted in Figure 2.2. The figure shows different examples for the structure levels ranging from unstructured, over semi-structured to fully structured.

There is unstructured metadata which can be of any type. This implies that it does not necessarily follow any format, sequence or rules. Unstructured metadata is not predictable. A typical example for unstructured metadata would be a plain text file that contains information about the resource. This is illustrated in Figure 2.2a. Further examples of unstructured data in general are media formats like videos, sounds or images.

On the opposite is structured metadata. Structured metadata is organized in semantic chunks (entities). Entities are grouped together to relations or classes if they are similar. If entities belong to the same group they share the same descriptions (attributes). If these attributes are part of of a group (schema) then the following statements are true [6]:

- Attributes have the same defined format.
- Attributes have a predefined length.
- Attributes are all present and follow the same order.

Structured metadata conforms to a somewhat predictable standardized or even proprietary format. A typical example for structured data are records of a relational database schema as shown in Figure 2.2c. Although structured metadata and unstructured metadata are both eligible, in practice a structure will be induced nonetheless. The prerequisite for metadata to provide the discussed functions is the accurate description of the information objects. This can also be achieved by using unstructured fields or other free-text annotations. In any case, essential attributes need to be captured. With unstructured data this is not possible explicitly, but implicitly. Due to lack of restrictions a metadata creator could simply establish a format. The problem of unpredictability, however, remains. There is no guarantee, no contract that the provided content complies to the model designed by the metadata creator.

As a matter of fact, most data has in one way or the other structure. For instance, even a normal text can be structured into sentences, paragraphs, sections, etc. Semi-structured metadata provides a trade-off between structured and unstructured data by reconciling between databases and documents. The idea of semi-structured data is to enforce well-formatted data. Typical examples include XML and JSON. This kind of data is partially available in database systems, file systems, but also data exchange formats.

Semi-structured Metadata

Semi-structured data is organized in semantic entities, whereas similar entities are grouped together. Entities in the same group may not have the same attributes. Also the order of the attributes is not necessarily important. From the set of attributes not all may be required (irregular structure). Parts of the data lack structure. The size and type of the same attribute in group may differ as well. Another distinct feature is the data model on which semi-structured data is based. The data model is a labelled graph, rather than a labelled tree (Figure 2.3). A graph does not automatically induce the hierarchy like a tree would do. The use is for data exchange among heterogeneous data sources. The schema information is stored in the edge labels. This kind of data model is sometimes called schemaless, respectively self-describing. Finally, the data itself is stored in the leaves [6].

The disadvantages of semi-structured data include the lose type information. Parts of the data may yield little structure. A database on the opposite, has a fixed schema. On the basis of the schema the database is populated. Every tuple conforms to a known schema. The downside is the data independence. Without the schema the context may be lost. Structured data lacks flexibility whereas semi-structured data is able to discover new data, load it as well as integrate heterogeneous data. Without knowing the data types the structure can be queried.

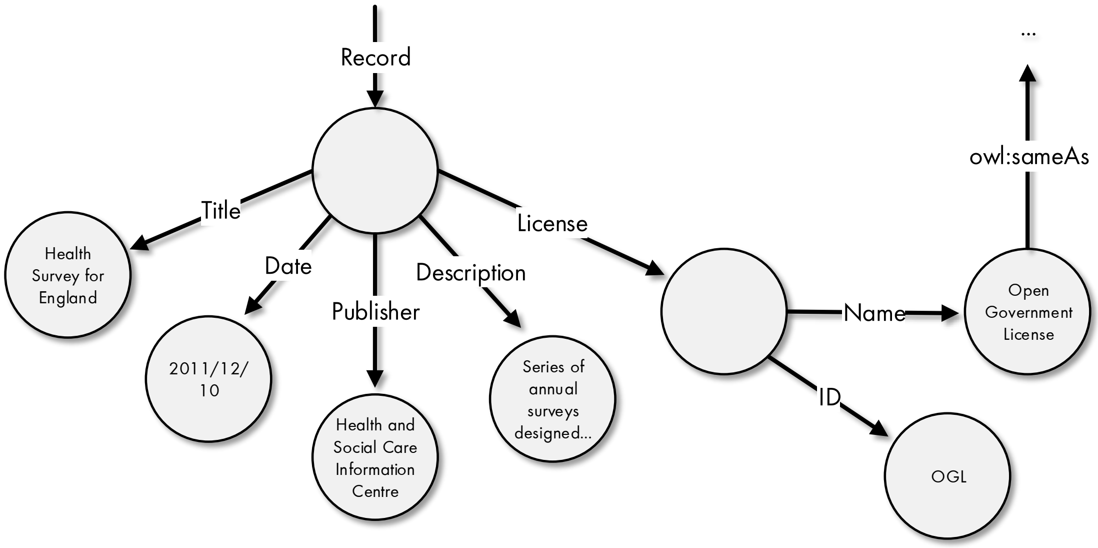

Record
Title
Health Survey for England
Date
2011/12/10
Publisher
Health and Social Care Information Centre
Description
Series of annual surveys designed...
License
Name
Open Government License
ID
OGL
owl:sameAs
...

Figure 2.3: The data model for semi-structured metadata is a labeled graph

Flexibility of Metadata Structures

All three types of data are used in the realm of metadata. While each type has advantages and disadvantages, semi-structured metadata strikes a balance between structured data and unstructured data. Although it offers the possibility of enforcing well-formated data it leaves the freedom of unstructured parts which can be designed as one likes. This discussion between flexibility and control is lead in the designs of metadata schemas, too. It might be tempting to design a metadata structure that defines every aspect of the data the metadata should describe. Ideally, this would lead to a well-defined metadata schema that includes every aspect of the data that needs to be represented.

This, however, ignores the inherent complexity that is generated by information. The actual data can become very complex. The context of the data is probably bigger than the data themselves. For every data that conforms to a certain metadata schema, there will be data that does not fit. The INSPIRE metadata schema illustrates this very well.

Geographic data is a field in which metadata is used extensively. The used metadata is often highly standardized. One standardization is ISO 19115. ISO 19115 defines a schema for describing geographic information and services providing information about the identification, extend, quality spatial and temporal schema, spatial references, and distribution of digital geographic data. Where ISO 19139 is the XML implementation of ISO 19115, INSPIRE is a XML profile based on ISO 19115 and ISO 19119. INSPIRE defines about 400 different fields and about

27 enumeration types. From a practical point of view this number is overwhelming. In fact, the vast majority of these fields are not mandatory.

INSPIRE is a good example for showing that a metadata schema to define all aspects of the context is often beyond being feasible. With so many fields it becomes hard to decide which field is appropriate for which piece of information. This is also true for the metadata consumer, where it is equally hard to decide in which fields to find which information. Yet each field in the INSPIRE standard probably has its right to exist due to reasonable rational.

2.1.3 Categorization

One reason for the confusing concept of metadata is their broad use for different domains and thus different use cases. It helps to categorize metadata into different domains. Namely, administrative, descriptive, preservation, use and technical metadata. These different domains, their definition and examples will be given in the following.

- **Administrative.** Metadata that is used for management and administration purposes. Examples include the acquisition of information, the tracking of rights and reproduction, documenting legal access requirements, use for locating information and as selection criteria for the process of digitization.
- **Descriptive.** Metadata that is used for identification and description, for instance cataloging record, finding aids, to differentiate between different versions, specializing indices, for curatorial information, linking between resources to persist the relationship or just annotations created by the author and users.
- **Preservation.** Besides the management of resources there is also preservation management. This includes to document the physical condition of these resources, to document actions taken in order to preserve the physical and digital version of resources (data refreshing and migration) as well as the documentation of any changes occurring during the digitization or preservation.
- **Technical.** Metadata which is in fact often seen very technical, can of course be applied to the technical domain as well: used to persist how a system functions. For example document hardware and software, information about the technical digitization (formats, compression ratios, scaling routines, etc.), tracking system response times and for purposes of authentication and security data (encryption keys, passwords, etc.)
- **Use.** Metadata that is related to the level and type of use including the circulation of records, to exhibit physical and digital records, in order to track use and users, also content reuse

and multiversioning of information, to establish search in logs and rights management of metadata.

It should be clear by now, that the context of metadata is an essential part of metadata itself. The context dominates, the model, the structure and the use of metadata. When dealing with aspects of metadata, and in this case that is the quality of metadata, the context has to be included and become part of the discussion. In the next section the software to store, manage and make metadata available is going to be discussed. Repositories have a crucial part in the assessment of metadata quality since to retrieve and analyse the metadata repositories are the interface for this task. After that quality in the context of metadata is introduced, defined and discussed.

2.2 Repositories

Repositories are the technical tool for implementing the metadata infrastructure. Repositories facilitate the collection, publishing and distribution of these data. Metadata is used to catalog and organize the data (resources). These metadata records describe the actual resources with additional information like authors, maintainers, formats, descriptive free text, etc. The referenced resources do not necessarily reside in the same repository. Centralized data is often hardly feasible and beyond administrative. The majority of data is heterogeneous and physically distributed which is also true for the people having the authority over these data. Metadata, in turn, is organized in a centralized and possibly standardized way using repositories. Semantic interoperability relies heavily on these metadata. The resources are published using web portals, but also web services and APIs.

2.2.1 CKAN

Among the repository software used for public government data the Comprehensive Knowledge Archive Network (CKAN) is one of the most popular. CKAN is a data catalogue for storing and distributing data which is developed and supported by the Open Knowledge Foundation (OKFN). The Open Knowledge Foundation is a non-profit organization founded in 2004 with the goal to promote open data. Even though CKAN is also capable of storing data in any format (unstructured, semi-structured or structured), the core component is the metadata management system.

In the beginning CKAN was promoted as the apt-get for the Debian of Data. Metadata records were coined as packages. Later the vocabulary changed the term dataset. Both terms, however, are still used. For instance, the CKAN code and API calls use the term package, but in newer code and through aliased API calls the term dataset has been established.

Metadata records are organized as datasets. Each dataset can reference and describe an arbitrary number of resources. The datasets are structured as associative arrays: each key maps to a certain value. Some values are associative arrays as well. As a matter of fact, CKAN can be comprehended as key-value store complying with a certain default schema.

The default schema contains fields like name, author, maintainer, license identifier, notes, tags, resources etc. Resources again, have their own default schema as well, containing fields like name, URL, description, format etc. The so called core metadata is fixed, for a dataset only the name field is required. In addition to the core metadata datasets comprise of relationships and unlimited additional metadata. Relationships can define dependencies between datasets, for instance depends on, child of, derived from, etc. The unlimited additional metadata is stored in a field called extras. Since the extras field is just another associative array it can contain an arbitrary hierarchy and depth. It means in effect, that data providers have the possibility to create their own metadata schema and constraints.

Metadata records can be added and modified through the CKAN powered web interface or REST API. All the functionality offered through the web interface can also be achieved with appropriate API calls. For instance: get JSON-formatted lists of a site's datasets, groups or other CKAN objects, get a full JSON representation of a dataset, resource or other object, create, update and delete packages, resources and other objects, etc. In the open data movement utilizing public APIs is understood as one of the most driving forces for new innovation as they enable programmers to explore new uses for the data [3].

More and more government organization choose CKAN for publishing their data. At the time being Austria, Brazil, Germany, Netherlands, Norway, United Kingdom and Uruguay use CKAN for their government data portals. The USA, still the pioneers in the field of open government and open data, have announced in January 2013 that they will publish their data using CKAN as well. In May 2013 they went public with that offering the most datasets so far.

2.2.2 Socrata

While CKAN is a general data repository Socrata is a more specialized to the domain of opening government data. Socrata is a Seattle-based software company that focuses on democratizing access to government data. As opposed to CKAN, Socrata is not a single software but a suite of different products. There is Open Data Portal, GovStat and API Foundry. Open Data Portal is presented as Socrata's open data portal to move data to the cloud in order to make it possible to review, compare, visualize and share the data. GovStat is a tool to measure the performance of government programs and make the results publicly available. API Foundry customized APIs can be created including the documentation. Socrata is widely used for many cities in the USA like New York City, San Francisco, Austin or Boston.

The Socrata Open Data API (SODA) is the programmatic interface to retrieve the data. Socrata models their resources through datasets, too. The schema of the API output is determined either by the structure of the data itself, or through the configuration applied to API Foundry. Dataset columns are represented as fields, keyed by a human readable field name. Socrata defines its own set of data types, including the usual suspects like strings, integers, double, boolean, etc. The datasets, when queried, can be serialized to different formats. Typically JSON is used, but XML, CSV and RDF are possible, too. Conveniently, the schema is included in the response of every HTTP request.

2.2.3 OGDI Data Lab

OGDI stands for Open Government Data Initiative. The OGDI Data is an open source cloud-based open data catalog developed by Microsoft. OGDI is used by organizations such as the Government of Columbia, Estonia & the European Union, City of Medicine Hat, Canada, City of Regina and most recently Niagara Region. OGDI is a data service like CKAN and Socrata and implements this service in a RESTful way. The number of supported formats include Atom Publishing Protocol (AtomPub), Keyhole Markup Language (KML), as well as JSON and JSONP.

OGDI Data Lab comprises of two other components: data loader and data browser. The data loader is a software utility which is used to import CSV and KML data into the catalog. This can be done either from a client machine or through command-line access to do so in a dynamically fashion from databased into Data Lab. The data browser then again provides a web interface to the data services. This way users can get a visual way to browse, query, interact with and download the data. Visualizations include tables, map, bar graphs and pie charts.

2.2.4 Harvesting

In the process of analyzing metadata, as it is required to assess the metadata quality, harvesting is a crucial component. While the data can be fetched and processed dynamically without storing them anywhere, this becomes infeasible when performing more complex computations. Most platforms offer so many data that it would take too long to fetch the datasets every time a computation should be executed. Harvesting the datasets beforehand gives the flexibility to create different snapshots of repositories locally and run the computations afterwards.

Harvesting is not only done by data consumers. As a matter of fact, data providers harvest metadata, too. For instance, the German government data platform (GovData.de) harvests data from its different states (Länder), but also single cities. Then again, GovData.de is harvested by the research prototype of the pan-European data catalog (PublicData.eu).

It is noteworthy, that there is a difference between scraping, crawling and harvesting, which should not be ignored in the context of their respective semantics.

- **Scraping.** Scraping, or data scraping, refers to the retrieval of information sources providing content. While scraping is typically performed on the web, scraping does not necessarily be limited to web scraping. Scraping can be performed on a local machine, on a database and on the Internet. In the process of extraction parts of the sources are selected to be transfered.
- **Crawling.** Crawling differs from scraping in scale and range. Crawler, or crawler agents, bots, spiders, etc. can be algorithmically designed to reach a certain depth. For instance a web crawler could be designed to recursively crawl and retrieve data from the web pages by following hyperlinks. Crawling, like scraping, is also a process of information extraction.
- **Harvesting.** Harvesting, refers to the semantic selection of information from a given source. This process is reflected in the required knowledge to perform the task. While crawling can be designed to just follow certain paths, like scraping harvesting needs to know what information is important and which should be ignored. This can be implemented algorithmically, too. What differs, is that metadata harvesting can include semantic mapping, too. This means transferring information from one format to another. This cannot always be done without losing or adding additional information. This knowledge requirement discriminates harvesting from scraping and crawling.

A standard has been developed for metadata harvesting: Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). For instance Google is using OAI-PMH to harvest information from the National Library of Australia. An implementation of OAI-PMH, however, is required to support representing metadata in Dublin Code. From there on it can also represent addition representations. Hence, OAI-PMH uses XML messages to retrieve data using the HTTP protocol.

2.3 Quality

Quality is the focus of this thesis, yet it is the hardest to grasp. Quality is highly subjective. Like metadata it depends on the context what quality means, how quality can be defined and what the implications are. Data quality will be discussed in general and metadata quality in particular.

2.3.1 Information Quality

There are institutional and individual processes which depend highly on information. The quality of information is a key element in the quality of decision making and action. There is a broad range of approaches to define information. The Oxford Reference Online states that information is whatever is capable of causing a human mind to change its opinion about the current state of the real world. In Science and engineering, information is whatever contributes to a reduction in the uncertainty of the state of a system. Here, uncertainty is usually expressed in an objectively measurable form.

Whereas data has a more neutral connotation, information always needs to be discussed in the context of the human mind. Quality means the standard of something as measured against other things of a similar kind, respectively the degree of excellence of something. In a generalized way this relates to a certain standard or level. Data quality has a more specific definition. The term was coined by Juran. Juran states that data are of high quality if the data is fit for their intended uses in operations, decision making and planning [7]. The question, however, is whether this definition is applicable to metadata quality. If so, it would mean that metadata are of high quality if they are fit for their intended use. What is the intended use of metadata? The intended use of metadata is the efficient cataloging and indexing of data so they can be found. While this definition is not wrong, it can be stated more concisely.

2.3.2 Metadata Quality

Today, the number of available datasets on an open government platform is instrumentalized for political reasons. The platforms advertise there effectiveness by displaying the total number of datasets available. While this is a great quantity factor, it is not a quality factor. Why is that?

Making the data accessible, does not imply that the users will find the resources they are looking for. Content publisher have to ensure that the resources are credible and discoverable. The credibility is bound to the quality of the content. The discoverability is bound to the quality of the metadata [8]. Hence, it is desirable to have high quality metadata. Ensuring a high quality is best done at the creation time and by an expert in the respective field. There are, however, several problems with this approach. Already, a lot of metadata exists, which needs to be evaluated and improved as well. The expert approach can never scale to the thousands of metadata. It would require to deposit all resources with the intention of publication into a queue which is then processed by this professional indexer. With the increasing number of organizations joining the open data trend this can only lead to a higher workload and it would become too hard to keep up with metadata record reviewing. Besides that, the problem would not be solved after one review. Content may change, provided information becomes invalid, outgoing links go dead, etc. and thus needs to be reevaluated. Furthermore, there are metadata which are auto-generated in the first place, for instance created due to the interoperability between repositories. For instance, data, especially geographic data are never current. For that reason it is naturally that metadata are effected by bad quality.

These difficulties drive the need for quality metrics which can be assigned automatically in order to help determining the metadata's fitness for a user's need. The metadata fitness could be defined by the effectiveness in supporting the functional requirements of the system it is designed for [9]. Evaluating the metadata quality of a repository can help to measure its efficiency, identify low quality metadata records and understand the reasons for the origin of the low quality. In the next step these metadata records could then be improved for achieving goals of higher-level criteria like a better searchability. A quality score would enable uniform comparison of qualities across repositories and allow to classify the metadata records in general.

Modeling the metadata quality can lead to the model. Above metadata quality we have information quality, which could be understood as goal. Metadata quality, data quality, information quality and intelligence quality. Each being the prerequisite of the next one. Quality helps to improve the decision making, this should be the use case from where the practical benefits derive. Based on this knowledge gain the more concrete definition of metadata quality can be given.

Definition 9 (Metadata Quality)**.** Metadata quality is the fitness of the metadata to describe the data, the resources, it is referring to. Whereas the metadata's fitness must support the task dimensions of finding, identifying, selecting and eventually obtaining the resources. The quality is inversely proportional to the metadata consumers', user's uncertainty about the described data, resources.

Before going into the solution domain for assessing metadata quality, namely metrics, an overview of known approaches will be given. This will help to understand what the state of research in this field is and how to proceed with this knowledge, but also give rational for the chosen solution.

2.4 State of the Art

Current research efforts come from different fields of study. In the work of Najjar, Stefaan and Duval an empirical analysis has been conducted to review the processing of metadata elements in a repository to understand the correlation between metadata elements filled-in by indexers [10]. Friesen, too, performed analysis on metadata records to find out how often metadata elements were put to use. The study showed that only 36% of the elements were used more than half of the time [11]. Park and Bui run similar analyses on metadata of a digital libraries to asses the quality based on the field usage [12]. All three works focused on finding the most frequently used fields and their associated values. In another study with respect to metadata creation, Greenberg showed that author-generated metadata can achieve acceptable metadata when created in an organization setting, where in some cases the results are better than produced by a metadata professional [13].

Taking visual aid into account Dushay and Hillman demonstrated how the evaluation of metadata records can be improved by using a visual graphical analysis tool. This was compared to random sampling with formatting and syntax highlighting and a spreadsheet approach. The graphical analysis tool allowed to view up to six data dimensions simultaneously [14].

Then there is the field of metadata quality assurance. Barton, Currier and Hey have discussed in their work the need for metadata quality as well as necessary principles and guidelines to implement this [15]. Guy, Powell and Day propose a quality assurance process for improving the metadata quality in an iterative way [9]. Currier, Barton, O'Beirne and Ryan survey the growing body of evidence that come from human-generated metadata creation [16]. Hillmann and Phipps propose the quality assurance by relying on application profiles[1] [17].

For a more generic approach there is a series of frameworks that can be used for the evaluation of metadata quality. This, more systematic approach, is for instance proposed by Moen, Steward and McClure. They describe the use of 23 different evaluation criteria like accuracy, completeness and serviceability to assess lack of quality [18]. Stvilia, Gasser Twidale and Smith created a even more generic framework which is based on the field of information quality. In this framework 32 information quality parameters are used which are then again classified into three dimensions: intrinsic, relational/contextual and reputational [19]. Hillmann and Phipps

[1] An application profile is a set of metadata elements, but also principles and guidelines that are defined in the context of a particular application. This element set can for instance originate from a larger super set of metadata elements.

discuss seven quality characteristics in more detail, namely: completeness, accuracy, provenance, conformance to expectations, logical consistency and coherence, timeliness, and accessibility [17]. Ochoa and also Duval in previous works have used this partly as theoretical basis to develop a set of metrics which can be implemented in order to automatically calculate values to assess the metadata quality [20]. Similar efforts can be found in the work of Hughes [21].

Yet another approach of metadata quality assessment can be found in the field of logic rules. Margaritopoulos, Margaritopoulos, Mavridis and Manitsaris propose a conceptual framework that uses structural and semantic relations among the metadata elements. Based on these relations logic rules which impose or prohibit values in the field of a metadata record are generated [22].

From the presented methods the quality metrics by Ochoa and Duval are picked up for the quality assessment of government data repositories. Quality metrics are modular. A metric is self-contained with respect to the attribute it is describing. In the original problem statement of Section 1.3 it was described that difficulties might arise when trying to quantify quality. The modularity of metrics would allow to add or remove specific metrics if required. This ability of individualizing the assessment is expected to be beneficial for the objective of this work.

Chapter 3

Quality Metrics

"Tell me how you measure me and I will tell you how I behave."

—Eliyahu M. Goldratt
The Haystack Syndrome (1990)

Above all, metrics make it possible to implement automatic quality evaluation. The manual evaluation by humans is considered to be the most meaningful. After all, it is a human being who has to use the repository. For a repository with a fixed size a manual evaluation can be feasible. Once asserted, the quality stays the same. Open government data portals, however, are only vital as long as new datasets are added and existing ones are updated. The environment changes, laws are passed or data bound to certain period of time become obsolete. The continuous reassessment is a crucial component. Allocating resources to cover the cost of manually evaluation once is feasible. The continuous manual evaluation becomes costly over time.

Automatic evaluation is seen as a trade-off between a method which is scalable and a method which is meaningful. Manual evaluation is meaningful, but not scalable. Simple statistical approaches are scalable, but not meaningful. It has been argued that automatic quality evaluation can be meaningful, too [20]. The criteria for this, is to use the same parameters that human reviewers would also use.

In this chapter the term metrics will be introduced in general and metadata quality metrics in particular. A comprehensive overview of applicable metrics with respect to metadata records will be given. These metrics can then be used to build tools for different kind of metadata repositories to provide a scalable, yet meaningful way to assert metadata quality.

3.1 Measurement

Metrics are a way of measuring something. That something is defined by the goal, but what is measurement? Measurement is a way of assigning a symbolic value to an object to enable the characterization of a certain attribute of that object. Measurement yields a useful abstraction. It allows to deal with complex subjects in a simple matter, yet keeping it precise and objective.

Definition 10 (Measurement)**.** The process P of using a measure M for assigning a value V to some object X in order to characterize attribute A of X.

A specific rule M for measuring A is $A_M(X) = V$. A measure is always on a particular scale. The scales can be equivalent, only differing in their measurement unit, or be non-compatible, differing in their approach to characterize A. The term metric itself is ambiguous. In mathematics a metric is a measure of distance between points. In software engineering it is rather a synonym for measure. In fact, for the latter case the term metric is more common than measure.

3.2 Scale Types

There are different levels of measurement, different scales of measures. The choice of methods and statistics is determined by the scale type, especially how to interpret the results. Five main scale types were described by Fenton and Pfleeger [23].

- **Nominal.** The nominal scale type, also categorical scale type identifies classes or categories. Each category groups entities based on value of their attributes. The data is qualitative so that the values are just names. Although numbers can be used as well, they would not have a numerical meaning. The inference for these values is the mode[1].
- **Ordinal.** The ordinal scale type, also rank scale type are ordered nominal data. While one value is larger than another, the size of the difference cannot be characterized. Practically this means that numerical operations like addition or subtraction cannot be applied. The inference is comparison and the median of the values.
- **Interval.** The interval scale type, also difference scale type supplements the ordinal scale type with information about the size that separates one category or class of another. The interval scale is a numerical scale so that numbers also have a numerical meaning permitting operations like addition or subtraction. The class zero, however, is not interpreted as the complete absence of the attribute being measured. As a matter of fact, the inference is done based on the difference and the mean.
- **Ratio.** The ratio scale type is an interval scale plus the existence of a zero element which represents the total absence of the attribute being measure. The ratio scale type includes most physical quantities.

[1] The mode is the value which appears the most frequent in a dataset.

- **Absolute.** The absolute scale type is a ratio scale where the classes or categories are restricted to a specific, fixed unit of measurement: counting. With a ratio scale type, attributes are measured in a certain unit. The class of this unit can be converted into another, which uses a different unit of measurement, while keeping the meaning of the obtained data unchanged. With the absolute scale, however, this is not possible.

3.3 Metric Engineering

Although metrics enable the precise and subjective description of objects, there are characteristics which determine the applicability of a metric. These characteristics of a measure are validity, reliability and precision. With validity the question is, how well does the metric really characterize the intended attribute? More specifically, how exact is it? The reliability of a metric is defined through the variance of its measurements: wow strong do multiple calculations on the same same object vary? Then there is precision, determining the maximal resolution of difference in the attribute.

Thus, for the effective use of metrics the domain needs to be studied in detail. This analysis is the foundation for finding appropriate metrics. It adheres to a certain research pattern. Firstly, data is obtained. This data represents the body of objects that are subject of the analysis. Secondly, this data is analyzed quantitatively giving evidence for the characteristics of the data. A model should be created of the process that produced the data. This model is then used to gain understanding of the process itself. Since the data are subject to the research the process is crucial. This understanding is harnessed by developing useful metrics summering the process characteristics. In the end is the use of these metric information to help and/or improve the same process or even related processes [20].

3.4 Quality Metrics

Quality is an unfavourable attribute for measurement. There are complex attribute which have no single measure. Quality belongs to that category. As stated before, quality is highly subjective. Different people will have different understandings of quality. This is also due to the fact, that quality is not one attribute but many. For example, in the case of metadata records there are attributes like accuracy, accessibility, conformance to expectations, completeness, comprehensibility or timeliness. For each attribute another measure is more appropriate. Thus, those measures are by no means equivalent, they rather measure different aspects of the attribute. Sometimes those metrics are representatives, so called proxies, of the actual attribute. Then there are attribute, which are hard to generalize at all. It becomes hard to construct an algorithmic metric which is sufficiently valid.

Ochoa and Duval have aggregated a rich set of metadata quality metrics. These metrics were developed for repositories managing metadata records of learning objects. They are sufficiently general for being applicable on metadata for government data, too. A selection of their metrics, including refinements and further metadata quality metrics will be represented in the following. Only quality metrics which have been implemented to test and evaluated their effectiveness will be discussed.

3.4.1 Formalism

Before a formalism for records and quality metrics will be given. This way the metrics can be defined in a more concise way. The metadata records are considered here as labeled records [24].

Definition 11 (Labeled Record)**.** A labeled record is the generalization of a n-tuple which consists of (label, value) pairs (l_i, v_i) where $i \in \mathbb{N}$. Each value v_i is annotated with a label l_i drawn from a predetermined set $\mathcal{L}$.

$$((l_1, v_1), (l_2, v_2), \dots, (l_n, v_n))$$

The special case of a 0-tuple is an empty record with no labels and no values. This definition suffices for semi-structured metadata. For structured metadata each label is associated with a type T_i making the underlying type and arity fixed.

$$((l_1, v_1), (l_2, v_2), \dots, (l_n, v_n)) : T_1 \times T_2 \times \dots \times T_n$$

All labels in a given record term or type have to be distinct. The extraction of a specific value v_j is done through the jth label projection π_{l_j}.

$$\pi_{l_j} : ((l_1, v_1), \dots, (l_n, v_n)) \rightarrow v_j$$

For the costs in terms of readability a labeled record $((l_1, v_1), \dots, (l_n, v_n))$ will be denoted as *record* and a projection $\pi_{l_j}(record)$ will be denoted as $record[l_j]$.

Definition 12 (Metadata Quality Metric)**.** A metadata quality metric for a metric m is a function $q_m : record_t \rightarrow V$, where $record_t$ is a metadata record of type t and V is a symbolic value. V is used to characterize the record through a specific quality attribute A.

The type of a record depends on the implementation details. For instance, a record provided through a CKAN repository has a different type than a record provided through a Socrata repository. The distinction is made because dependent on the record type the implementation might vary.

3.4.2 Completeness

The completeness metric deals with the number of completed fields in a metadata record. A metadata record is considered complete, if the record contains all the information required to have an ideal representation of the described resource. While the attribute of completeness again can be very vague, one way of constructing a metric for this is to simply count the total number of fields and all fields which have been set to a value which is not null. The completeness metric q_c is then defined as the ratio of number of fields and number of completed fields:

$$q_c(record) = \frac{\sum_{i=1}^{n} [\ record[field_i] \neq null\ ]}{n}$$

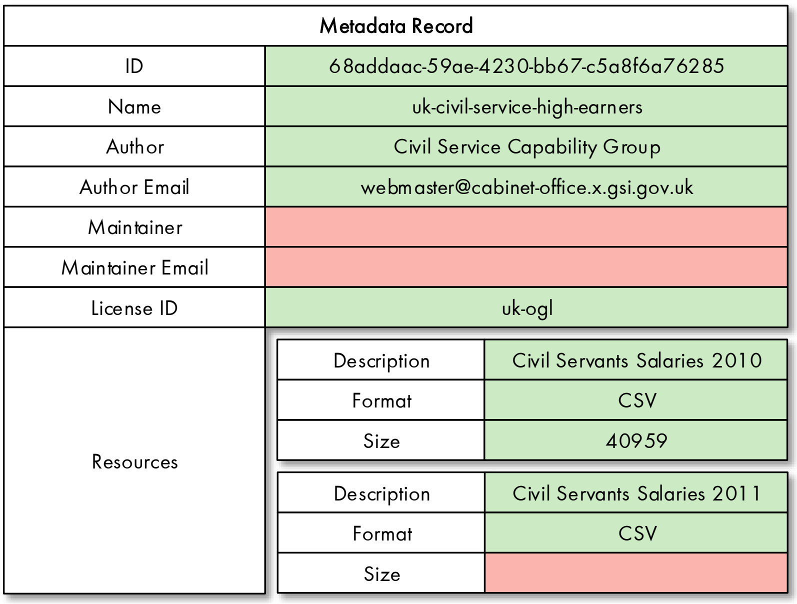
Dependent on the type t of $record \in R_t$ a set of different fields is associated with the record. The value for $field_i$ with $i \in \mathbb{N}$ is defined as $record[field_i]$. The outer term $[record[field_i] \neq null]$ uses the Inversion bracket $[P]$ which denotes a number that is 1 if the condition P is satisfied and 0 otherwise. The total number of fields is represented by n.

Example

Figure 3.1 depicts a metadata record with color highlighted cells. Green cells are completed fields and red cells are incomplete fields. Fields with an array value, for example the *Resources* field, are considered complete if they have at least one value that is not null. Associative arrays, for example values of the *Resources* field, are considered complete if they have at least one key with a value that is not null. Here the calculation is as follows:

$$\frac{1+1+1+1+0+0+1+1+(1+1+1)+(1+1+0)}{14} = \frac{11}{14} \approx 0.79$$

The brackets in the nominator denote the subfields of the *Resources* field. Thus, using this metric definition the metadata record is considered 79% complete.



Metadata Record		
ID		68addaac-59ae-4230-bb67-c5a8f6a76285
Name		uk-civil-service-high-earners
Author		Civil Service Capability Group
Author Email		webmaster@cabinet-office.x.gsi.gov.uk
Maintainer		
Maintainer Email		
License ID		uk-ogl
Resources	Description	Civil Servants Salaries 2010
	Format	CSV
	Size	40959
	Description	Civil Servants Salaries 2011
	Format	CSV
	Size	

Figure 3.1: Example metadata record for completeness metric

3.4.3 Weighted Completeness

While the completeness metric is straightforward it comes with the drawback of treating every field with the same importance. The relevance of a certain metadata field depends strongly on the context. Not all fields might be relevant for the user when deciding whether the metadata record describes the resources he/she is looking for.

To address this problem an alternative approach adds specified weights to each field. The value for the weighting could be assigned by an expert. The weight is a numerical value which expresses the relative importance for the fields to each other. This would allow to assign a weight of 1 for semi-important or regular fields, a weight of 3 for important fields, but also a weight of 0 for fields which should be excluded completely from the measurement as they do not add any additional information. The weighted completeness metric q_w is then defined as follows:

$$q_w(record) = \frac{\sum_{i=1}^{n} w_i \, [record[field_i] \neq null]}{\sum_{i=1}^{n} w_i}$$

Where w_i defines the assigned weight of the ith field. The question is, what source, respectively heuristic is used to allocate the weighting. While consultation of an expert is reasonable, a community survey might be of more use. Alternatively w_i could originate from the frequency

Field	Subfield	Weight
ID		0
Name		0
Author		1
Author Email		1
Maintainer		2
Maintainer Email		2
License ID		3
Resources	Description	3
	Size	1
	Format	1

Table 3.1: Weighting schema for the weighted completeness metric

users have relied on a certain field when searching for resources. These kind of heuristics, however, require further investigation of the repository usage.

Example

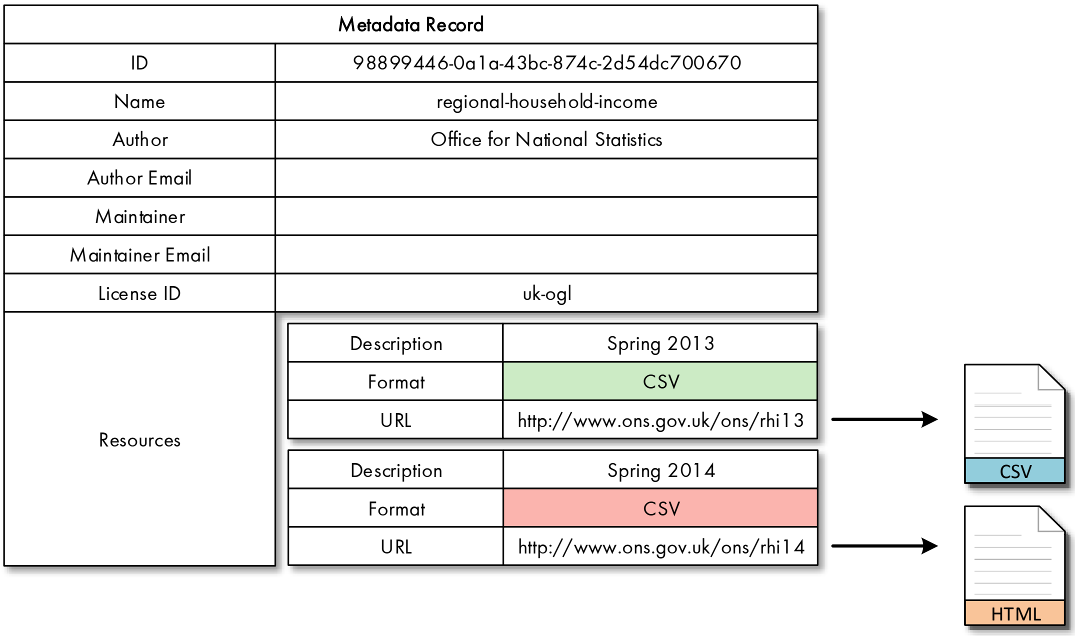
For the example the same metadata record of Figure 3.1 is used. Additionally, a weighting table is brought in (Table 3.1). Due to the weighting schema the fields *ID* and *Name* will be ignored. In this example these fields are mandatory and thus there are not as important as optional fields. Whereas the fields *License ID* and *Resource Description* are considered the most important. The calculation for this example is:

$$\frac{0+0+1+1+0+0+3+(3+1+1)+(3+1+0)}{1+1+2+2+3+2\cdot(3+1+1)} = \frac{14}{19} \approx 0.74$$

With this extended definition, the metric record is considered 74% complete.

3.4.4 Accuracy

The accuracy of a metadata record states whether the field values are correct with respect to the resources. In other words, how well does the metadata describe the actual resources? There are field types where this can be expressed with a boolean value. Either the given information is correct with respect to the resource or it is not. For instance, if the metadata record has a field for the file size of the resource. Either the actual file size of the retrieved resource corresponds to the definition or not.



Metadata Record		
ID	98899446-0a1a-43bc-874c-2d54dc700670	
Name	regional-household-income	
Author	Office for National Statistics	
Author Email		
Maintainer		
Maintainer Email		
License ID	uk-ogl	
Resources	Description	Spring 2013
	Format	🟩 CSV
	URL	http://www.ons.gov.uk/ons/rhi13
	Description	Spring 2014
	Format	🟥 CSV
	URL	http://www.ons.gov.uk/ons/rhi14

Figure 3.2: Example metadata record for the accuracy metric

Ochoa and Duval [25] propose that the correctness could be understood as the semantic distance between the information given through the metadata record and the information given through the resource. The semantic distance is the difference between the information a user can extract from the record and the information the same user could extract from the referenced resource itself. A shorter distance implies a higher accuracy of the metadata record. With this approach the metric q_a could be expressed with the following calculation:

$$q_a(record) = 1 - \frac{\sum_{i=1}^{n} d_i\,(record[field_i])}{n}$$

The difficulty resides in d, which is the distance measurement of the field value $record[field_i]$. Different fields require different, tailored distance measurements. For numbers and dates the offset can be computed, for categorical values a predefined distance table can be used, e.g. declared language and actual language. The language distance between Spanish and Italian is shorter than between Spanish and Japanese.

Example

This example is limited to d functions with a boolean value: the distance of a field *Resource Format* is measured through the file type of the actual resource. Either the specified format complies with the file type of the resource or not.

$$d(field_i) = \begin{cases} 0 & \text{if} field_i = resource_{value_i} \\ 1 & \text{otherwise} \end{cases}$$

The example record being subject to this metric is depicted in Figure 3.2. The URL of the first resource points to a CSV. This complies with the given format. The URL of the second resource, however, points to a HTML. Without further investigation it is unknown, whether the HTML contains another link to the actual resource. The calculation would be:

$$\frac{1+0}{2} = 0.5$$

With respect to the given formats, the metadata record has an accuracy of 50%.

3.4.5 Richness of Information

The vocabulary terms and the description used in a metadata record should be meaningful to the user. For that the metadata need to contain enough information for describing uniquely the referred resource. From the user perspective, the metadata record is of high quality if he/she is confident enough about what the referenced resources contain. Ochoa and Duval propose to do this by measuring the amount of unique information present in the metadata. They call this metric conformance to expectations. The approach originates from the field of information theory. In this work the metric will be called richness of information, as it describes the procedure better. The information content q_i is generally measured as follows:

$$q_i(record) = \frac{\sum_{i=1}^{n} I\left(resource[field_i]\right)}{n}$$

Where the function I returns a quantification of the information content, respectively the estimated amount of unique information. The difficulty here as well, is to define the function I for different types of fields. Ochoa and Duval propose for numerical values, vocabulary values

Record	Tags	Notes
A	Health, Children	Quarterly release of the Hospital Standardised Mortality Ratios (HSMR) of all hospitals participating in the Scottish Patient Safety Programme.
B	Finance, Spendings	A monthly-updated list of all financial spend transactions made by the Department for Business, Innovation and Skills, as part of the Government's commitment to transparency in expenditure.
C	Social, Health	Series of annual surveys designed to measure health and health related behaviours in adults and children.

Table 3.2: Example categories and text for the richness of information metric.

and free text two different functions. For numerical and vocabulary values they define it as 1 minus the entropy [26] which can be expressed with the following function:

$$I(field) = -\log P\left(resource[field]\right) \tag{3.1}$$

Where $P(value)$ is the probability for *value* to occur in a set of metadata records. For free text the term frequency-inverse document frequency (tf-idf) is proposed. A numerical statistic which reflects how important single words are relative to a collection of documents:

$$I(text) = \frac{\sum_{i=1}^{n} tf(word_i) \cdot \log\left(\frac{m}{df(word_i)}\right)}{n} \tag{3.2}$$

The function *tf* returns the term frequency: how often does $word_i$ occur in the current metadata record. The function *df* returns the total number of documents in which the $word_i$ is present at all. The number of documents is denoted by m and the number of different words by n.

Example

For the purpose of demonstration a hypothetical repository with three metadata records is used (Table 3.2). The records comprise out of two relevant fields: *Tags* and *Notes*. The record field *Tags* is a categorical field and *Notes* is a field with textual content. The richness of information metric is one of the metrics which need to do preprocessing before the single record scores can be computed. The preprocessing is done for categorical values and for textual values separately. For the categorical values the occurrence of each value is counted. There are 6 tags. Except

Word	Term Frequency (tf)	Document Frequency (df)	$\log\left(\frac{m}{df(word_i)}\right)$	tf-idf
quarterly	1	A	1.10	1.10
release	1	A	1.10	1.10
of	2	A, B, C	0.00	0.00
the	2	A, B	0.41	0.81
hospital	1	A	1.10	1.10
standardised	1	A	1.10	1.10
mortality	1	A	1.10	1.10
ratios	1	A	1.10	1.10
hsmr	1	A	1.10	1.10
all	1	A, B	0.41	0.41
hospitals	1	A	1.10	1.10
participating	1	A	1.10	1.10
in	1	A, B, C	0.00	0.00
scottish	1	A	1.10	1.10
patient	1	A	1.10	1.10
safety	1	A	1.10	1.10
programme	1	A	1.10	1.10

Table 3.3: tf-idf data table for Record A with 17 distinct words

Health, which occurs 2 times, every other tag occurs just once. Thus the probability of *Health* is:

$$P(Health) = \frac{2}{6} = \frac{1}{3}$$

For the other tags *Children*, *Finance*, *Spendings*, *Social* that is:

$$P(Children, Finance, Spendings, Social) = \frac{1}{6}$$

Eventually leading to the following information content:

$$I(Health) = -\log\frac{1}{3} \approx 1.099$$
$$I(Others) = -\log\frac{1}{6} \approx 1.792$$

It should become clear, that the tag *Health* contributes less information than the others as it occurs more often. The textual preprocessing is a bit more complex. Different components have to be counted. The number of documents m, here the number of records, the number of words n, the number of times a word occurs in a single record and the number of records in which a word occurs (df). The required informations are shown in Table 3.3, Table 3.4, Table 3.5.

Word	Term Frequency (tf)	Document Frequency	$\log\left(\frac{m}{df(word_i)}\right)$	tf-idf
a	1	B	1.10	1.10
monthly	1	B	1.10	1.10
updated	1	B	1.10	1.10
list	1	B	1.10	1.10
of	3	A, B, C	0.00	0.00
all	1	A, B	0.41	0.41
financial	1	B	1.10	1.10
spend	1	B	1.10	1.10
transactions	1	B	1.10	1.10
made	1	B	1.10	1.10
by	1	B	1.10	1.10
the	3	A, B	0.41	1.22
department	1	B	1.10	1.10
for	1	B	1.10	1.10
business	1	B	1.10	1.10
innovation	1	B	1.10	1.10
and	1	B, C	0.41	0.41
skills	1	B	1.10	1.10
as	1	B	1.10	1.10
part	1	B	1.10	1.10
commitment	1	B	1.10	1.10
government	1	B	1.10	1.10
to	1	B, C	0.41	0.41
transparency	1	B	1.10	1.10
in	1	A, B, C	0.00	0.00
expenditure	1	B	1.10	1.10

Table 3.4: tf-idf data table for Record B with 26 distinct words

Word	Term Frequency (tf)	Document Frequency	$\log\left(\frac{m}{df(word_i)}\right)$	tf-idf
series	1	C	1.10	1.10
of	1	A, B, C	0.00	0.00
annual	1	C	1.10	1.10
surveys	1	C	1.10	1.10
designed	1	C	1.10	1.10
to	1	B, C	0.41	0.41
measure	1	C	1.10	1.10
health	2	C	1.10	2.20
and	2	B, C	0.41	0.81
related	1	C	1.10	1.10
behaviours	1	C	1.10	1.10
in	1	A, B, C	0.00	0.00
adults	1	C	1.10	1.10
children	1	C	1.10	1.10

Table 3.5: tf-idf data table for Record C with 14 distinct words

Each table shows the preprocessed data of one record. The term frequency table shows how often the word occurred in this one text. The document frequency shows the number of different record the word occurred. Thus, the document frequency is actually the number of records in this column. The inner part of the sum in Equation 3.2 comprises of the term frequency (tf) and the inverse document frequency (idf). Finally, its product results in the term frequency-inverse document frequency (tf-idf). Due to the few cases in which $tf > 1$ the *idf* seldom differs from the *tf-idf*. The last steps involves the aggregation of these values. For the three records that is as follows:

$$I(A_{notes}) = \frac{13 \cdot \left(1 \cdot \log \frac{3}{1}\right) + 2 \cdot \left(2 \cdot \log \frac{3}{3}\right) + 1 \cdot \left(2 \cdot \log \frac{3}{2}\right) + 1 \cdot \left(1 \cdot \log \frac{3}{2}\right)}{17} \approx 0.912$$

$$I(B_{notes}) = \frac{20 \cdot \left(1 \cdot \log \frac{3}{1}\right) + 1 \cdot \left(3 \cdot \log \frac{3}{3}\right) + 1 \cdot \left(1 \cdot \log \frac{3}{3}\right) + 3 \cdot \left(1 \cdot \log \frac{3}{2}\right) + 1 \cdot \left(3 \cdot \log \frac{3}{2}\right)}{26} \approx 0.939$$

$$I(C_{notes}) = \frac{9 \cdot \left(1 \cdot \log \frac{3}{1}\right) + 2 \cdot \left(1 \cdot \log \frac{3}{3}\right) + 1 \cdot \left(1 \cdot \log \frac{3}{2}\right) + 1 \cdot \left(2 \cdot \frac{3}{1}\right) + 1 \cdot \left(2 \cdot \log \frac{3}{2}\right)}{14} \approx 0.95$$

The difference between all three note fields is quite marginal. Performing the tf-idf on a large corpus would yield more differential results. Nevertheless, the following statement is true:

$$I(A_{notes}) < I(B_{notes}) < I(C_{notes})$$

There are more unique words in the *Notes* field of Record A than in Record B and Record C taking the text corpus of all records into account.

3.4.6 Accessibility

Accessibility measures the degree to which a metadata record is accessible in terms of cognitive accessibility, but also physical, respectively logical accessibility. The cognitive accessibility describes how easy a user can comprehend what the resource is about after reading the metadata record. In the matter of searchability this could decide, whether the user finds what he/she is looking for or not. Due to the domain-specific vocabulary of government it might be difficult to understand the description with ease. Thus, the readability might be an indicator for the general cognitive accessibility. To implement this metric several readability indexes could be used. One of these is the Flesch-Kincaid Reading Ease which measures the comprehension difficulty when reading an academic text. The reading ease score for English texts can be computed by applying the following function q_r:

Reading Ease Score	Style Description	Estimated Reading Grade
0 - 30	Very Difficult	College graduate
30 - 40	Difficult	College student
50 - 60	Fairly Difficult	10th to 12th grade
60 - 70	Standard	8th and 9th grade
70 - 80	Fairly Easy	7th grade
80 - 90	Easy	6th grade
90 - 100	Very Easy	5th grade

Table 3.6: Score interpretation for the Flesch-Kincaid Reading Ease

$$\begin{aligned} Q_r(record) &= \textit{Flesch-Kincaid}(text) \\ &= 206.836 - 1.015\left(\frac{words}{sentences}\right) - 84.6\left(\frac{syllables}{words}\right) \end{aligned}$$

For this calculation the total number of words, sentences and syllables is required. For a broad interpretation of the results the definitions in Table 3.6 can be used. Although the described scores are on a scale between 0.0 and 100.0, negative values and values above 100.0 are possible, as well.

Example

For an example calculation the following text[2] is used:

> ”The 1906 San Francisco earthquake was the largest event (magnitude 8.3) to occur in the conterminous United States in the 20th Century. Recent estimates indicate that as many as 3,000 people lost their lives in the earthquake and ensuing fire. In terms of 1906 dollars, the total property damage amounted to about $24 million from the earthquake and $350 million from the fire. The fire destroyed 28,000 buildings in a 520-block area of San Francisco.”

There are four sentences, 75 words and 111 syllables. A hyphenated version of the text is below:

[2] National Oceanic and Atmospheric Administration, Department of Commerce. 1906 San Francisco, USA Images. `http://catalog.data.gov/dataset/1906-san-francisco-usa-images`

> ”The 1906 San Fran-cis-co earth-quake was the largest e-vent (mag-ni-tude 8.3) to oc-cur in the con-ter-mi-nous U-nit-ed States in the 20th Cen-tu-ry. Re-cent es-ti-mates in-di-cate that as many as 3,000 peo-ple lost their lives in the earth-quake and en-su-ing fire. In terms of 1906 dol-lars the to-tal prop-er-ty dam-age am-ount-ed to about $24 mil-lion from the earth-quake and $350 mil-lion from the fire. The fire de-stroyed 28,000 build-ings in a 520 block area of San Fran-cis-co.”

This leads to the following Flesch-Kincaid equation:

$$206.836 - 1.015\left(\frac{75}{4}\right) - 84.6\left(\frac{111}{75}\right) \approx 62$$

3.4.7 Availability

With the availability not the metadata record itself is meant, but its resources. Metadata records define URLs which point to the actual resources. The availability metric assesses the number of reachable resources. A resource is available, if the resource can be retrieved. This could also mean, if the accessed page actually returns the described format. That would, however, rather be task of the accuracy metric. Different concerns are kept separated between different metrics. The definition is as follows:

$$q_{av}(record) = \frac{\sum_{i=1}^{n}[a_i]}{n}$$

Where a_i is true if the ith resource of *record* is available through uri_i and n is the total number of resources in *record*. How the condition of *URI reachable* is checked is left for the implementation detail and will be discussed in Chapter 4. It should be noted, that this cannot always be decided clearly. For instance, what if the resource is not directly available, but through HTTP redirection (HTTP 301 and HTTP 302)? Eventually, the resource is reachable, but what about the original URL? Will this URL be active for the rest of the time or will the URL be shutdown at some point?

Example

A record with four resources where three of four resources are available through their respective URI would lead to the following calculation:

$$\frac{1+1+1+0}{4} = 0.75$$

The metadata record would have an availability of 75%.

3.4.8 Intrinsic Precision

The intrinsic precision is about the content of textual fields. Similar to the accessibility metric, this metric is about the reading fluency. The reading fluency is directly influenced by orthography of a text. Readers which are proficient in a language might halt for a moment on words written incorrectly. The number of spelling mistakes might not be a very important measure, as opposed to the availability of resources, nevertheless it influences the information quality. For the reading fluency metric q_{ip} the number of spelling mistakes are counted:

$$q_{ip}(record) = 1 - \frac{m}{n}$$

Where m is the number of spelling mistakes and n is the total number of words. A text with 50 words and 10 spelling mistakes would have a reading fluency of 80%.

Example

For the sake of the example a given text[3] has been used where misspellings, respectively typos were added afterwards. Spelling mistakes are colored in red.

> ”This data set contains roadway centerlines for city streets found on the USGS 1:24,000 mapping series. In som areas, these roadways are current through the 2000 construction season, elsewheere they depict features as represented on the USGS map.”

The text has 36 words and two spelling mistakes. The metadata record would have an intrinsic precision of $1 - \frac{2}{36} = 0.9\overline{4}$, respectively $94.\overline{4}\%$

3.4.9 Licenses

The license of a metadata record does not influence its quality. In the context of open government data the license is an important factor. Thus, it does influence the repository as a whole. The number of open data licenses is computed as a ratio, as well:

[3] Minnesota Department of Natural Resources. City Streets. `http://catalog.data.gov/dataset/city-streets`

Metric	Field Types	Scale Type	Description
Completeness	All	Ratio	Number of completed fields
Weighted Completeness	All	Ratio	Number of completed fields + weight
Richness of Information	Categories, Numbers, Text	Ratio	Measures the information content
Accuracy	Direct resource related	Ratio	Measures the semantic distance
Accessibility	Text	Interval	Measures the readability
Availability	URI	Ratio	Checks the availability of resources
Intrinsic Precision	Text	Ratio	Number of spelling mistakes
Licenses	License ID	Ratio	Number of open data licenses

Table 3.7: All metrics, their objective, scale type and description

$$q_l(record) = [license = open]$$

The score is binary. Either the record has an open license or the record has not an open license.

3.5 Summary

Seven different metrics to assess the metadata record quality have been introduced and explained on various example. In summary all metrics are shown in Table 3.7. The metric name, their objective, the scale type and a short description. The objective states on what kind of fields a metric focuses. Except the richness of information metric and the accessibility metric, all other metrics are on a ratio scale. For the quality assessment the scores should not only be evaluated isolated but in combination. The goal of this thesis is to find a method for measure the quality of a whole repository. This raises need to aggregate all the different values.

3.5.1 Normalization

When a metric computes the ratio between a part and the whole the score will lie within the $[0, 1]$ interval. For the intent of this work it is beneficial, because it can clearly be stated when the metadata record has its best quality and when its worst quality for a certain attribute. For two discussed metrics this is not the case: richness of information and accessibility. For the categorical richness of information the values can be within the interval $[0, \infty]$. The same is true for the textual richness of information. Although the accessibility metric has categories for values between 0 and 100, the Flesch index can also calculate a score below 0 and above 100.

When metric scores are on different measurement scales, normalization, respectively standardization becomes necessary. Traditionally, normalization means to rescale the values from their natural range to the range $[0, 1]$. Standardization typically means to rescale the value range to measure how many standardization deviation value is off the average. Standardization is often preferred as it produces meaningful information about each value and what is more, it provides an indicator for outliers.

Normalization is always an applicable method scaling the values to the $[0, 1]$ interval by using the minimum x_{min} and maximum value x_{max} of the data:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

The goal should be to rescale the metric scores not afterwards, but within the metric computation. Outliers do not have to be considered afterwards, instead the rescaling is done in a way that is natural to the metric computation. This way a potential post-processing is avoided. For the categorical richness of information Equation 3.1 the following alternative is used [20].

$$I(field) = 1 - \frac{\log(m)}{\log(n)}$$

The number of total occurrences for the field's value is denoted by m and the number of total possible instances for this specific field is denoted by n. For the textual fields, and thus tf-idf, it is more complex. In the work of Ochoa and Duval the logarithm is used. This way the document length is preserved. For example, two documents having similar content can have a significant difference in the score, just because one of the document is much longer.

This, however, must not be an objective for the context of open government data portals. The description length should rather be concise. Instead a standard approach is the cosine similarity [27]. After all, tf-idf is often used in the field of Machine Learning using the vector space model. The term frequencies of one document are represented by a vector $\vec{v}$. The term frequencies are normalized by computing the vector's L2-norm and dividing them:

$$\frac{tf(word_j)}{\sqrt{tf(word_1)^2 + \ldots tf(word_n)^2}}$$

Here the term frequency is normalized for the term $word_j$. Normalizing the accessibility metric is easier. There are reasonable definitions for values between 0 and 100. Thus, truncation can be applied. Values above 100 are reset to 100 and values below 0 are reset to 0. Then the score only needs to be divided by 100 to scale the values to $[0, 1]$. Possibly, another value can be chosen, too. For instance 80. It depends on the kind of users to target.

3.5.2 Metric Score Aggregation

Different metrics show different aspects in the lack of quality in a metadata record and thus the whole repository. The score assignment of a single metadata record is consequently followed by the score assignment for the whole quality metric of a repository. All the record scores need to be aggregated. After all the advantage of a unified score assignment method is to enable comparison of different repositories with each other. The comparative evaluation of open government data portals is based on the premise that sensibly comparison of repositories is possible based on their aggregate performance over a selected set of aspects. The assumption is, that the performance of a repository can be represented by a single overall quality score. This score could be the average or another central statistic.

A score providing empirical evidence that the hypothesised level of quality is attained would be desirable. This would also give better proof of overall quality improvement for a repository when this score increases. Hence, the goal an aggregate statistic that summarizes the intermediate results.

The simplest of these methods is the arithmetic mean. For a set of n metric scores $\{s_i \mid i \in 1 \dots n\}$ the arithmetic mean AM is computed as:

$$AM(s_1, \dots, s_n) = \frac{1}{n} \sum_{i=1}^{n} s_i$$

For the arithmetic mean all values have to be on the same scale. For the discussed quality metrics that would practically mean, that the richness of information and accessibility metric need to be converted to a $[0, 1]$ ratio scale. Another key aggregation mechanism is the geometric mean. The geometric mean GM is defined as the nth root of the score product.

$$GM(s_1, \dots, s_n) = \left(\prod_{i=1}^{n} s_i \right)^{\frac{1}{n}}$$

The geometric mean is less affected by outlying values than the arithmetic mean. If one or more scores are zero, the aggregated score becomes zero, too. Since this can be the case, it is rather problematic. Robertson has proposed a pragmatic solution [28] using a variation of the geometric mean which is computed using the logarithm. The adjustment to circumvent the problem, is to add a small quantity to the estimate before taking the logarithm and removing it again afterwards: the ε-adjusted geometric mean AGM.

$$\underset{\varepsilon}{AGM}(s_1, \ldots, s_n) = \exp\left(\frac{1}{n}\sum_{i=1}^{n}\log(s_i + \varepsilon)\right) - \varepsilon$$

Where ε is a small number. This burdens the specification of an ε and allows comparison only of values where the same ε has been used. It has to be tested, how sensitive an aggregated score is to the chosen ε. Another problem, is that even with the adjusted geometric mean, negative values are not possible, too. The accessibility metric can yield a negative score though. Another central tendency to combine scores is the reciprocal of the average of the reciprocals: the harmonic mean.

$$HM(s_1, \ldots, s_n) = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{s_i}\right)^{-1}$$

The harmonic mean is undefined if any score of the set of values is zero. Here too, a ε-adjusted version of the harmonic mean is convenient:

$$\underset{\varepsilon}{AHM}(s_1, \ldots, s_n) = \left(\sum_{i=1}^{n}\frac{1}{s_i + \varepsilon}\right)^{-1} - \varepsilon$$

Finally, the fifth tendency is the median MD. Informally, the median is the halfway point of a set. There are two cases for n scores $s_1, \ldots, s_n$. When n is odd, the median is unique: the element at position $\frac{n+1}{2}$. When n is even, there are two median candidates. The lower median occurring at $\left\lfloor \frac{n+1}{2} \right\rfloor$ and the upper median occurring at $\left\lceil \frac{n+1}{2} \right\rceil$. On the one hand, the median is relatively unaffected by outliers, on the other hand it is completely insensitive to value changes that do not affect the set ordering expect for the case in which the middle values change [29].

It would have been an advantage if one of these tendencies could be used for scores which are not the same scale. It would fail, because with excluding normalization completely scores from the accessibility metric can be negative, too. Due to the negligible number of metrics the mean is not an option. Instead the normalization is applied as discussed and the score is aggregated using the arithmetic mean. Metric score outliers are not a statistical problem in the context of quality assessment. How should score outliers be handled when computing the aggregate score for a repository? An answer to this question is discussed in the upcoming Chapter 4.

Chapter 4

Implementation: Metadata Census

In the previous chapter, it was shown how metrics can be concisely expressed by using mathematical notations. This theoretic view does not include all parameters required to solve this problem for real data. The practical view introduces a series of constraints. In Chapter 2 it has been stated that quality is highly subjective. An implementation limited to the metric computation would not exactly meet this standard. Instead the quality metrics are implemented as part of a web application which is called Metadata Census.

Metadata Census is a tool for the automatic metadata quality assessment. Moreover, it provides a portal to make the information available and comprehensible. This chapter focuses on the requirements, architecture, design and problems that have been faced in the development process.

4.1 Requirements

The central idea is to build upon the provided metric abstraction. The quality metrics encapsulate certain aspects of metadata records into a single number. This abstraction can be useful, but it is far more useful to become specific if one wants to understand. In the case of the implementation it is necessary when one wants to understand the rational behind the quality assessment.

A desired outcome is an application indexing a range of open government data portals and continuously monitoring their metadata quality while providing extensive information about the reasons and implications of a low or high quality. The user needs to be able to investigate the results. Not only as a method to scrutinize the metrics in their validity, but also to find out which metadata records need an improvement. Hence, after using the application the user should know the following. What is the metadata quality of a specific repository? What is the rational for a given quality score? What has to be done in order to improve the quality?

Before going into detail about the solution domain, the functional and non-functional requirements are addressed. The functional requirements for the implementation take the problem statement of Section 1.3 into account.

- **Metadata Harvester.** If the metadata is stored locally it is possible to access it afterwards, run additional analyzes, etc. It is faster and the metadata is available even if the repository is not.
- **Schemaless Data Store.** Metadata is persisted through different formats. A schemaless database can organize and manage the metadata in a natural way.
- **Quality Metrics.** The metrics form the core functionality of the implementation. It should be possible to apply the metrics on different repositories.
- **Scheduler.** A continuous quality monitor includes the need for scheduled computation tasks. These tasks need to manage the repository analysis.
- **Metric Reports.** How is a single metric score computed? The results need to be broken down into small information pieces that make the outcome understandable. A metric report also needs to contain general statistics about a repository that possibly affect the score.
- **Visualization.** The problem of making the assessment comprehensible is not necessarily solved by a large number of scores. Visualization can help to reduce the information noise for natural interpretation.
- **Leaderboard.** Open data is inherently political. In fact, open data has a competitive appeal. A leaderboard could be instrumentalized to compare the metric scores of different repositories with each other and encourage this competition.

Non-functional requirements describe characteristics of the implementation's functional behavior. The realization of non-functional requirements influences how well the application performs. There are two non-functional requirements which need to be addressed:

- **Scalability.** The number of metadata records inhabited by a repository can vary heavily. The implementation in general and the metric algorithms specifically, should be able to deal with different magnitudes of input data.
- **Extensibility.** A limited set of quality metrics is not the answer to every aspect of metadata quality. It is one approach and effectively a start. Thus, the platform should be open for extensions. For example, it should be easy to add new quality metrics.

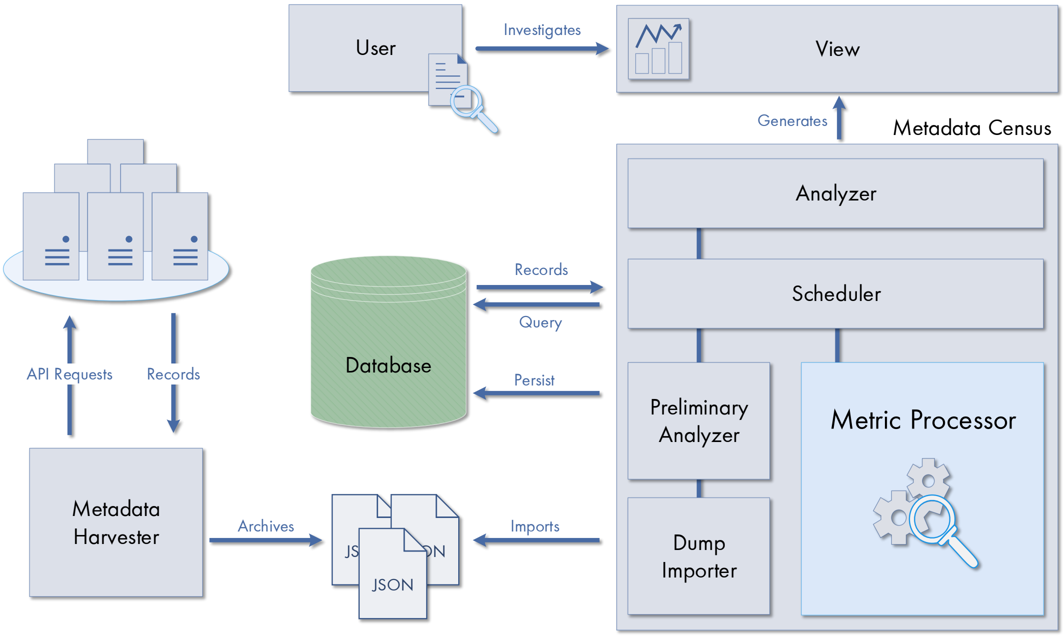


Figure 4.1: Architecture of the implementation

4.2 Architecture

While the application could be designed as one monolithic component it would introduce a potential drawback. If metadata records are requested, retrieved and directly piped into the database then this becomes inflexible with respect to future changes. For instance, the underlying database schema changes leading to complex migration queries. Thus, the harvester component and the analysis platform are kept separated.

The architecture is illustrated in Figure 4.1. The metadata repositories form the head in the process of metadata quality assessment. Based on a time schedule the metadata repositories are harvested continuously. The metadata records are stored on the file system. An importer component is responsible for retrieving the static record dumps from the file system. Before the records are inserted into the database, a preliminary analysis is run on them. This way statistical information such as number of resources, use of different languages, etc. can be collected and inserted, too.

The scheduler component is responsible for instantiating tasks that execute the metric calculation. The metric processor incorporates the application's main functionality delivering the quality metric scores for each metadata record. The rational behind a scheduler and a separate metric processing unit is performance. The metric calculations are rather CPU-intensive.

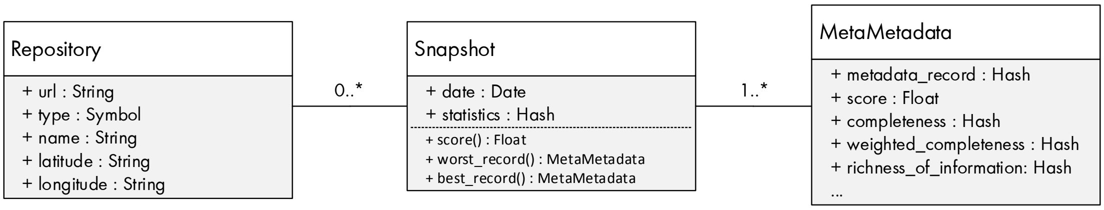


Figure 4.2: Domain model of the implementation

By decoupling their retrieval and computation from the rest of the application, the portal can function on its own.

Finally, there is an analyzer component. Based on the information collected during the preliminary analysis and during the metric computation the collected knowledge is rehashed for presentation. Based on this data a view is generated for the user. Eventually, the user should be able to investigate the state of a repository through this view.

4.3 Design

The design is a refinement of the requirements and the architecture. It gives a more concrete description how the functionality should be implemented.

4.3.1 Harvesting

The harvester and the importer have to be able to handle a large number of metadata records. The records cannot be retrieved and written to the file system in one iteration. While this is feasible for small repositories with a total number of 100 to 1,000 datasets, it quickly becomes infeasible for large repositories with a total number of 100,000 records and more. For instance, a JSON dump of 200,000 metadata records can have a size of approximate 3 GB. In memory, through associative array data structures it would consume even more space. Therefore a pipelined process is used. Only a fixed number of records are retrieved through one request, for example 1,000, and written directly to the file system by using a streamed file writer. In addition, the stream uses compression.

4.3.2 Domain Model

Normally, the domain model would comprise two entities: the repositories and the metadata records. Due to the continuous harvesting and monitoring a third entity has to be modelled:

snapshots. A snapshot is the logical unit for enfolding metadata records from certain repositories harvested at a specific point in time. Snapshots depend on the time and their associated repository. The time granularity for snapshots is set to days. This implies for a metadata record that it is unique based on the snapshot identifier (date) and its internal identifier. The snapshot are unique based on their date and the repository they are associated with. The relationship is depicted in Figure 4.2.

The metadata entity is actually a meta-metadata record. By wrapping the metadata records into another entity additional data about the metadata record can be stored. There is the `score` attribute which is the aggregate score of different metric scores and there are metric attributes. The metric attributes are stored as Hash (associative array), which contains the single metric score as well as additional information about the record.

4.3.3 Metric Algorithms

Desirably, the metrics would be applicable on different types of metadata. As a matter of fact, the algorithms implementing the metric functions do not only need the record as input, but also a schema. This is inherent to the problem, because the schema defines where what kind of information is stored. For instance, which fields contain textual content or which fields contain URLs. These kind of fields could be detected based on a fuzzy logic, but it becomes questionable whether the results will be sufficiently reliable. In order to keep the complexity down, the metric algorithms are based on the schema of respective repository to analyze.

4.3.4 Metric Classes

Classes are used to encapsulate the metric algorithms. For a snapshot calculation, a metric object is instantiated. If necessary, the object is instantiated with the metadata records in order to perform analyses. After that, the metric object is used to compute for every metadata record a score in an iterative fashion.

In Figure 4.3 a simplified class diagram is shown. Basically, there are two super classes that organize the structure. A metric interface from which every quality metric inherits to implement its individual metric algorithm: 1) a metric worker class encapsulating the computation task itself, and 2) a metric worker class instantiates a specific metric class. There is also a generic worker class which might be used for simplistic quality metrics as it is the case with the license metric. If there has to be a more complex initialization phase, then a more specific metric worker is created. As opposed to the metric interface, every metric worker object delegates the rest of the execution back to its parent class. Practically, this means that the last metric worker

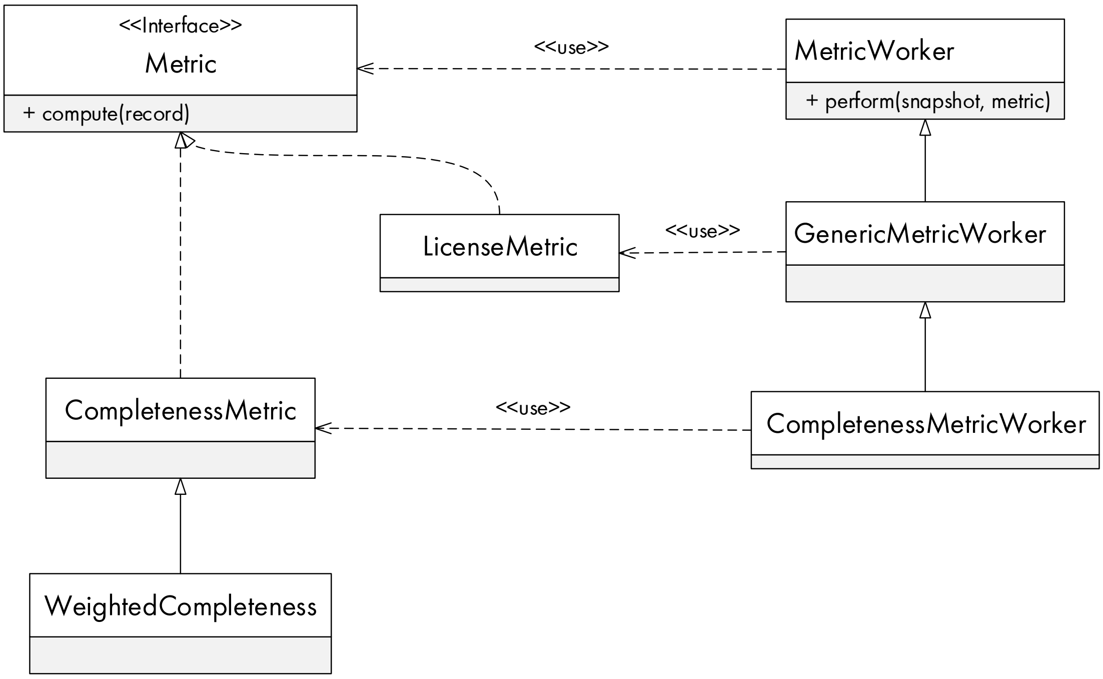


Figure 4.3: Class diagram of quality metrics and metric worker

class in the lookup path is responsible for iterating the metadata records and writing the metric results back into the database.

4.3.5 Metric Report

Essentially, the metric report is the view component which visualizes the results. This can include graphical visualization with the use of charts, but also simple tables. The metric report comprises four main parts that should help to divide the different types of information:

- Aggregate Score
- Score Distribution
- Metric Analysis
- Metric Statistics

The aggregate score is a combination of all metric scores for a selected snapshot. The score distribution is a histogram of records based on their score. Since all scores are presented as percentage, there are ten groups: one for every 10%. The metric analysis might vary dependent on the current metric. It shows the details that go beyond the sole metric score including sub-scores for single fields or highlighting of relevant field values. For metrics where a lot of detail are included in the metadata record, a record comparison approach is chosen. This way it is possible to compare one metadata record with high asserted quality against a metadata record with low asserted quality. Finally, the metric statistics constitute a summary of observations that have been made through the metric. For instance, it provides information regarding, which metadata record group has the best results or which fields have been completed the most often.

4.3.6 Individual Weighting

Not everyone has the same perception of what is important in a metadata record. A user should not be forced to deal with the fact that every metric is equally important. Instead this choice can be delegated to the user by allowing him/her to weight each metric score. There are eleven different weights, from zero to ten.

$$score(snapshot) = \frac{\sum_{i=1}^{n} w_i \cdot q_i(snapshot)}{\sum_{i=1}^{n} w_i}$$

Where w_i is the individual weight for quality metric q_i. The advantage comes through the zero weight. Often it is reasonable to ignore certain metrics completely within score computation. This solution is also an answer to the question of score outliers. There might be a specific reason why a certain score is 0.0.

4.4 Implementation

The implementation section covers the details of the programming part of the development. For instance, it became quickly evident that the metric algorithms cannot be implemented directly as specified in Chapter 3 and be done with it. Therefore, rather an iterative approach was used: 1) implement the metrics based on their specification, 2) run them on a set of metadata records, 3) analyze the results, and 4) extend the metrics to cover more special cases.

4.4.1 Technologies

The whole application is implemented in Ruby. The web framework is Ruby on Rails. The choice of the database reflects the functional requirement of a schemaless data store. A relational database could be used, but instead of squashing JSON data as string serialization into table rows a document-oriented database is used. The first choice was Elasticsearch. Like Solr, Elasticsearch is a search engine. Normally, search engines are used as an addition to the technology stack to efficiently search through the data. There is, however, no aspect of Elasticsearch which does not fulfill the requirements of a database. There was one drawback, which made Elasticsearch not usable. While Elasticsearch is document-oriented and advertised as schemaless, this does not always work out. Complex insert operations with documents that are deeply nested might fail.

Therefore another document-oriented database is used: MongoDB. Here the schema-less characteristic works as aspected. It only comes with the restriction that each document needs to have an `_id` field and that field names may not contain dots `.` or the dollar sign `$`.

For computational tasks, like metadata harvesting or metric computation the background processing framework Sidekiq is used. Sidekiq uses threads to handle multiple working tasks at the same time in the same process. On the front-end CoffeeScript (JavaScript) is used for adding dynamic content to a page.

4.4.2 Algorithms

There is a wide range of subtleties that have to be considered for each metric algorithm. Also what kind of libraries are used to cover the domain targeted by the specific metric?

Completeness

In order to implement the completeness metric the CKAN schema was required. Although all CKAN fields are known, a separate schema file can be used to implement an algorithm which is generic for other metadata structures as well. The CKAN schema is hard-coded in the Python source code of CKAN. In order to solve this in a sustainable fashion a Python tool has been developed to generate a JSON Schema out of the CKAN schema. Even more important is the nested structure of the JSON schema. With both, the metadata records and the schema, the fields can be iterated in a recursive fashion while counting the non-null valued fields. For this task the most important part is to differ between properties and items. A field with a value of type array is considered to be complete, if the array contains at least one item.

Accessibility

The accessibility metric typically resides in the domain of natural language processing. There is a rich library developed at the Standford University in the Natural Language Processing Group including the Stanford CoreNLP, the Stanford Parser and the Stanford Word Segmenter.

This suite is also implemented in Java. While there are features of interoperability to make this library callable, it would always mean to start a Java Virtual Machine (JVM) through the Ruby process. This has impact on the execution time and the memory consumption. One way to integrate this seamlessly is to use JRuby. JRuby is a Ruby implementation on the JVM. This way, however, libraries which are using C extensions cannot be used.

Eventually, this comes down to choosing either MRI Ruby and native C extensions or JRuby and native Java implementations. The former case is followed for the rest of the implementation. The natural language processing has been solved as follows. There are three requirements for the accessibility: word tokenizing, sentence tokenizing and syllables tokenizing.

Word tokenizing can be approached with regular expressions. One way is by using `\w+` which is a meta-character for one or more word character `[a-zA-Z0-9_]`. Apparently, this does not work sufficiently reliable on numbers and Unicode characters. Numbers with a decimal dot, or a delimiter like `12,000` are separated into two strings. The result is that the number is counted as two words where it should only be one.

Until Ruby 1.8 the expression `\w+` worked for Unicode, too. This has changed, when a Unicode character occurs, the word is separated. Since numbers are counted as words in the Flesch-Formula [30] the opposite approach is taken, separate on `\S+` which are non-whitespace characters `/[^ \t\r\n\f]`.

For the accessibility metric, sole word counting suffices. This does not matter if the regular expression can extract the exact word. For the intrinsic precision metric this is important, otherwise the check for common misspellings cannot be made. This has been solved by applying character stripping afterwards, which eliminates special characters from the left and right side of an extracted string.

The second problem of natural language processing is the sentence tokenizing. For this the Ruby gem[1] TactfulTokenizer is used. TactfulTokenizer uses a Naive Bayesian statistical model for extracting sentences. The third problem is to determine the number of syllables. Determining the number of syllables is a problem that cannot be solved reliably for every kind of language. Yet this was approached by using the TeX hyphenation algorithm.

Intrinsic Precision

For the intrinsic precision the misspelling checks are based on the language. This means the language needs to be detected beforehand. For Ruby there is native port of the Chromium

[1] Gems are packages as part of RubyGems, a package manager for the Ruby programming language.

language detection. It works fast and very reliable and what is more, it reports back if the language could not be detected with sufficient confidence.

Availability

The availability metric is computed based on the response of HTTP requests on resource’s URL. A metadata record can contain multiple resources and thus URLs. With a repositories with hundreds to thousands metadata records, this results in many HTTP requests. In order to process these efficiently the gem Typhoeus is used. Typhoeus executes multiple HTTP requests in parallel. It is based on libcurl.

Sending for every URL a HTTP GET request would take far too long. The URLs are pointing to files, downloading every file would generate too much bandwidth and lengthen the metric computation. Instead header requests are made through HTTP HEAD. During iterations of its implementation many servers denied the response. Apparently, some servers use a whitelist for the requesters user-agent. The default user-agent of Typhoeus is not part of this whitelist. The problem was solved by setting the user-agent manually to the one of libcurl which is acknowledged widely.

Another problem arise through HTTP response relocating the resources either temporarily (HTTP 302) or permanently (HTTP 301). Typhoeus, respectively libcurl can handle this with ease if configured correctly. At one point even a redirection loop was found in one of the resources. The proper libcurl configuration is an important asset in the implementation of this metric. In Listing 4.1 the configuration is shown.

```
configuration = { :headers => { ’User-Agent’ => ’curl/7.29.0’ },
  :ssl_verifypeer => false,
  :ssl_verifyhost => 2,  # disable host verification
  :connecttimeout => 60,
  :maxredirs => 50,
  :followlocation => true,
  :method => :head,
  :nosignal => true,
  :timeout => 240 }
```

Listing 4.1: Typhoeus (libcurl) configuration used to deal with a series of edge cases

Accuracy

The accuracy metric deals first and foremost with the specified format of a resource. Similar to the availability metric, it is validated through header requests by inspecting the header's content type. A general problem with MIME types is consistency. Often there is exactly one official identifier for a certain format, but many variations are used as well. One way to deal with this, is to ignore the variation and count this as invalid. Yet this would not reflect the quality of the metadata record, because above all it is a quality issue of the endpoint serving the resource. Hence, an extensive dictionary has been created which maps a given format to a series of possible MIME types. During the time of this thesis this dictionary has been extended continuously.

4.4.3 View Generation

Each quality metric handles different attributes of a metadata record. Thus, the different attributes should be reflected in the choice of visualization. Yet not all metrics are that complex in their result characteristics. Taking the non-functional requirement extensibility into account, it is desirable to have default views that work for every kind of metric as long as their implementation adheres to the design. The view generation is the concept for this implementation. The selection procedure is shown in Listing 4.2. Since the metric classes are hierarchical, too, the views are as well.

The current metric's name is used to look up if there is a specific metric view report. If not, the metric's ancestor are iterated to see if there is a specific view. The iteration stops with the base class. For which the generic metric report view is returned.

```
##
# Selects the partial for displaying the metric report.
#
def select_partial
  partials = ”metrics/partials”
  directory = ”app/views/” + partials

  ancestors = Metrics.from_sym(@metric).ancestors
  ancestors = ancestors.select { |cls| cls < Metrics::Metric }

  ancestors.map { |cls| cls.to_s.demodulize.underscore }.each do |candidate|
    file = ”#{directory}/_#{candidate}.html.erb”
    return ”#{partials}/#{candidate}” if File.exists?(file)
  end

  ”metrics/partials/generic”
end
```

Listing 4.2: Selection of a specific metric report view is based on the current metric

The generic metric report view uses metadata record comparison. Two records are compared side by side. The metric interface states that each field as part of metric computation needs to get a score value, too. These values are retrieved and displayed in this view. It is a superficial solution, but it provides the possibility to add a new metric class and it is integrated without implementing a sophisticated view to make use of it.

4.5 Visualization

For the visualization a range of chart types are used. Every graphical implementation is done by using D3.js. D3 is a JavaScript library for visualizing data with the use of HTML, SVG and CSS. With D3 three types of charts have been implemented.

A histogram is used to illustrate the distribution of metadata records along their score (Figure 4.4). In addition, a so called metric meter is used to present the score. The metric meter is used for both, single metric scores and aggregate scores for a whole snapshot. The idea of the metric meter is to communicate that the score should be understood as a process. With raising percentage the color changes from red over yellow into green. Bar charts are used to

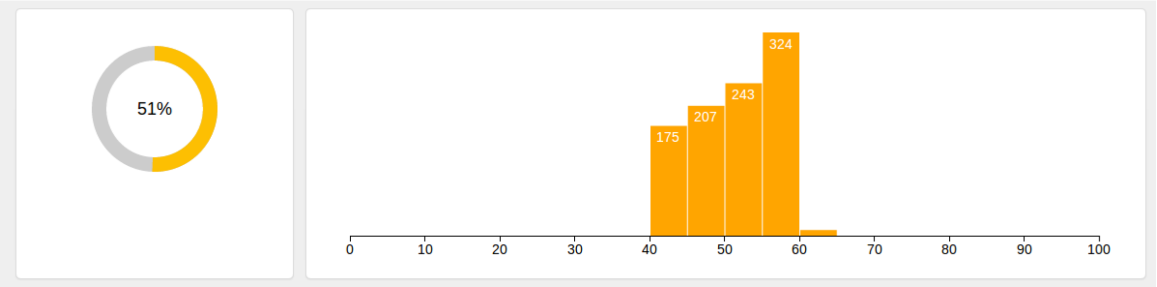


Figure 4.4: Histogram and metric meter used to visualize the score outcome

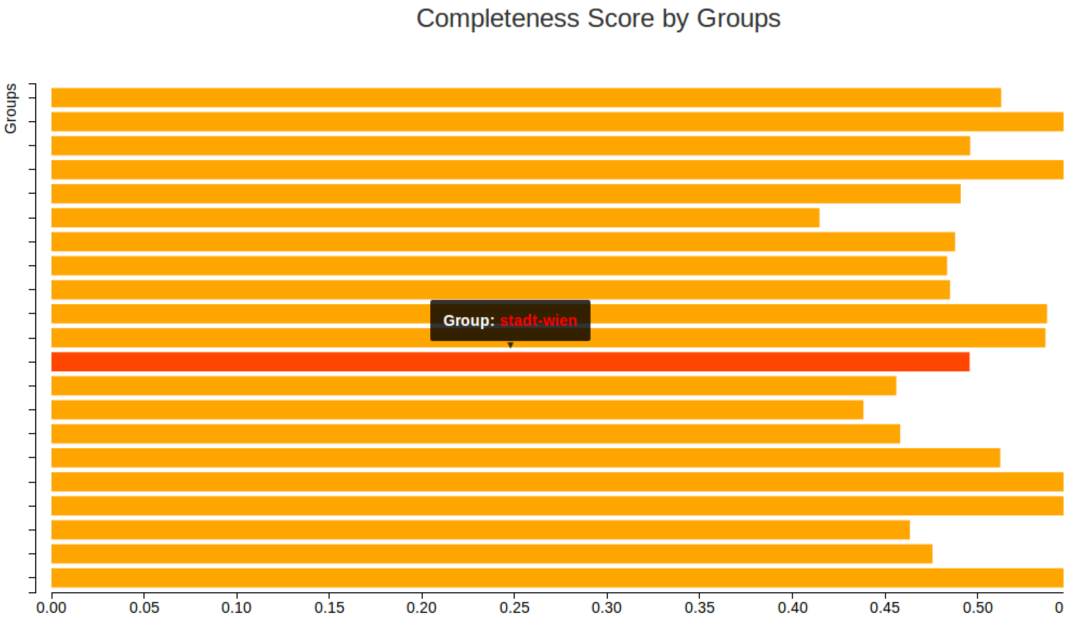
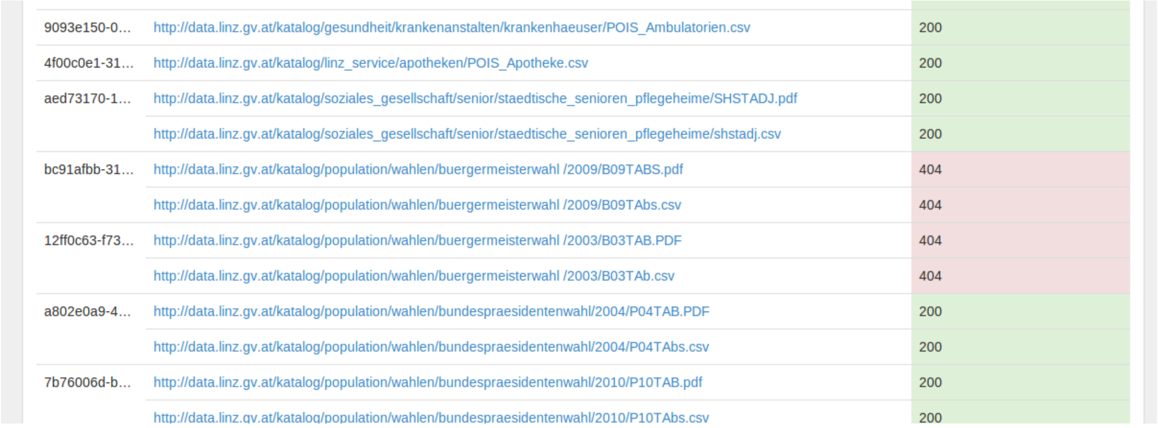


Figure 4.5: Bar chart to visualize the score distribution among groups

show difference between a set of values. For every metric report a score distribution among groups is generated (Figure 4.5).

Pie charts are ideally used to show the relation of a part and the whole. Consequently, an ideal pie chart contains just two numbers. This could be used on the availability metric to show the relation between number of working URLs and number of not working URLs. The availability metric is also a striking example for specific metric reports. In Figure 4.6 it can be seen how the availability metric results are illustrated. Obviously, it would not be very efficient to limit this view to record comparison. It is easier to just look at all the URLs and the HTTP status retrieved in the metric computation.

Finally, a more specialized visualization type is used for the completeness metric: Treemap.



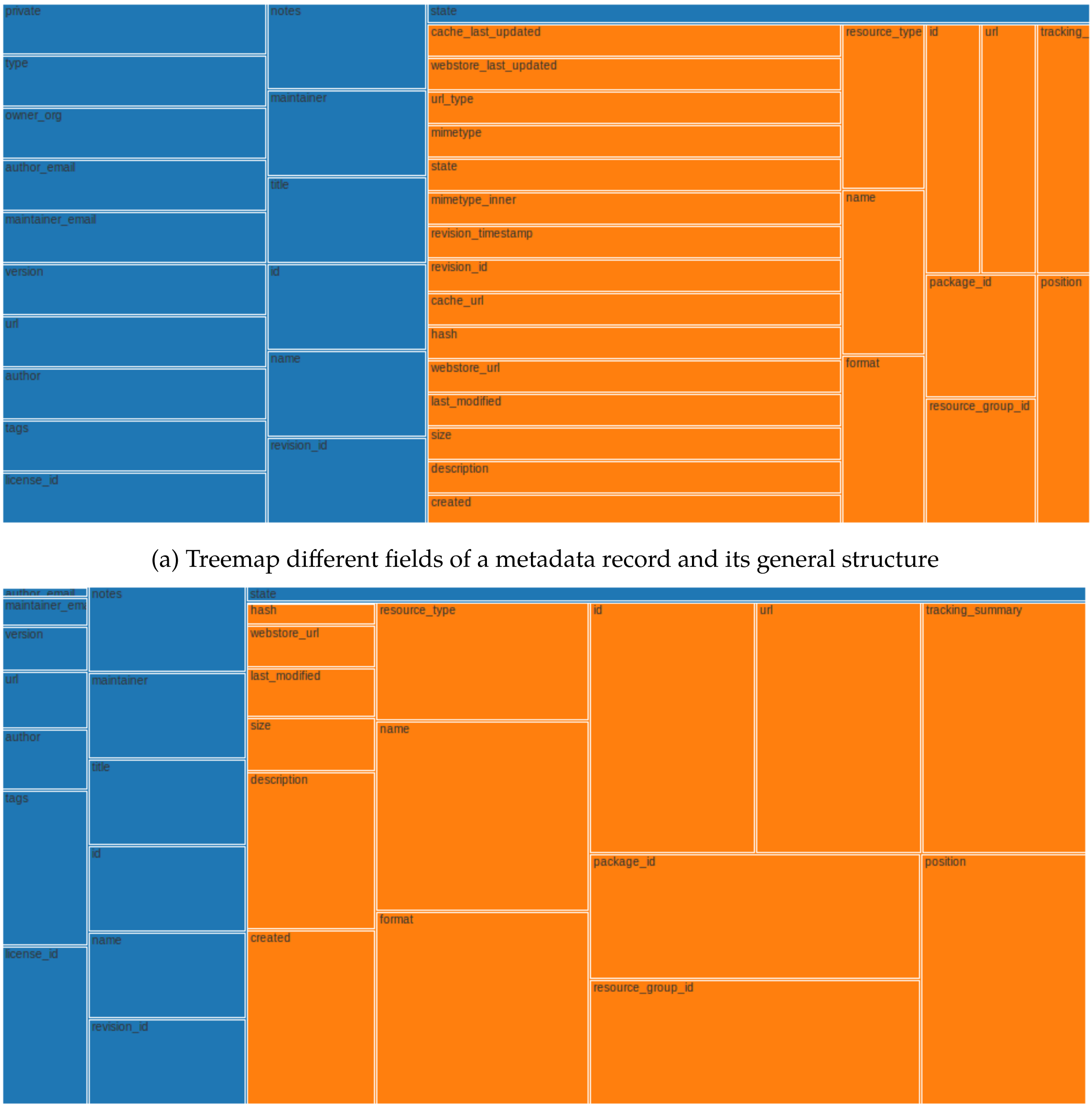
9093e150-0...	http://data.linz.gv.at/katalog/gesundheit/krankenanstalten/krankenhaeuser/POIS_Ambulatorien.csv	200
4f00c0e1-31...	http://data.linz.gv.at/katalog/linz_service/apotheken/POIS_Apotheke.csv	200
aed73170-1...	http://data.linz.gv.at/katalog/soziales_gesellschaft/senior/staedtische_senioren_pflegeheime/SHSTADJ.pdf	200
	http://data.linz.gv.at/katalog/soziales_gesellschaft/senior/staedtische_senioren_pflegeheime/shstadj.csv	200
bc91afbb-31...	http://data.linz.gv.at/katalog/population/wahlen/buergermeisterwahl /2009/B09TABS.pdf	404
	http://data.linz.gv.at/katalog/population/wahlen/buergermeisterwahl /2009/B09TAbs.csv	404
12ff0c63-f73...	http://data.linz.gv.at/katalog/population/wahlen/buergermeisterwahl /2003/B03TAB.PDF	404
	http://data.linz.gv.at/katalog/population/wahlen/buergermeisterwahl /2003/B03TAb.csv	404
a802e0a9-4...	http://data.linz.gv.at/katalog/population/wahlen/bundespraesidentenwahl/2004/P04TAB.PDF	200
	http://data.linz.gv.at/katalog/population/wahlen/bundespraesidentenwahl/2004/P04TAbs.csv	200
7b76006d-b...	http://data.linz.gv.at/katalog/population/wahlen/bundespraesidentenwahl/2010/P10TAB.pdf	200
	http://data.linz.gv.at/katalog/population/wahlen/bundespraesidentenwahl/2010/P10TAbs.csv	200

Figure 4.6: Visualizing the results of the availability metrics: HTTP Status Codes

Treemaps are used to display data which has a hierarchical structure. By using rectangles that are nested into each other the hierarchy is made clear. It is used for the completeness metric, because the completeness metric is about structure, counting fields on different depth levels.

The input data is the metadata schema, whereas each field is associated with the number of completions for this snapshot. The dynamic is exploited to switch between two different modes for the Treemap to interpret the data. This is shown in Figure 4.7. In the first picture the record structure according to the schema is shown. Each color represents the depth. In this case there are only two levels. The top-level (blue) and one level down, the fields of a resource (orange). When swapping the mode, as can be seen in the second picture, then some fields disappear. These are the fields that have only been completed very few times or none at all. Obviously, the same data could be shown in a table where the values are sorted, but it demonstrates how different kind of data can be explored in different ways.

> *"I had botched a great many pieces of wood before I mastered the right angle with a saw, botched even more before I learned to miter a joint. The knowledge of these things resides in my hands and eyes and the webwork of muscles, not in the tools. There are machines for sale—powered miter boxes and radial arm saws, for instance—that will enable any casual soul to cut proper angles in boards. The skill is invested in the gadget instead of the person who uses it, and this is what distinguishes a machine from a tool."*
>
> —Scott Russell Sanders
> Paradise of Bombs: The Inheritance of Tools (1987)



(a) Treemap different fields of a metadata record and its general structure

(b) Treemap visualizing the number of times a field has been completed in the whole snapshot

Figure 4.7: Switching between two Treemap modes to show which fields are completed often and which disappear

Chapter 5

"How can you own these numbers? You know, the numbers belong to the world."
—DONALD KNUTH
Web of Stories (2006)

Case Study: Open Government Data

A method for the automatic quality assessment of metadata has been discussed and implemented: quality metrics. This approach is put to test on open government data portals. The case study reveals how reasonable the algorithmic quality determination works. After looking at the results it should become clear, whether this method can be pursued in future works. Fourteen open government data portals from around the world have been selected for the analysis including data.gov.uk (United Kingdom), GovData.de (Germany), PublicData.eu (European Union), catalogodatos.gub.uy (Uruguay), data.qld.gov.au (Queensland, Australia), data.gc.ca (Canada) and opendata.admin.ch (Switzerland).

5.1 Quantitative Analysis

Before looking at the results of the quality metrics and statistics generated by the application, the repositories are investigated. The data portals have been described as an abstract entity so far. With this analysis it should become more clear in what dimensions the computations are performed. How big are the repositories? Do they grow significantly? How many languages are used? In Section 4.2 the architecture's implementation has been described. The preliminary analyzer component has been used to collect the statistics that are presented in the following.

5.1.1 Size, Type and Location

In Figure 5.1 the different repositories are shown on a world map. Still missing are open data portals from the Asian continent. Nevertheless, the list of repositories covers repositories from very different locations. The importance reside in cultural difference, for instance the language.

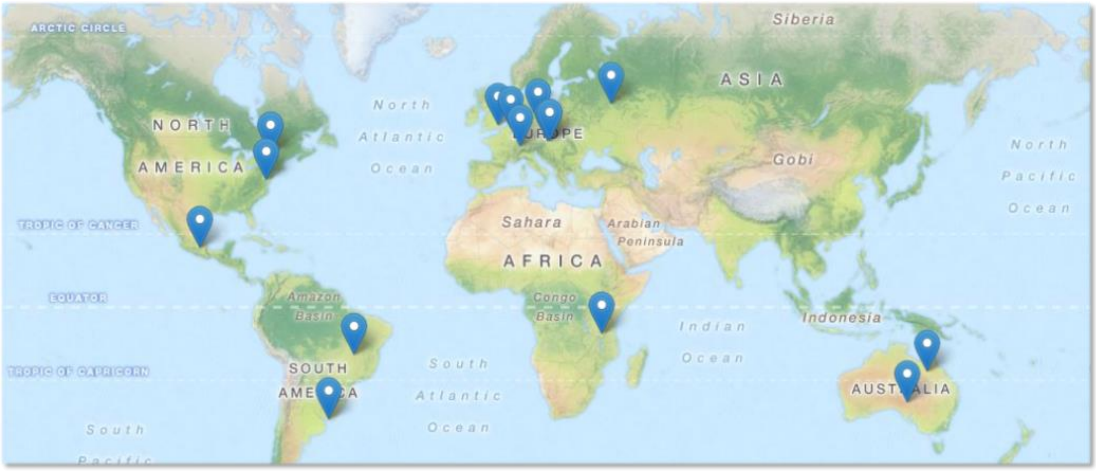


Figure 5.1: World map with the repositories being part of the case study

Repository	Metadata Records	Geographic Location	Type	Snapshot Date
data.gc.ca	197,824	Canada	CKAN	2013-10-14
PublicData.eu	24,250	European Union	CKAN	2013-10-14
data.gov.uk	14,281	United Kingdom	CKAN	2013-10-14
GovData.de	4,596	Germany	CKAN	2013-10-14
opendata.admin.ch	1,264	Switzerland	CKAN	2013-10-14
data.gv.at	958	Austria	CKAN	2013-10-14
africaopendata.org	932	Africa	CKAN	2013-10-14
data.qld.gov.au	500	Australia	CKAN	2013-10-14
data.sa.gov.au	230	Australia	CKAN	2013-10-14
data.gov.sk	250	Slovakia	CKAN	2013-10-14
dados.gov.br	106	Brazil	CKAN	2013-10-14
data.openpolice.ru	68	Russia	CKAN	2013-10-14
catalogodatos.gub.uy	67	Uruguay	CKAN	2013-10-14
datos.codeandomexico.org	31	Mexico	CKAN	2013-10-14

Table 5.1: Size and type of different open government data portals

	Number of Resources				
Repository	Minimum	Average	Median	Maximum	Sum
data.gc.ca	1	7	3	570	1,416,913
PublicData.eu	0	4	1	305	85,232
data.gov.uk	0	4	1	191	50,950
GovData.de	0	3	3	41	13,646
data.gv.at	1	3	1	61	2,913
opendata.admin.ch	2	2	2	4	2,571
data.qld.gov.au	1	5	1	106	2,335
dados.gov.br	0	18	4	810	1,930
africaopendata.org	0	1	1	20	1,058
data.gov.sk	1	3	1	18	513
data.sa.gov.au	1	2	2	17	472
data.openpolice.ru	2	3	3	3	201
catalogodatos.gub.uy	1	3	2	14	200
datos.codeandomexico.org	1	4	1	39	133

Table 5.2: Number of resources per metadata records on a repository

In Table 5.1 the repositories are named explicitly. For this work the convention is used to identify the repositories by their domain name. This is also a convention which proves to be popular as more and more portals use the domain name as the name for their portal. It should be noted that although a geographic location is given it does not necessarily mean that the repository is an official entity provided through the country's government. Yet this was pursued, as opposed to repositories which are created as an effort limited to citizens. Striking is the wide range of number of metadata records. This shows that a scaling implementation for the quality metrics is a necessity.

The metadata records are the wrapping entity for the actual resources. How does the number of resources correlate? In Table 5.2 the number of resources is shown. Besides some repositories having metadata records in the hundreds the order stays the same. The maximum number of resources is sometimes very high. For dados.gov.br this seems to be true for many records resulting in a high average number of resources. It can be expected that a metadata record with too many resources makes it hard to find the appropriate resource. This way of structuring can be observed for open data which publishes their data on a monthly basis. This way the number of resources grow quickly. A short and distinctive resource description is crucial in this case.

The median is helpful here, it proves that there are many outliers. The minimum number of resources has to be seen critical. It is questionable what value a metadata record carries without resources as it is the case for PublicData.eu, data.gov.uk, GovData.de, dados.gov.br and africaopendata.org. PublicData.eu again is a data aggregator harvesting repositories like

Repository	Languages
opendata.admin.ch	German
data.qld.gov.au	English
data.sa.gov.au	English
catalogodatos.gub.uy	Spanish
datos.codeandomexico.org	Spanish
data.openpolice.ru	Russian
dados.gov.br	Portuguese
GovData.de	German, English
data.gov.uk	English, Norwegian
data.gov.sk	Slovak, English, Czech
data.gv.at	German, English, Italian
africaopendata.org	English, Swahili, Spanish
PublicData.eu	English, German, Spanish, Norwegian, Italian, Slovenian, Czech, Danish, Icelandic, French, Hungarian, Croatian, Dutch, Lithuanian, Swedish, Estonian, Slovene

Table 5.3: Languages used in the metadata records

GovData.de and data.gov.uk. Hence, the same minimum value may result from this fact.

5.1.2 Languages

The use of languages has an impact on the user, but also on certain quality metrics. Open data is supposed to be accessible for its consumer. Users might be not familiar with the used language in descriptions and/or resources. Some quality metric implementations need the used language as an input along the metadata record. For the intrinsic precision metric a common misspelling dictionary will be loaded for the prevalent language. For the accessibility metric a variation of the Flesch index needs to be used instead. In Table 5.3 repositories and the languages used by the metadata records are listed. Only languages are included which could have been detected reliably.

The number of different languages stands out for PublicData.eu. This should not come as a surprise as stated before PublicData.eu acts as a pan-European data aggregator, but it also shows how much different regions PublicData.eu already covers. Prevalent languages are German, English and Spanish. It can become costly to support all the languages with respect to the language dependent metrics. This analysis reveals that the focus should be at least on these three languages.

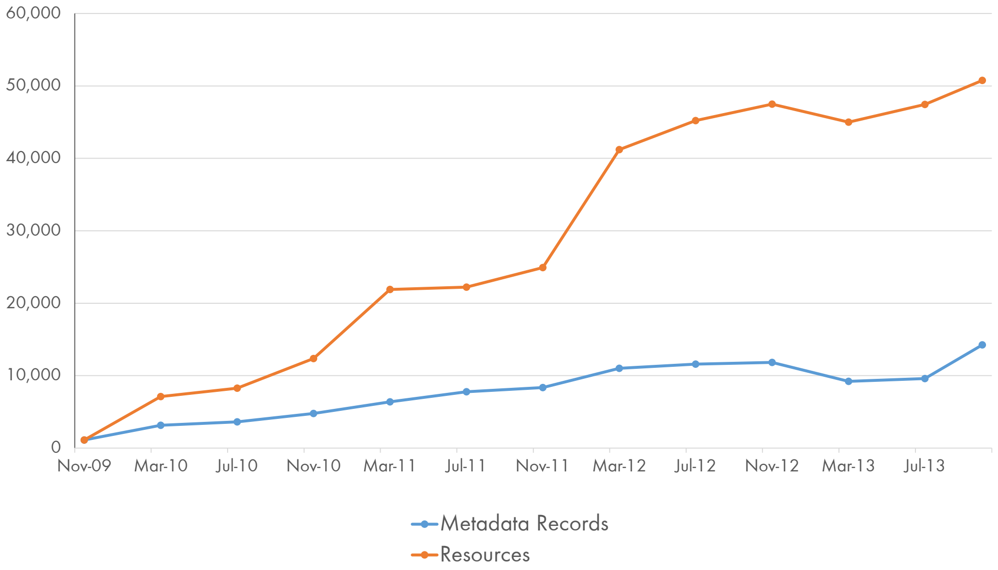


Figure 5.2: Content growth over time of data.gov.uk from 2009 to 2013

5.1.3 Growth over Time

An open government data portal which is not continuously fed with new data loses its relevance. This requires a growth over time with respect to the metadata records, but also resources. It would be interesting to look at this kind of data for all repositories. After all the implementation Metadata Census is designed to monitor the quality change over time. At the time being this data is, however, only available for GovData.de and data.gov.uk. The latter provides an extensive archive of static dumps back to the portal's launch in 2009. For GovData.de, which launched in January 2013, there are two snapshots available: February and August.

data.gov.uk

The growth is visible linear. Interestingly, there was a decrease of total metadata records between November 2012 and July 2013. Such a decrease is a signal for a potential measure to improve the metadata quality. A large quantity of metadata records are dropped, because they are not qualified, licensed properly or for other reasons. It is an indicator at best.

At the first dump, from November 2009, resources and metadata records were encapsulated

Type	February 2013	August 2013
Datasets	1,123	3,797
Documents	12	230
Applications	25	15

Table 5.4: Content growth over time of GovData.de in 2013

by the same entity. Thus, the total number of metadata records equals the total number of resources at that point. This has changed quickly over the time. Especially, after November 2011 the number of resources increased dramatically.

GovData.de

The data portal GovData.de does not exist as long as data.gov.uk. Only for two points in time the number of metadata records can be quantified. In exchange, an additional differentiation is available, the metadata record type. CKAN has an optional field *Type* which, allows to establish a differentiation between different records. The default is dataset, which addresses resources that are available in a machine-readable format like CSV or JSON. This kind of differentiation is not available on data.gov.uk. There the field *Type* is simply never set, respectively it is always *null*. This shows the difficulty in the approach to compare different repositories. The result is shown in Table 5.4. Documents, as opposed to datasets, are not machine-readable and include formats like PDF or word processing documents.

Similar to data.gov.uk the growth is linear. The number of documents increased dramatically. Interestingly the total number of applications dropped from 25 to 15. This has to do with metadata quality as well. The constraint for the record type *Application* on GovData.de is that the application is based on datasets that are actually published under an open license. This constraint was not met for all the application, for which reasons they were removed.

5.1.4 Creation and Update Time

New metadata records are added, but what about the updating of existing metadata records? Does a repository just grow, adding new metadata records while ignoring metadata records that became obsolete, or are old metadata records also updated? The total record decline in data.gov.uk hinted this. More insight would give an analysis, showing when and how much metadata records were created initially and when and how much metadata records have been updated.

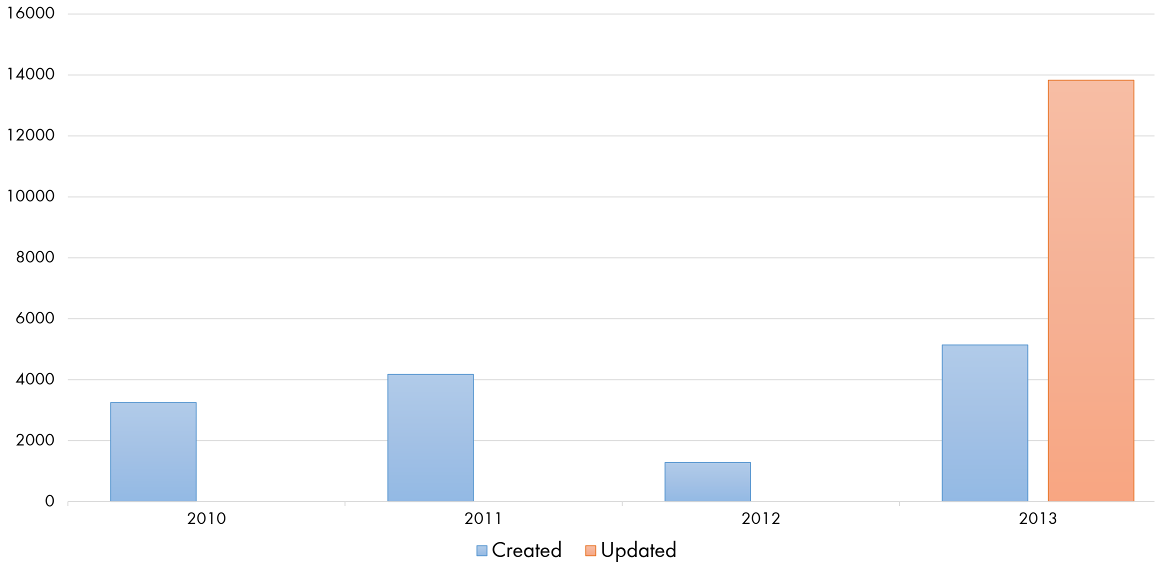


Figure 5.3: Number of metadata record creations and updates over time of data.gov.uk

In Figure 5.3 the distribution of creations and updates of data.gov.uk is presented in a bar chart. Interestingly, it looks like metadata records have not been updated before 2013. In fact, it just means that there are no metadata records which have not been updated in 2013. This is a sign for measures to improve the metadata quality. For instance, the schema changes or data is added to every record. This is considered to be a good sign.

In Figure 5.4 the same chart is plotted for the data of GovData.de. Here, of course, in a time span which is a lot shorter. Due to the fact, that the data is grouped by months, there are much more record updates. Again, there is a short time span in which more metadata records have been updated than usually.

This data which has been presented in this section are not attributes for metadata quality, but it helps to look at some quantitative characteristics to get an overview of how the data is managed and changes. Furthermore, it can be used to find groups of metadata records which may be useful for a more detailed metric analysis. In the next section, the actual metadata quality results are presented and investigated.

5.2 Metric Scores

The results of the metric computations are shown in Table 5.5. The repositories are sorted by their aggregated score. What conclusions can be drawn here? In order to understand the results better, each metric will be investigated in detail for a selected repository.

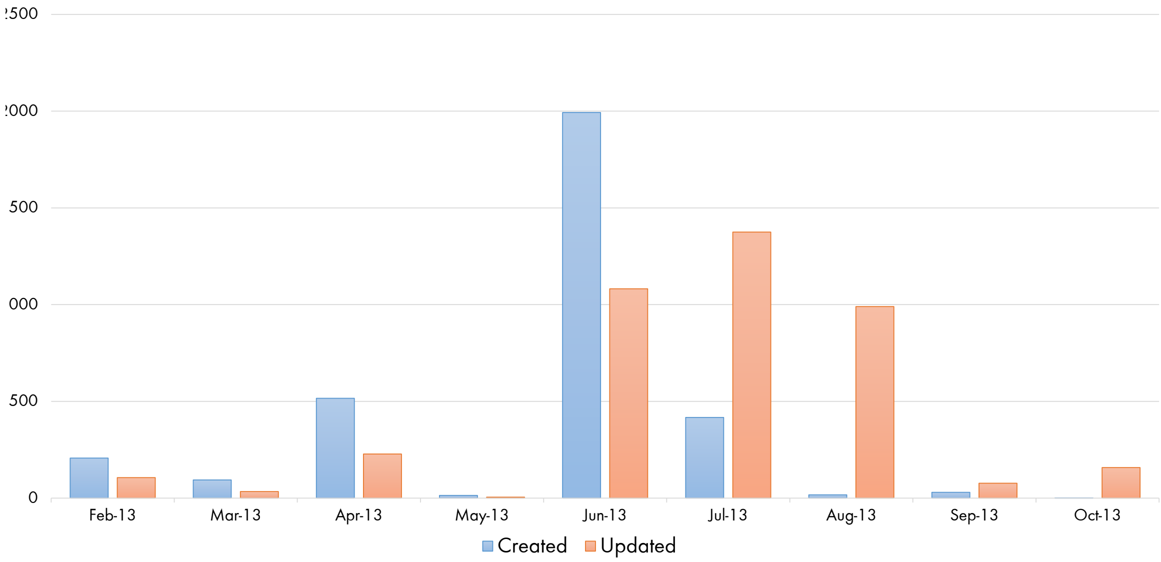


Figure 5.4: Number of metadata record creations and updates over time of GovData.de

Rank	Repository	**Score**	Intrinsic Precision	Richness of Information	Licenses	Completeness	Availability	Weighted Completeness	Accessibility	Accuracy
1	data.gc.ca	74	97	86	80	79	79	81	71	20
2	data.sa.gov.au	71	98	63	94	77	86	82	72	0
3	GovData.de	67	99	44	38	55	81	87	79	56
4	data.qld.gov.au	66	99	67	96	73	60	78	59	0
4	PublicData.eu	66	98	84	69	64	70	67	42	32
4	data.gov.uk	66	97	85	69	62	74	67	44	28
4	africaopendata.org	66	100	20	78	70	87	68	55	53
5	datos.codeandomexico.org	65	100	55	84	65	100	75	37	0
6	catalogodatos.gub.uy	63	100	64	1	70	74	78	65	52
6	data.openpolice.ru	63	100	0	0	58	100	81	100	64
7	dados.gov.br	61	100	87	36	53	57	72	44	39
8	opendata.admin.ch	59	100	12	0	58	100	68	35	100
9	data.gv.at	57	100	21	99	51	68	65	59	0
10	data.gov.sk	49	100	51	0	48	92	58	37	7

Table 5.5: Leaderboard ranking different government data portals by their metadata quality

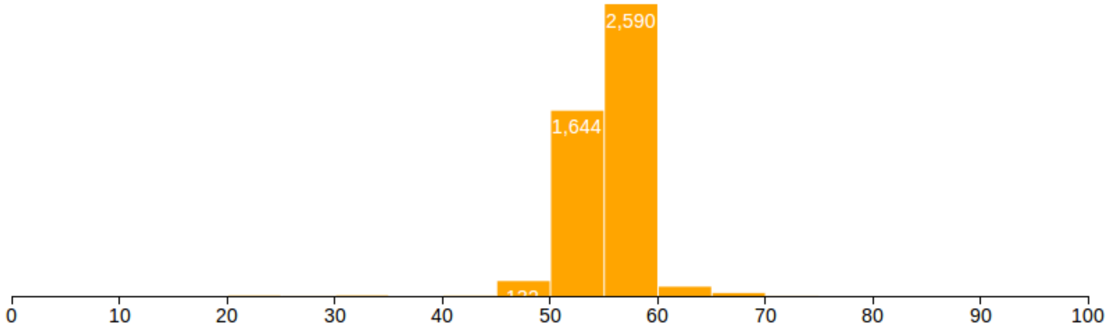


Figure 5.5: Completeness score distribution

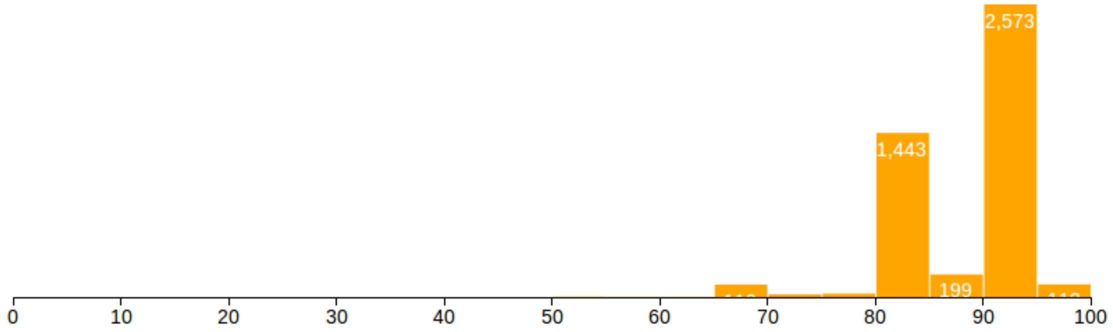


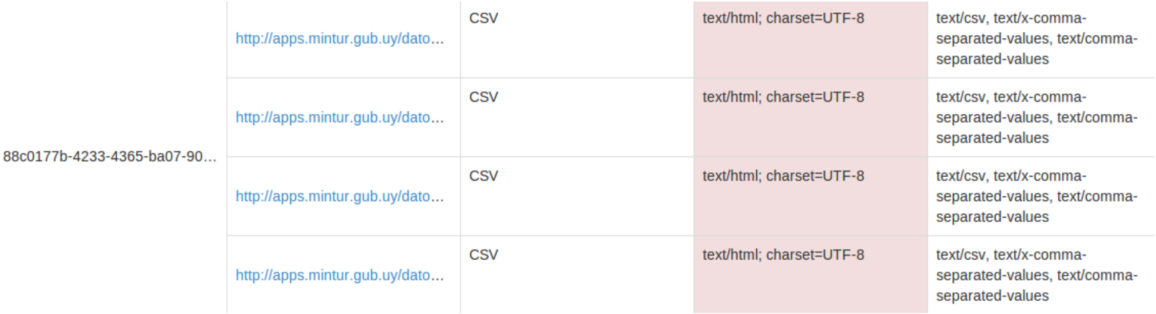
Figure 5.6: Weighted completeness score distribution

5.2.1 Completeness

A typical result for the completeness metric can be seen in Figure 5.5. There are no metadata records, which fill out every available field. There are only few fields which are never used at all, for instance *Owner Organization* and *Resource URL Type*. Mostly this is questionable anyway, if these fields add any value to the metadata record. Some fields are meta-metadata for the repository itself. Other fields like *Resource Hash*, *Resource Size*, but also *Maintainer Email* are seldom completed. Actual quality impacts arise on fields which are more important. For example if the field *Author* has only been completed on 80% of the records, the focus should be to improve the remaining 20%.

5.2.2 Weighted Completeness

With field weighting the overall score increases. This can be seen in Figure 5.6, where the mean is now in the higher score ranges. Now there are metadata records which satisfy the



88c0177b-4233-4365-ba07-90...	http://apps.mintur.gub.uy/dato...	CSV	text/html; charset=UTF-8	text/csv, text/x-comma-separated-values, text/comma-separated-values
	http://apps.mintur.gub.uy/dato...	CSV	text/html; charset=UTF-8	text/csv, text/x-comma-separated-values, text/comma-separated-values
	http://apps.mintur.gub.uy/dato...	CSV	text/html; charset=UTF-8	text/csv, text/x-comma-separated-values, text/comma-separated-values
	http://apps.mintur.gub.uy/dato...	CSV	text/html; charset=UTF-8	text/csv, text/x-comma-separated-values, text/comma-separated-values

Figure 5.7: Mismatch between resources' format and MIME type

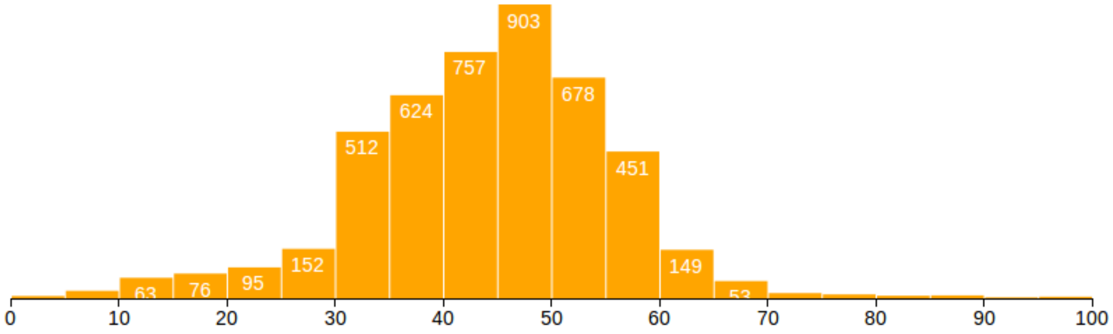


Figure 5.8: Richness of information score distribution

completeness for every field. For quality improvement, especially the records at the lower score end are interesting.

5.2.3 Accuracy

The accuracy has the worst overall results for most of the repositories. In most cases the advertised resource format could simply not be verified through the MIME type. The resources' file size is a field which is seldom set at all, but if it is set it is correct. A typical case can be seen in Figure 5.7. Surprisingly, there are sometimes odd exceptions. The Swiss open data portal opendata.admin.ch has an accuracy of 100%. There are 2,571 resources and there the only format used are Microsoft Excel spreadsheets verified through the MIME type `application/vnd.ms-excel`.

5.2.4 Richness of Information

Apparently, the richness of information score is often normally distributed (Figure 5.8). The tables with the tf-idf data does not reveal much information. As explained, tf-idf scales down frequent terms, while scaling up rare terms. For a user it might be very unclear, how the result of the richness of information metric comes into being. Another problem arises through the normalization: outliers. Due to outliers, a lot of scores are scaled down. This could be improved by either trying to detect outliers or apply the logarithm.

5.2.5 Accessibility

The accessibility metric does not reveal a lot information. Some repositories do better, some do worse, but when investigating the results something becomes evident: most descriptions are too short. The Flesch index is supposed to measure the readability of a text and not of a description consisting of four keywords. Some of these very short descriptions do well, some do not.

5.2.6 Availability

The availability metric which is effectively a link checker is one of the most useful metrics and easy to analyze. Due to the HTTP responses it becomes quickly clear, why a certain resource is not available. Invalid responses include HTTP 502 (Bad Gateway), HTTP 404 (Not Found), but also exceptions where the client failed to receive data from the peer. If there are many URLs that are not working, then it is mostly due to time outs. In the experimenting phase, the number of valid URLs could be increased dramatically by increasing the connection timeout.

5.2.7 Intrinsic Precision

The intrinsic precision metric detects some typical typos. One obvious flaw is the number of missing misspelling dictionaries. Not all available languages are covered, hence some high score results. It is questionable how to proceed in this case. Most fair would be to remove it from the score until it can be checked in a feasible fashion.

5.3 Summary

The case study showed the applicable of most of the metrics, but also revealed problems which need to be solved to improve the metric's significance and thus validity. It is beneficial to test the quality metrics on repositories from very different locations. The cultural influence adds many edge cases that need to be considered. This can be of technical nature, for instance encoding clashes between ASCII, UTF-8 and the database, but also of more algorithmic nature.

Chapter 6

Evaluation

> *"There are definitely some dark futures ahead of us and again a good reason to work on stuff that matters, because we may not have the luxury of working on trivia and I'd love just to have people take seriously that we need to make a better world."*
>
> —Tim O'Reilly
> SXSW 2011

The experimental results demonstrated the applicability of the developed platform. Already this outcome shows that there are many subtleties which need to be investigated in more detail. In this chapter the quality metrics are evaluated, the whole solution approach is discussed and eventually the research result is concluded with an outlook for future work.

6.1 Analysis

Each metric has its own algorithm to reveal problems in metadata records and thus the whole repository. There are many boundary conditions which are not covered yet by the algorithms. It should be clear, that the used set of quality metrics is not sufficient to describe the full range of quality aspects.

6.1.1 Completeness

The completeness metric is based on the simple principle of counting. Either a value is present or not. A human can have a very different notion of completeness. One problem not being addressed by the completeness metric is the content of a field. For instance, the CKAN *Tag* field is an array. An array is considered complete if it contains at least one value. Just one tag for a metadata record is still rather incomplete, but is counted as completed nonetheless. It is hard to imagine that the possible keywords for a metadata record is limited to just one term.

These types of addition can be covered if additional semantics are available. For instance, by providing further semantic rules as input. The insensitivity regarding field priorities is already handled by the weighted completeness metric.

6.1.2 Weighted Completeness

The weighted completeness metric is obviously an improvement to one disadvantage of the completeness metric. Why does the completeness metric not replace the simple completeness metric altogether? The completeness metric has a statistical significance. It is easily conceivable to move this statistical function into the analysis part of the metric. This is due to the metric design which allows to collect further information while iterating the records.

6.1.3 Accuracy

Staying in the context of completeness it can be stated that the accuracy metric is too incomplete. Measuring the semantic distance between the resource format field and the resource file size field does not provide sufficient knowledge about the accuracy. From the semantic point of view, the most interesting distance is the one between description and resource content.

Then again, the problem resides in the implementation for cases like dataset resources. How can the content of a table be validated against a description? Some CSV files are limited to their sheer numbers. Some quality attributes cannot be assessed using automatic methods only. Semi-automatism needs to become part of the assessment process. User feedbacks have great importance and can become an invaluable part of the assessment loop.

In general, the accuracy metric is an example for a quality metric which needs an extensive quantitative analysis beforehand. Without knowledge about number of different MIME types or possible HTTP responses the accuracy metric score will be for unjust reasons too low for many metadata records.

6.1.4 Richness of Information

The richness of information metric uses proven methods from the field of information retrieval. A typical recommendations for the creation and maintenance of open data is to use a uniform vocabulary to increase the overall recognizability. Taking the tf-idf approach into account this type of strategy would not be very beneficial for the metric results. Terms that are used across many documents are simply scaled down.

Then again, for categorical values the richness of information can give valuable information. As a matter of fact, CKAN uses the *Tag* field for indexing the search engine. With too many metadata records using the same tag it becomes harder to find a certain metadata record. This influences the discoverability and adding more unique keywords could improve this.

6.1.5 Accessibility

The Flesch Reading Ease is a widely acknowledged index for measuring the cognitive complexity of a text. A quality metric is supposed to be a surrogate for a more broader quality aspect. Thus, the function definition of a metric could be improved by including further data. Using an aggregate of different scores can help. For instance, the accessibility of a metadata record ought to state the complexity of a record. While a text can be complex, images can help to reduce the complexity. Therefore it thinkable to combine the Flesch Reading Ease with number of images in a metadata record:

$$0.7 \cdot \text{Flesch Reading Ease} + 0.3 \cdot \text{Number of Images}$$

6.1.6 Availability

The availability metric is very functional with respect to its implementation. Either a URL is reachable or not. What has not been considered so far is temporal availability. A availability of 100% for servers is very unlikely. Hence, the availability metric can be improved by replacing a one time check with a continuous checking of the availability.

6.1.7 Intrinsic Precision

Like other quality metrics one important asset is additional data input. The intrinsic precision metric can only be applied when there is a dictionary for common misspelling in the language of the metadata record. Not always is there a database available for this kind of information. It becomes clear, that here too, extensive data mining can help to improve the overall applicability.

6.2 Missing Quality Metrics

The quality metrics were picked for their modular characteristics. The implementation Metadata Census was built with the non-functional requirement extensibility in mind. The present metrics may not cover the full quality range, but it is a foundation for more metrics.

- **Discoverability.** The discoverability is another non-functional requirement. How easy can a metadata record be found? As stated before, there are fields which are indexed by the search engines. With every additional tag, a metadata record might be easier to find. Also more distinctive titles would improve this.

- **Coherence.** The freedom of semi-structured metadata comes with the price of potentially invalid field values. A maintainer's email might not be an actual email address. With additional rules, for instance expressed through regular expressions, these properties can become testable as well.
- **Advancement.** Metadata records need to be maintained. Data becomes outdated, laws are changing, etc. Like the timeline implementation, a dedicated metric can be used to measure the quality improvement over time. This can also include other attributes like the last time the record was updated.
- **Reputation.** Reputation and provenance information are still rare in government data repositories. The data is invaluable. Obviously, this requires a higher engagement with open data portals, but if this kind of data is available it should become quickly part of the assessment.

6.3 Towards Low Quality Detection

The purpose of this master's thesis was the research of a method to assess metadata quality. For this a number of attributes were discussed and analyzed. Certain attributes clearly influence the quality of metadata. Their quantification was addressed by quality metric functions. Effectively, metrics are used to measure these attributes. Although a quantification is performed it became quickly evident, that they do not cover every possible quality characteristic. This type of quantification cannot satisfy a metadata quality assessment completely.

Obviously, the proposed method has a clear weakness. Quality is understood as excellence. The use of an algorithmic approach is too limited to discover all subtleties that result in quality flaws. However, keeping the actual objective of improving metadata quality, this is not necessary at all. The importance does not reside in creating excellent metadata records, but in improving those who have a very low quality. The implementation Metadata Census provides ways to sort these records out. For instance, the quality distribution histogram can list those which have a very low quality. From there on, a repository can be advanced heavily by improving this group of metadata.

A platform like Metadata Census has two functions. On the one hand as an investigation tool to find metadata of low quality and on the other hand as a beacon. Open data is instrumentalized and so can metadata quality. A leaderboard, such as implemented, can be used to engage data provider in improving their metadata. This, of course, requires publicly acceptance of such a tool. Hence, the continuous research for more sound methods is inevitably.

6.4 Future Work

Some metrics originate from a research field of its own. There are many ways to improve each one of them. Besides, the technical implementation is an early design. There are many ways to improve its functions, as well as the function's behavior.

Repository Support CKAN is just one repository software. Socrata and Microsoft's OGDI Data Lab were discussed, too. The metric classes are specified against a set of fields. This input is important as it defines where to look for certain values when analysing a record. This input can be specified for other repositories, too. For metadata which is not based on tree-like data structure it is more difficult. Above all, the function definitions are abstract. Data extraction layers can be implemented to deliver the relevant data for computing the metric scores.

Metadata Revision System With every dump for all the repositories, the database grows linearly. Already the analyzed repositories consume 18 GB of database space on the file system. Previous snapshots could be removed and archived with compression. This does not serve the purpose of the implementation: being able to go back in time and inspect the state of an earlier version of the repository. Instead the structural difference between JSON documents could be computed. A revision system for JSON would allow to have one basis JSON document for every record and with each new snapshot, the documents are compared and only the difference is persisted in the database.

Quality Feed The repositories often offer news feeds about record changes. As an alternative approach this data could be evaluated as well. If the information in this field is complete with respect to the whole repository, then similar computations could be run on this data, too.

Domain-Specific Language The individualization is crucial in the context of quality. Defining one's own quality metrics is a desirable goal. Due to its syntax Ruby offers rich ways to implement domain-specific languages (DSL) natively. It would be of great interest, to develop such a DSL for the metric computation. In the implementation it became clear, that there is a lot of redundancy. This redundancy could be extracted into suitable blocks creating a language to define quality metrics.

Bibliography

[1] Open Knowledge Foundation. Open Definition. http://opendefinition.org/okd/, June 2011. Retrieved October 16, 2013.

[2] Sunlight Foundation. Guidelines for Open Data Policies. http://sunlightfoundation.com/opendataguidelines, August 2013. Retrieved October 16, 2013.

[3] D. Lathrop and L. Ruma. *Open Government: Collaboration, Transparency, and Participation in Practice*. Theory in Practice. O'Reilly Media, 2010.

[4] J. Tauberer. *Open Government Data*. Joshua Tauberer, 2012.

[5] Tony Gill, Anne J. Gilliland, Maureen Whalen, and Mary Woodley. *Introduction to Metadata*. Getty Research Insitute, Los Angeles, online edition, version 3.0 edition, 1998.

[6] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Data Management Systems Series. Morgan Kaufmann, 2000.

[7] J.M. Juran and A.B. Godfrey. *Juran's quality handbook*. Juran's quality handbook, 5e. McGraw Hill, 1999.

[8] Jenny Walker. *New Resource Discovery Mechanisms*, pages 78–89. UKSG, March 2006.

[9] Marieke Guy, Andy Powell, and Michael Day. Improving the Quality of Metadata in Eprint Archives. *Ariadne*, 38, 2004.

[10] Jehad Najjar, Stefaan Ternier, and Erik Duval. The actual use of metadata in ariadne: an empirical analysis. In *In Proceedings of ARIADNE Conference*, pages 1–6, 2003.

[11] Norm Friesen. International LOM Survey: Report (Draft). http://hdl.handle.net/10150/106473, July 2004. Retrieved October 16, 2013.

[12] Jung ran Park and Ann Bui. An Assessment of Metadata Quality: A Case Study of the National Science Digital Library Metadata Repository. In Haidar Moukdad, editor, *Information Science Revisited: Approaches to Innovation*, June 2006.

[13] Jane Greenberg, Maria Cristina Pattuelli, Bijan Parsia, and W. Davenport Robertson. Author-generated dublin core metadata for web resources: A baseline study in an organization. *Journal of Digital Information*, 2:38–46, 2001.

[14] Naomi Dushay and Diane I. Hillmann. Analyzing metadata for effective use and re-use. In *Proceedings of the 2003 international conference on Dublin Core and metadata applications: supporting communities of discourse and practice—metadata research & applications*, DCMI '03, pages 17:1–17:10. Dublin Core Metadata Initiative, 2003.

[15] Jane Barton, Sarah Currier, and Jessie M. N. Hey. Building Quality Assurance into Metadata Creation: An Analysis Based on the Learning Objects and E-Prints Communities of Practice. In *In DC-2003: Proceedings of the International DCMI Metadata Conference and Workshop.*, 2003.

[16] S. Currier, J. Barton, R. O'Beirne, and B. Ryan. Quality assurance for digital learning object repositories: issues for the metadata creation process. *Research in Learning Technology*, 12(1):5–20, 2004.

[17] T. R. Bruce and D. Hillmann. *The Continuum of Metadata Quality: Defining, Expressing, Exploiting*, pages 238–255. American Library Association, Chicago, 2004.

[18] W.E. Moen, E.L. Stewart, and C.R. McClure. Assessing metadata quality: findings and methodological considerations from an evaluation of the us government information locator service (gils). In *Research and Technology Advances in Digital Libraries, 1998. ADL 98. Proceedings. IEEE International Forum on*, pages 246–255, 1998.

[19] Besiki Stvilia, Les Gasser, Michael B. Twidale, and Linda C. Smith. A Framework for Information Quality Assessment. *JASIST*, 58:1720–1733, 2007.

[20] X Ochoa. *Learnometrics: Metrics for Learning Objects*. PhD thesis, Katholieke Universiteit Leuven, 2008.

[21] Baden Hughes. Metadata Quality Evaluation: Experience from the Open Language Archives Community. In Zhaoneng Chen, Hsinchun Chen, Qihao Miao, Yuxi Fu, Edward Fox, and Ee-peng Lim, editors, *Digital Libraries: International Collaboration and Cross-Fertilization*, volume 3334 of *Lecture Notes in Computer Science*, pages 320–329. Springer, 2005.

[22] Thomas Margaritopoulos, Merkourios Margaritopoulos, Ioannis Mavridis, and Athanasios Manitsaris. A conceptual framework for metadata quality assessment. In *Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications*, DCMI '08, pages 104–113. Dublin Core Metadata Initiative, 2008.

[23] Norman E. Fenton and Shari Lawrence Pfleeger. *Software Metrics: A Rigorous and Practical Approach*. International Thomson Computer Press, Boston, MA, USA, 2nd edition, 1996.

[24] Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, Cambridge, MA, USA, 2002.

[25] Xavier Ochoa and Erik Duval. Quality Metrics for Learning Object Metadata. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006*, pages 1004–1011. AACE, 2006.

[26] Claude E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[27] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[28] Stephen Robertson. On GMAP: And Other Transformations. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 78–83, New York, NY, USA, 2006. ACM.

[29] Sri Devi Ravana and Alistair Moffat. Score Aggregation Techniques in Retrieval Experimentation. In *Proceedings of the Twentieth Australasian Conference on Australasian Database - Volume 92*, ADC '09, pages 57–66, Darlinghurst, Australia, Australia, 2009. Australian Computer Society, Inc.

[30] W.H. DuBay. *Unlocking Language: The Classic Readability Studies*. Impact Information, 2007.