Semantic Enrichment of Open Data on the Web

Or: How to build an Open Data Knowledge Graph

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Dipl.-Ing. Sebastian Neumaier
Registration Number 0925308

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.-Prof. Dr. Axel Polleres

The dissertation has been reviewed by:

Prof. Dr. Christian Bizer

Prof. Dr. Elena Simperl

Vienna, 20th November, 2019

Sebastian Neumaier

The research presented herein was funded by the Austrian Research Promotion Agency (FFG) under the projects ADEQUATe (grant no. 849982) and CommuniData (grant no. 855407).

Declaration of Authorship

Dipl.-Ing. Sebastian Neumaier

I hereby declare that I have written this Doctoral Thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

Vienna, 20th November, 2019

Sebastian Neumaier

Acknowledgements

It is a pleasure to acknowledge and thank the many people who helped, in one way or another, to write this dissertation. The past years were full of challenges, drawbacks, learning, and achievements, and I am glad to have made this experience.

I first want to thank my supervisor Axel Polleres. Without his support, encouragement, motivation, creativity, and – above all – scientific guidance this thesis would not have been possible.

A very special thanks to Jürgen Umbrich with whom I have worked very closely and successfully over several years. His support and guidance, in particular at the start of my studies, was of invaluable help and I am very grateful for that.

I want to thank all my colleagues and co-authors for all their feedback and discussions – a particular thanks to Javier, Vadim, Svitlana, Sylvain, Hans, Josi, Erwin, Sabrina, and all those who have helped during my studies.

I am also thankful to my examiners Christian Bizer and Elena Simperl for their critical review and constructive feedback on an earlier version of this thesis.

Finally, I want to thank my friends, my family, and my partner Felicitas for all their support, love, and understanding.

Kurzfassung

In den letzten Jahren konnte vermehrt der Trend hin zu “Open Data” Portalen beobachtet werden: Damit ist das Publizieren von Informationen des öffentlichen Sektors an einem “single point of access” gemeint, mit dem Ziel die Transparenz und das öffentliche Engagement zu fördern. Aus wissenschaftlicher Sicht stellen die großen Datenmengen, die auf diesen (Regierungs-)Datenportalen veröffentlicht werden, eine vielversprechende Quelle zur Integration in das “Web of Data” dar. Natürlich bringen solch große Datenmengen immer auch das Problem der Datenqualität mit sich; vor allem wenn diese von verschiedenen Quellen stammen und dadurch sehr heterogen publiziert werden. Diese Heterogenität – vor allem der Metadaten – wirkt sich auf die Such- bzw. Auffindbarkeit, und damit auf die allgemeine Nutzbarkeit der Daten aus. In der vorliegenden Dissertation werden spezifische, und für diesen Zweck relevante Qualitätsmetriken definiert, und anhand dieser die Qualität zahlreicher Open Data Portale gemessen und analysiert.

Die Forschung am Semantic Web liefert die Technologien um ein Web of Data umzusetzen; so kann zum Beispiel mittels semantischer Suchfunktionen eine portalübergreifende Suche nach zugehörigen und verwandten Daten umgesetzt werden. Der Prozess der automatisierten Datenintegration hängt aber natürlich sehr von der Datenqualität ab. Mithilfe des in dieser Arbeit durchgeführten Qualitätsberichts konnten zahlreiche Qualitätsmängel, sowie ein sehr heterogenes Metadatenvokabular feststellt werden. Um diese Qualitätsmängel zu bereiningen, werden in dieser Dissertation mehrere, auf Semantic Web Technologien basierende, Ansätze präsentiert: Es wird ein Algorithmus zur Wiederherstellung von semantischen Informationen von tabellarischen, offenen Daten evaluiert, sowie weitere Ansätze, die zeitliche und räumliche Zugehörigkeit und Granularität von Datensätzen zu extrahieren. Als grundsätzliches Ziel dieser Arbeit kann einerseits das Verbessern der Qualität von verfügbaren offenen Daten, und andereseits das Integrieren der semantischen Informationen in einen Open Data Wissensgraph ausgegeben werden.

Abstract

In the past years Open Data has become a trend among governments to increase transparency and public engagement by opening up national, regional, and local datasets. A huge amount of datasets became available that could potentially be integrated and linked into the Web of (Linked) Data. However, with the increasing number of published resources, there are a number of concerns with regards to the quality of the data sources and the corresponding metadata, which compromise the searchability, discoverability and usability of resources. In this work, we define quality dimension and metrics, and subsequently report findings based on a continuous monitoring and quality assessment of numerous Open Data portals.

Semantic Web technologies provide enhanced search functionalities and allow to explore related content across data portals. However, as our report shows, current Open Data lacks in sufficient data quality, rich/consistent descriptions, and uniform vocabularies. Having identified and measured the existing quality issues, we outline methods to restore the quality of published resources, methods to recover the semantics of tabular Open Data, and methods to extract taxonomic, spatial and temporal information. Eventually, the aim of this work is to improve the overall quality and value of Open Data and to use the extracted semantic information to build an Open Data Knowledge Graph.

Contents

CHAPTER 1

Introduction

In the last decade we have seen the *World Wide Web* being populated more and more by "machines". The Web has evolved from its original form as a network of linked documents, readable by humans, to more and more a Web of data and APIs. That is, nowadays, even if we still interact as humans with Web pages, in most cases (i) the contents of Web pages are generated from databases in the back-end, (ii) the Web content we see as humans contains annotations readable by machines, and even (iii) the way we interact with Web pages generates data (often without the users being aware of), collected and stored again in databases around the globe. It is therefore valid to say that the Web of Data has become a reality and – to some extent – even the vision of the *Semantic Web*.

In fact, this vision itself evolved over the years, starting with Berners-Lee et al.'s seminal article in 2001 [Berners-Lee et al., 2001] that already envisioned the future Web as "federating particular knowledge bases and databases to perform anticipated tasks for humans and their agents". Based on these ideas a lot of effort and research has been devoted to the World Wide Web Consortium (W3C) Semantic Web activity,[1] which in 2013 has been subsumed by (i.e., renamed to) "Data Activity".[2]

Data published according to the Semantic Web principles has become an important source of openly available data on the Web, however, it is by far not the prevalent one. In this thesis we put a focus on an alternative, relatively untapped data source: datasets found on *Open Data portals* which are central points of access to large collections of data provided by a variety of governments, cities, and public institutions. These portals typically provide catalogs of datasets and their corresponding metadata, and provide access to a data resources in various (semi-)structured formats.

[1] `https://www.w3.org/2001/sw/`, last accessed 24/6/2019

[2] `https://www.w3.org/2013/data/`, last accessed 24/6/2019

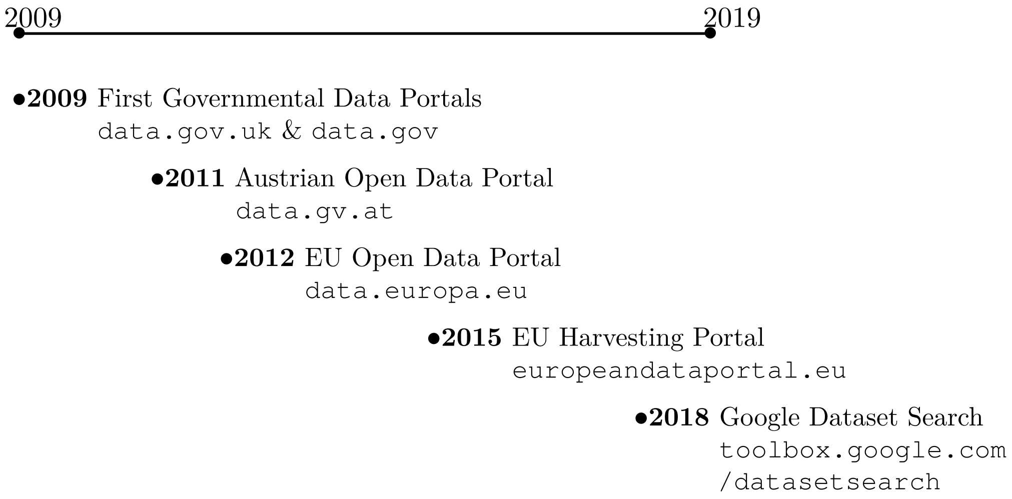

2009
2019
•2009 First Governmental Data Portals
data.gov.uk & data.gov
•2011 Austrian Open Data Portal
data.gv.at
•2012 EU Open Data Portal
data.europa.eu
•2015 EU Harvesting Portal
europeandataportal.eu
•2018 Google Dataset Search
toolbox.google.com
/datasetsearch

Figure 1.1: Some milestones in a decade of Open Government Data initiatives.

In the last decade *Open Data* as a paradigm and movement has gained a lot of popularity and support by governments in terms of improving transparency and enabling new business models: Governments and public institutions, but also private companies, provide open access to raw data with the goal to present accountable records [Attard et al., 2015], for instance in terms of statistical data, but also in fulfillment of regulatory requirements such as, e.g., the EU's INSPIRE directive.[3] The idea to provide raw data, instead of only human-readable reports and documents, is mainly driven by providing direct, machine-processable access to the data, and enable broad and arbitrary (through open licences) reuse of such data [Gurstein, 2011]. The growing number of Open Data catalogs is interesting also, particularly from an academic perspective, constituting a rich lode of freely available real world data, open for research experiments.

Also, the Austrian government continues to follow this trend. The governmental program of 2017 stated the following goal – prominently placed under the slogan "Digitaler Standort Österreich" (p. 19): "Stärkung und Förderung des Open-Data-Prinzips durch Veröffentlichung von behördlichen Daten, soweit nicht andere Rechtsprinzipien (Privatheit) dem entgegenstehen."[4] which translates as "strengthening and support of the Open Data principle by publishing governmental datasets, provided that doing so is not contrary to other legal principles (privacy)."

Over this decade, as a direct result of the increased momentum within the Open Data movement, more data is made available online and the expectation rises that people can use and exploit this data in innovative ways and generate added value out of it. We

[3] https://inspire.ec.europa.eu/, last accessed 24/6/2019

[4] https://www.dieneuevolkspartei.at/download/Regierungsprogramm.pdf, last accessed 24/6/2019

could identify many areas where Open Data is used and valuable, e.g., by governments to increase transparency and democratic control, or by private companies to encourage innovative use of their data. Having said that, it is impossible to predict how, when and where value can be created in the future: innovations enabled by freely available data can come from any unforeseen place or use case.

Our goal is to support and strengthen this paradigm; to this end we draw upon years of research in the Semantic Web field, which provides the technologies to perform integration and semantic enrichment of datasets published on the Web, ideally more efficiently and in an automated way. Eventually, we want to actively contribute to lift the growing amounts of Open Data catalogs to the Web of (Linked) Data – and demonstrate how both, portal providers and consumers can benefit from the approaches developed in the course of this thesis.

1.1 Research Questions & Hypotheses

While we already mentioned the motivations behind publishing Open Data, such as the increased transparency and democratic accountability, there are a number of potential barriers to its adoption, that can create such a high threshold that "the data is still private in practice" [Janssen et al., 2012]. In particular, the quality of information and data is listed as a key barrier in the literature when it comes to using Open Data [Janssen et al., 2012, Beno et al., 2017]. However, there is no quantitative, large-scale, study of the actual quality issues, which is required to derive concrete quality improvement approaches. We will present such a study, and a system to continuously monitor and profile the quality of Open Data. As we will see, this study shows a landscape of low quality and heterogeneous metadata, a limited use of common vocabularies, and insufficient descriptions of (tabular) datasets. To foster more homogeneous and richer dataset descriptions, we will give guidelines on how we use and extend existing vocabularies, how to automatically enrich the datasets, and how to expose the resulting descriptions as Linked Data.

Homogenised and enriched descriptions are certainly a step towards a better usability of Open Data; the natural next barrier that we want to tackle is the interlinking and integration of external knowledge: as we will show, we observe a risk of Open Data portals being isolated "data silos", due to missing integration of existing Knowledge Graphs. Our approach will focus on the semantic labelling of certain, specific, dimensions – we will describe how to add temporal and geospatial annotations – which in fact have been confirmed as the top-two query types on Open Data portals [Kacprzak et al., 2019]. Such annotations rely on mappings of textual information to classes, properties, or instances, in Knowledge Graphs in order to link – and eventually transform – tabular data into RDF. However, as we will illustrate, Open Data typically also contains a large portion of numerical columns and/or non-textual headers; therefore solutions that solely focus on textual "cues" are only partially applicable for mapping such data sources.[5]

[5] In some parts of this thesis we particularly focus on *tabular data* which is currently the predominant format on Open Data portals.

To sum up, the guiding idea of this thesis is to first assess the current state of Open Data, identify quality issues, and then use Semantic Web technologies to improve, enrich and link existing 3-star[6] Open Data – lift it to the Web of Data.

By doing so, we want to demonstrate how we can tackle existing Open Data challenges such as the problems of findability and search functionalities [Kacprzak et al., 2019]. It is important to note, however, that we do not want to force Open Data publishers into semantic technologies by all means; and that we do not demand all Open Data published as Linked Data: consistent with James A. Hendler's hypothesis "*a little semantics goes a long way*",[7] this work intends to increase the value of Open Data by *automatically* enriching and integrating resources using Semantic Web technologies.

We showcase solutions and services, we publish mapped, enriched and integrated meta-information of openly available resources, and eventually, we hope that portal providers and users – but also the Semantic Web community – benefit from our work and output.

The research questions of this thesis can be stated as follows:

Q1 *How is the current state of published Open Data in terms of quality? How can we use Semantic Web technologies for the semantic enrichment and integration of Open Data, and what are the limitations and upcoming challenges?*

There is an ongoing trend of publishing Open Data, however, the quality of the data and its description has a non-negligible impact on the reputation of the (governmental) organization publishing the data, but also on decision-making and business revenues that can be generated from open data.

In order to leverage Semantic Web technologies to increase the quality and usability of Open Data, we first have to assess and profile the current state. Existing work on semantic enrichment of tabular data mainly assume Web tables as a source and domain; our experiments will show distinctive characteristics of tabular open data and web tables so that existing techniques may not be directly applicable.

Q2 *How to describe and publish datasets, and (potentially enriched/improved) metadata, as Linked Data in an homogenised way?*

There are ongoing efforts on establishing best practices and standards (such as common vocabularies for metadata) for publishing datasets on the Web. However, the current Open Data landscape is still very heterogeneous: we face low quality and heterogeneous metadata across portals and catalogs.

Our research goal is to describe ways of publishing datasets, by using and extending standardised vocabularies and interfaces, in order to enable further integration, enrichment and homogenised data access.

[6] According to the 5-star deployment scheme [Berners-Lee, 2006], cf. Section 2.3.

[7] `http://www.cs.rpi.edu/~hendler/LittleSemanticsWeb.html`

Q3 *How to extract spatial, and temporal information from the data sources and, orthogonally, how to assign semantic labels (e.g. spatial or temporal contexts such as locations, events, etc.) to (combinations of) values in rows of such tables?*

Most datasets found in Open Data are regional/national census-based and therefore organised by spatio-temporal scopes. They typically provide data for certain regions, and they are valid for a certain time period, however, portals lack fine-grained and standardised descriptions along the spatial and temporal dimension. Our goal is to develop a semantic labelling algorithm to cover exactly these two prevalent dimensions.

Q4 *How to assign semantic labels to columns lacking textual information?*

To clarify terminology, in the course of this thesis by *semantic labelling* we mean the process of *assigning tables, rows, and columns of tabular data to entities, classes and properties in a Knowledge Graph.* While for Q2 and Q3 we presume that there is some textual information in and about the datasets available, here we research how to understand data that predominantly is non-human-readable, such as numerical columns in tables. A semantic labelling of numerical values based on descriptive features and the distribution of the values is currently not part of existing labelling approaches for tables.

To address the above research questions our overall goal is to test the following main hypothesis:

> **Given a corpus of (tabular) Open Data resources, the metadata descriptions can be enriched, and therefore the data quality can be increased, by semantically analyzing Open Data CSVs, assigning semantic labels, and integrating the extracted knowledge into a Knowledge Graph.**

This implicitly includes testing the following sub-hypotheses (which relate to research questions Q1-Q4):

H1 Open Data quality and characteristics can be assessed at scale and in an automated fashion; these are different from web tables and the knowledge covered in existing Knowledge Graphs.

H2 By reusing and extending standardised W3C vocabularies and interfaces, the homogenised and enriched/improved metadata descriptions can be published as Linked Data and enable further integration, e.g., in major search engines.

H3 We can extract spatial information (e.g., geolocations and regions), and temporal information (e.g., specific time periods) from open datasets using domain-specific heuristics and existing Semantic Web technologies.

Expecting full mappings is unrealistic for many CSVs due to the lack of structure; however, finding partial mappings of columns will already allow the extraction of a richer description of the datasets.

H4 Based on preliminary results of H1, we know that Open Data tables contain many numerical columns with non human-readable headers. Numerical tabular data sources can be labeled based on the distribution of their values using a background Knowledge Graph – or, at least, the distribution of the values of their columns and rows can be used to get hints on such labels.

1.2 Contributions and Thesis Structure

We now introduce the contributions of this thesis to the respective sub-hypotheses H1-H4:

- Contribution to **H1**: A major challenge for data – also often related to its provenance – is quality; on the one hand, the re-use of poor quality data is obviously not advisable, but on the other hand different applications might have different demands/definitions of quality.

  We introduce 18 concrete, Open Data-specific, *quality metrics along five dimensions* and use these metrics to measure and profile existing (governmental) data portals. Our *continuous assessment of 261 data portals* and around 1.1 million datasets is – to the best of our knowledge – the most comprehensive, quantitative, study on the quality of Open Data. The findings of this assessment provide insights in the use of metadata descriptions, file formats, licenses, etc. and therefore help us to select suitable methods for (meta)data enrichment and improvement, and semantic labelling of the datasets.

- Contribution to **H2**: We study how to *enrich and publish datasets using standard vocabularies* such as the W3C's DCAT vocabulary [Maali and Erickson, 2014], Schema.org, the Data Quality vocabulary [Debattista et al., 2016], the PROV Ontology [Lebo et al., 2013], and the CSV on the Web vocabulary [Pollock et al., 2015]. In the course of H1 we collect the data and metadata descriptions of the monitored portals, which we re-expose for querying and crawling, using these vocabularies. Also, we discuss how to automatically enrich the existing descriptions by quality measurements and by standardised metadata for CSVs (comma-separated-values files).

- Contribution to **H3**: We detail the construction of a *hierarchical Knowledge Graph of geo- and temporal entities* from existing Knowledge Graphs, and links between them, and present an algorithm for linking open datasets (both on a dataset- and on a record-level) to this Knowledge Graph. To illustrate utility of the approach we implement a prototype search interface where the indexed and annotated datasets and metadata can be queried.

- Contribution to **H4**: As a result of H1, we show that indeed a major part of the datasets published in Open Data portals are tables containing pre-dominantly numerical columns with missing, or non human-readable headers. We tackle this problem by developing an algorithm for identifying the most likely property or classes for instances described by a bag of numerical values. To the best of our knowledge, this is the first unsupervised approach for *semantic labelling of numerical value sets*.

The remainder of this thesis is structured as follows:

Chapter 2 provides background information on sources of openly available data, popular data formats on the Web, Semantic Web standards, as well as existing Knowledge Graphs;

Chapter 3 presents our approach and findings on building a large-scale quality assessment and monitoring framework for Open Data on the Web;

Chapter 4 details how to lift existing portals to the Web of Data by using and extending standardised vocabularies, enriched dataset descriptions, and Linked Data access points and interfaces;

Chapter 5 describes our approaches for automated semantic enrichment of open datasets: first, focusing on the numerical contents, and then on the temporal and spatial cues in the data and metadata;

Chapter 6 summarises this work, and critically reviews the research questions. We conclude the thesis with a discussion of interesting future research directions.

1.3 Impact

Parts of the work presented herein have been published in international workshops, conferences, journal articles, as well as a book chapter, which we now briefly introduce in chronological order.

- We presented our first results on the analysis of the quality and evolution of Open Data portals at the International Conference on Open and Big Data [Umbrich et al., 2015] where it was **awarded as best paper**.

  A substantially extended and improved version of the above paper was published in the Journal of Data and Information Quality [Neumaier et al., 2016b]; this work is included in Chapter 3.

- We presented a large-scale profiling study of Open Data CSV files at the International Conference on Open and Big Data [Mitlöhner et al., 2016]; the findings can be found in Section 3.6.

- We presented our approach on measuring the “freshness” of resources found on Open Data portals at the International Conference on Open and Big Data [Neumaier and Umbrich, 2016]; the results of this paper are discussed in Section 3.2.
- We presented our approach to find and rank candidates of semantic labels and context descriptions for labelling numeric columns of Open Data tables at the International Semantic Web Conference [Neumaier et al., 2016a] where it was **nominated for best student paper award** – this work is presented in Section 5.1.
- We discuss and introduce challenges of integrating openly available Web data in a book chapter of the Reasoning Web Summer School 2017 lecture notes [Neumaier et al., 2017a], which serves as a starting point for this thesis and as a basis for Chapter 2.
- We presented our work on mapping, linking, and re-exposing metadata from Open Data portals at the Workshop on Linked Data on the Web, co-located with the International World Wide Web Conference [Neumaier et al., 2017b]; this approach is presented in Chapter 4.
- We presented a large-scale comparison of metadata quality across Open Data portals (using the Analytic Hierarchy Process) in the Government Information Quarterly journal [Kubler et al., 2018]. This work is an outcome of a collaboration with the University of Luxembourg and parts of the findings can be found in Section 3.5.
- We presented our approach to add geo-semantic labels to datasets from Open Data portals at the International Conference on Semantic Systems (SEMANTiCS) [Neumaier et al., 2018], which serves as groundwork for the work in Section 5.2.

  We published an extension of the above paper – we extended the approach by temporal annotation, we annotated a larger corpus of datasets, and enabled structured and full-text query – in the Journal of Web Semantics [Neumaier and Polleres, 2019].
- We participated in the seminar “Knowledge Graphs: New Directions for Knowledge Representation on the Semantic Web” at Schloss Dagstuhl, and contributed to the report [Bonatti et al., 2019]; some compiled results of the seminar can be found in Chapter 2.

Besides the above publications, we have also been involved in other projects and community works which have provided much inspiration for this thesis. Most of the research presented herein was conducted in the course of two projects, funded by the Austrian Research Promotion Agency: In the ADEQUATe project[8] we focused on research and development of automated and community-driven data quality improvement techniques. As a result of this project we integrated a quality assessment and improvement tool into the

[8] https://adequate.at/

Austrian data portal data.gv.at (in collaboration with the respective administrative office): It reports back any data quality issues to the publishers and provides improved metadata descriptions. Complementary to ADEQUATe, in the CommuniData project[9] we developed an Urban Participation Platform for a local Viennese district, combining online discussions, (local) dataset search and dataset visualizations.

In the course of our work on metadata standards for datasets we participated in the W3C working groups on CSV on the Web[10] and on Dataset Exchange (DXWG).[11] The source code of our quality framework, experiments, and prototypes is available as open source on Github[12], and the running software and re-published corpus of datasets is accessible at data.wu.ac.at, from where it gets harvested and integrated in the Google Dataset Search Engine [Brickley et al., 2019].

[9] https://www.communidata.at/

[10] https://www.w3.org/2013/csvw/wiki/Main_Page

[11] https://www.w3.org/2017/dxwg/wiki/Main_Page

[12] https://github.com/sebneu and https://github.com/ADEQUATeDQ

CHAPTER 2

Towards a Web of Open Data

This chapter may be viewed as a background for the topics covered in this thesis as well as – hopefully – a guide through challenges that arise when using (open) data from the Web. Over the past view years we have been involved in several projects and publications around the topic of Open Data integration, monitoring, and processing. The foundations and basic challenges we faced in all these projects are largely overlapping and therefore we decided to present them in the present chapter:

(i) **Where to find Open Data?** (Section 2.1) There are various sources of openly available data; we will discuss the main starting points. Later on this thesis particularly focuses on so-called Open Data portals, that is, data catalogs, typically allowing API access and hosting dataset descriptions and links to actual data resources.

(ii) **“Low-level” data heterogeneity** (Section 2.2) As we will see, most of the structured data provided as Open Data is not readily available as RDF or Linked Data – the preferred formats for semantic data access. Different formats are much more prevalent, plus encoding issues make it difficult to access those datasets. We discuss popular, well-known, formats on the Web to give an idea of the challenges in terms of heterogeneity.

(iii) **The Semantic Web to the rescue?** (Section 2.3) The vision of a “Web of Data” is based on agreed-upon data publishing methods, and common schemata and vocabularies. After revisiting the Linked Data principles we discuss how to use the DCAT vocabulary to describe datasets on the Web.

(iv) **Licenses and Provenance** (Section 2.4) Not all *Open Data* is really completely open, since most data on the Web is attached to different licences, terms and conditions, so we will discuss how and whether these licenses can be interpreted by machines, or, respectively how the provenance of different integrated data sources can be tracked.

(v) **Knowledge Graphs – A new hope** (Section 2.5) The phrase “Knowledge Graph” has recently become popular in both industry and academia: It subsums different approaches of curating, extracting and representing knowledge (partially using Semantic Web technologies). We discuss different definition attempts and give a brief history of important Knowledge Graphs at the end of this chapter.

2.1 Where to find Web Data?

In the following, we want to introduce sources of openly available data on the Web, and in particular, show the role of Open Data portals, which are the focus of our work in the later chapters. If we look for such sources that are widely discussed in the literature, we mainly identify four starting points, which are partially overlapping:

- User-created open databases
- The Linked Open Data “Cloud”
- Webcrawls
- Open Data portals

The union of all these sources, as they exist now, do not (yet) form a Web of Data as such, for which Linked Data (cf. Section 2.3) provides a (still small) crystallisation point.

User-created open databases: Through efforts such as Wikipedia large amounts of data and data-bases have been co-created by user communities distributed around the globe; the most important ones being listed as follows:

- DBpedia [Lehmann et al., 2015c] is an information extraction effort that has created one of the biggest and most important cross-domain dataset in RDF [Brickley and Guha, 2014] in the focal point of the so called Linked Open Data (LOD) cloud [Auer et al., 2007]. At its core is a set of declarative mappings extracting data from Wikipedia *infoboxes* and tables into RDF. It is accessible through dumps as well as through an open query interface supporting the SPARQL [Harris and Seaborne, 2013] query language. DBpedia can therefore be well called one of the cornerstones of Semantic Web and Linked Data research being the subject and center of a large number of research papers over the past few years. Reported numbers vary as DBpedia is modular and steadily growing with Wikipedia, e.g., in 2015 DBpedia contained overall more than 3B RDF Statements,[1] whereof the English DBpedia

[1] `http://wiki.dbpedia.org/about/facts-figures`, last accessed 12/02/2019

contributed 837M statements (RDF triples). Those 837M RDF triples alone amount to 4.7GB when stored in the compressed RDF format HDT [Fernández et al., 2013].[2] However, as we will see there are many, indeed far bigger other openly accessible data sources, that yet remain to be integrated, which are rather in the focus of the present thesis.

- Wikidata [Vrandecic and Krötzsch, 2014] a similar, but conceptually different effort has been started in 2012 to bring order into data items in Wikipedia. Instead of extracting data from semi-structured Wiki-pages, Wikidata is a database for data observations with fixed properties and data types, mainly with the idea to avoid extraction errors and provide means to record provenance directly with the data, with likewise 100s of millions of facts in the meantime: Exact numbers are hard to give, but [Tanon et al., 2016] report some statistics of the status of 2015, when Freebase was included into Wikidata; we note that counting RDF triples[3] is only partially useful, since the data representation of Wikidata is not directly comparable with the one from DBpedia [Hernández et al., 2015, Hernández et al., 2016]. Both, DBpedia and Wikidata, can be characterized as centralized, open Knowledge Graphs (cf. Section 2.5), intended as public resources.

- OpenStreetMap as another example of an openly available data base that has largely been created by users contains a vast amount of geographic features to obtain an openly available and re-usable map; with currently over 1028.9 GB (uncompressed) data in OSM's native XML format (and still 43.6GB compressed).[4]

The Linked Open Data "Cloud": The LOD cloud is a manually curated collection of datasets that are published on the Web openly, adhering to the so-called Linked Data principles, discussed in Section 2.3 below. The latest iteration of the LOD Cloud [Abele et al., 2017] contains – with DBpedia in its center – hundreds of datasets with equal or even larger sizes than DBpedia, documenting a significant growth of Linked Data over the past years. Still, while often in the Semantic Web literature the LOD cloud and the "Web of Data" are implicitly equated, there is a lot of structured data available on the Web (a) either, while using RDF, not being linked to other datasets, or (b) provided in other, popular formats than RDF.

Web Crawls: Crawling the Web is the only way to actually find and discover structured Web Data, which is both resource intensive and challenging in terms of respecting politeness rules when crawling. However, some Web crawls have been made openly available, such as the Common Crawl corpus which contains "petabytes of data collected over the last 7 years".[5] Indeed, the project has already been used to collect and analyse

[2] `http://www.rdfhdt.org/datasets/`, last accessed 12/02/2019

[3] Executing the SPARQL query `SELECT (count(*) as ?C ) WHERE {?S ?P ?O }` on `https://query.wikidata.org/` gives 7.1B triples, last accessed 12/02/2019.

[4] `http://wiki.openstreetmap.org/wiki/Planet.osm`, last accessed 12/02/2019

[5] `http://commoncrawl.org/`, last accessed 12/02/2019

the availability (and quality) of structured data on the Web, e.g., in the Web Data Commons Project [Meusel et al., 2014, Meusel et al., 2016].

Data Portals & Catalogs: Data portals are collections, or catalogs, that index metadata and link to actual data resources. Data portals as central points of data access have become popular over the past few years through various Open Government Data Initiatives, but also in the private sector. Apart from all the other sources mentioned so far, most of the data published openly is indexed in some kind of Open Data portal. We therefore will discuss these portals in the following Section 2.1.1 in more detail.

2.1.1 Open Data Portals

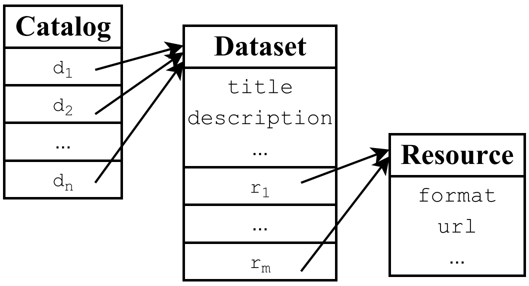


Figure 2.1: High-level structure of a Data Portal.

Most of the current “open” data is published in so called Open Data portals: data catalogues similar to digital libraries (cf. Figure 2.1). In such catalogues, a *dataset* aggregates a group of data files (referred to as *resources* or distributions) which are available for access or download in one or more formats (e.g., CSV, PDF, Microsoft Excel, etc.). Associated to a dataset is *metadata*, i.e. basic descriptive information in structured format, about these resources, for instance, about the authorship, provenance or licensing of the dataset.

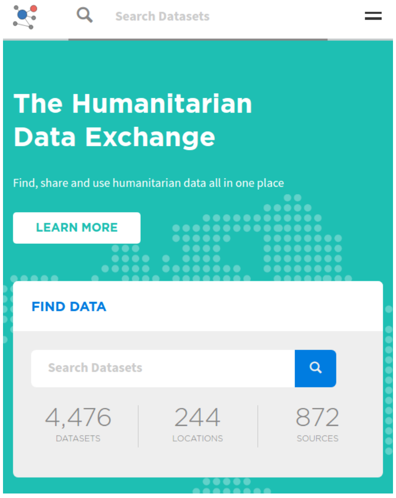
Most of these portals rely on existing software frameworks, such as CKAN[6] or Socrata.[7] CKAN is the most prominent portal software framework for publishing Open Data and is used by several governmental portals including the UK’s `data.gov.uk` and the US’s `data.gov`. While the CKAN software is increasingly popular among cities, governments and private data providers worldwide, including governmental portals of countries in Europe, South and North American and the Middle East, the customers of Socrata can be found mainly in the US. Generally, these portal frameworks provide ecosystems to describe, publish and consume datasets. The frameworks typically consist of a content management system, some query and search functionality, as well as RESTful APIs to allow agents to interact with the platform and automatically retrieve the metadata and data from the portals. Also the metadata can be retrieved in a structured format via the

[6] `https://ckan.org/`, last accessed 12/02/2019

[7] `https://socrata.com/`, last accessed 12/02/2019

Table 2.1: Data portals of the G7 countries plus Australia, Russia, and the EU as in February 2019.

domain	Country	Software	$\lvert datasets \rvert$
data.gov	US	CKAN	170.7k
open.canada.ca	Canada	CKAN	79.1k
data.gov.uk	UK	CKAN	45.1k
www.data.gouv.fr	France	*other*	34.2k
data.go.jp	Japan	CKAN	21k
dati.gov.it	Italy	CKAN	20.4k
govdata.de	Germany	CKAN	19.8k
data.gov.au	Australia	CKAN	73k
opengovdata.ru	Russia	CKAN	30.3k
data.europa.eu	EU	CKAN	13.2k



Search Datasets

The Humanitarian Data Exchange

Find, share and use humanitarian data all in one place

LEARN MORE

FIND DATA

Search Datasets

4,476 DATASETS

244 LOCATIONS

872 SOURCES

```
{
  "name":"amounts-paid-by-refugees-...",
  "title": "Amounts paid by refugees...",
  "license": "Creative Commons Attribution",
  "tags": [
    "europe",
    "mediterranean",
    "refugee"
  ],
  "resources": [
    {
      "format": "CSV",
      "name": "The Money Trail - South - Prices",
      "created": "2015-10-28T21:20:40.006453",
      "url": "https://docs.google.com/...",
    }
  ],
  ...
}
```

Figure 2.2: Example dataset description from the Humanitarian Data Exchange portal.

API (e.g., as JSON data); however, the metadata schemata are potentially heterogeneous wrt. the underlying software framework.

To illustrate the ongoing data-publishing trend among governments we list in Table 2.1 the portals of the G7 countries plus Australia, Russia, and the EU, together with the underlying software framework and number of published datasets (as in February 2019).

As an example, we have a look at the Humanitarian Data Exchange[8] (see Figure 2.2), a portal by the United Nations. It aggregates and publishes data about the context in which a humanitarian crisis is occurring, e.g., damage assessments and geospatial

[8] https://data.humdata.org/, last accessed 12/02/2019

Table 2.2: The tabular content of the dataset in Figure 2.2

Route	Period	Ref crossing	Total in EUR 2014
Central Med	2010-2015	285,700	3,643,000,000
East Borders	2010-2015	5,217	72,000,000
East Med Land	2010-2015	108,089	1,751,000,000
East Med Sea	2010-2015	61,922	1,053,000,000
West African	2010-2015	1,040	4,000,000
West Balkans	2010-2015	74,347	1,589,000,000
West Med	2010-2015	29,487	251,000,000

data, and data about the people affected by the crisis. The datasets on this portal are described using several metadata fields; the metadata description can be retrieved in JSON format using the Web API of the data portal (cf. Figure 2.2). The metadata description of a dataset at the Humanitarian Data Exchange provides download links for the actual content. For instance, the particular dataset description in Figure 2.2 – a dataset reporting the amounts paid by refugees to facilitate their movement to Europe – holds a URL which refers to a table (a CSV file) containing the corresponding data, displayed in Table 2.2.

2.2 Data Formats on the Web

When we discussed different sources of data on the Web, we already emphasized that – despite being subject of a lot of research – RDF and Linked Data are not necessary the prevalent formats for published data on the Web. An analysis of the datasets systematically catalogued in Open Data portals in Chapter 3 will confirm this; likewise, we will discuss *metadata* formats on these portals in the next chapter.

In the following, however, we introduce some of the popular, well known, data formats on the Web and categorize them by their structure, namely, graph-based, tree-shaped, and tabular formats. Although this might be a very low level and basic discourse, we want to give the reader an idea of the potential heterogeneity and diversity we are facing when crawling Open Data on the Web.

2.2.1 Graph-based formats

RDF, W3C recommendation since 2004 [Klyne and Carroll, 2004] and "refurbished" in 2014 [Cyganiak et al., 2014, Brickley and Guha, 2014], was originally conceived as a metadata model language for describing resources on the web. It evolved (also through deployment) to a universal model and format to describe arbitrary relations between resources identified, typically, by URIs, such that they can be read and understood by machines.

RDF itself consists of statements in the form of *subject*, *predicate*, *object* triples. RDF triples can be displayed as graphs where the subjects and objects are nodes and the

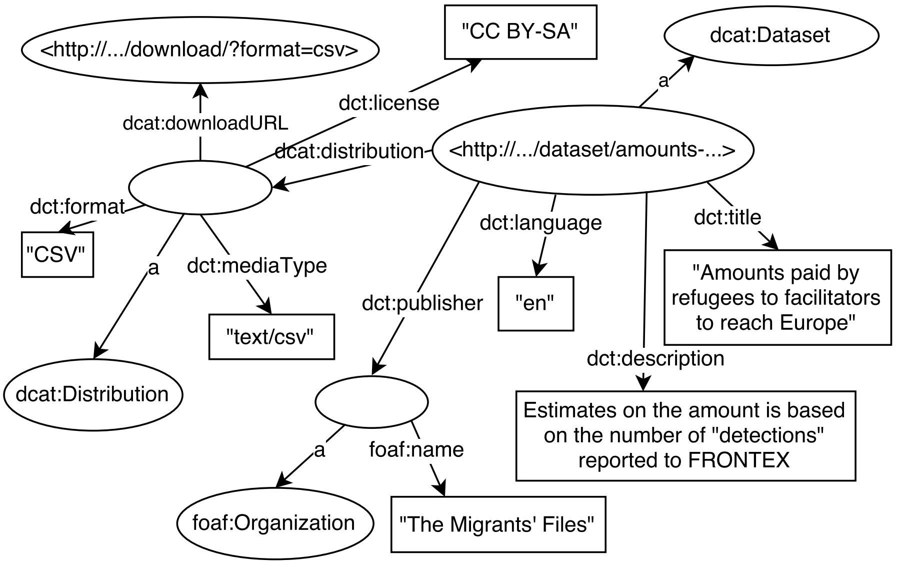


Figure 2.3: RDF representation of the JSON metadata from Figure 2.2

predicates are directed edges. RDF uses vocabularies to define the set of elements that can be used in an application. Vocabularies are similar to schemas for RDF datasets and can also define the domain and range of predicates. The example graph in Figure 2.3 represents the metadata description of the dataset in Figure 2.2 as RDF. The semantic mapping of the information from the original metadata document to the RDF representation, using standard vocabularies, will be part of Section 3.2.1.

There exist several formats to serialize RDF data. Most prominent is RDF/XML, the XML serialization first introduced in the course of 1999 W3C specification of the RDF data model, but there are also a more readable/concise textual serialization formats such as the line-based N-Triples [Carothers and Seaborne, 2014] and the "Terse RDF Language" TURTLE [Beckett et al., 2014] syntax. An example RDF file in TURTLE syntax can be found in Listing 2.1. More recent, in 2014, W3C released the first recommendation for JSON-LD [Sporny et al., 2014]. JSON-LD is an extension for the JSON format (see below) mostly allowing to specify namespaces for identifiers and support of URIs (supporting Linked Data principles natively in JSON) which allows the serialization of RDF as JSON, or vice versa, the transformation of JSON as RDF: conventional JSON parser and databases can be used; users of JSON-LD which are mainly interested in conventional JSON, are not required to understand RDF and do not have to use the Linked Data additions.

2.2.2 Tree-shaped formats

The JSON file format [Bray, 2014] is a so-called semi-structured file format, i.e. where documents are loosely structured without a fixed schema (as for example data in relational databases) as attribute–value pairs where values can be primitive (Strings, numbers, Booleans), arrays (sequences of values enclosed in square brackets [,]), or nested JSON objects (enclosed in curly braces {, }), thus – essentially – providing a serialization format for tree-shaped, nested structures. For an example for JSON we refer to Figure 2.2.

Initially, the JSON format was mainly intended to transmit data between servers and web applications, supported by web services and APIs. In the context of Open Data we often find JSON as a format to describe metadata but also to publish the actual data: also raw tabular data can easily be transformed into semi-structured and tree-based formats like JSON[9] and, therefore, is often used as alternative representation to access the data. On the other hand, JSON is the de facto standard for retrieving metadata from Open Data portals.

XML. For the sake of completeness, due to its long history, and also due to its still striking prevalence as a data exchange format of choice, we shall also mention some observations on XML. This prevalence is not really surprising since many industry standards and tools export and deliver XML, which is then used as the output for many legacy applications or still popular for many Web APIs, e.g., in the area of geographical information systems (e.g. KML,[10] GML,[11] WFS,[12] etc.). Likewise, XML has a large number of associated standards around it such as query, navigation, transformation and schema languages like XQuery,[13] XPath,[14] XSLT[15], and XML Schema[16] which are still actively developed, supported by semi-structured database systems, and other tools. XML by itself has been subject to extensive research, for example in the fields of data exchange [Arenas et al., 2014, Part III] or query languages [Bailey et al., 2005].

Particularly, in the context of the Semantic Web, there have also been proposals to combine XQuery with SPARQL, cf. for instance [Bischof et al., 2012, Dell'Aglio et al., 2014] and references therein. The issue of interoperability between RDF and XML indeed is further discussed within the W3C in their recently started "RDF and XML Interoperability Community Group",[17] see also [Borriello et al., 2016] for a summary. So, whereas JSON has probably better support in terms of developer-friendliness and recent uptake particularly through Web APIs, there is still a strong community with well-established standards behind XML technologies. For instance, schema languages or query languages for JSON exist as proposals, but their formal underpinning is still under discussion, cf. e.g. [Pezoa et al., 2016, Bourhis et al., 2017]. Another approach would be to adopt, reuse and extend XML technologies to work on JSON itself, as for instance proposed in [Dell'Aglio et al., 2014]. On an abstract level, there is not much to

[9] E.g., see Converter Tools on `https://project-open-data.cio.gov/`, last accessed 01/07/2019

[10] `https://developers.google.com/kml/documentation/`, last accessed 13/02/2019

[11] `http://www.opengeospatial.org/standards/gml`, last accessed 13/02/2019

[12] `http://www.opengeospatial.org/standards/wfs`, last accessed 13/02/2019

[13] `https://www.w3.org/TR/xquery-30/`, last accessed 13/02/2019

[14] `https://www.w3.org/TR/xpath-30/`, last accessed 13/02/2019

[15] `https://www.w3.org/TR/xslt-30/`, last accessed 13/02/2019

[16] `https://www.w3.org/XML/Schema`, last accessed 13/02/2019

[17] `https://www.w3.org/community/rax/`, last accessed 13/02/2019

argue about JSON and XML just being two syntactic variants for serializing arbitrary, tree-shaped data.

2.2.3 Tabular data formats

Last but not least, potentially driven also by the fact that the vast majority of Open Data on the Web originates from relational databases or simply from spreadsheets, a large part of the Web of Open Data consists of tabular data. This is illustrated by the fact that two of the most prominent formats for publishing Open Data[18] cover tabular data: **CSV** and **XLS**. Note particularly that both of these formats are present on more Open Data portals than for instance XML. We will discuss the file format distribution further in the next chapter.

While XLS (the export format of Microsoft Excel) is obviously a proprietary open format, CSV (comma-separated values) is a simple, open format with a standard specification allowing to serialize arbitrary tables as text (RFC4180) [Shafranovich, 2005]. However, as we will show in Section 3.6, compliance with this standard across published CSVs is not consistent: in a corpus of 200K CSV resources with a total file size of 413GB we found out that out of the resources in Open Data portals labelled as a tabular only 50% can be considered CSV files. In this work we also investigated different use of delimiters, the availability of (multiple) header rows or cases where single CSV files actually contain multiple tables as common problems.

Last, but not least, as opposed to tabular data in relational databases, which typically adhere to a fixed schema and constraints, these constraints, datatype information and other schema information is typically lost when being exported and re-published as CSVs. This loss can be compensated partially by adding this information as additional metadata to the published tables; one particular format for such kind of metadata has been recently standardized by the W3C [Pollock et al., 2015]. For more details on the importance of metadata we again refer to Chapter 3.

2.2.4 Data Formats – Summary

Overall, while data formats are often only considered syntactic sugar, one should not underestimate the issues about conversions, scripts parsing errors, stability of tools, etc. where often significant amounts of work incurs. While any data can be converted/represented in principle into a CSV, XML, or RDF serialization, one should keep in mind that a canonical, “dumb” serialization in RDF by itself, does not “add” any “semantics”.

For instance, a naive RDF conversion (in Turtle syntax) of the CSV in Table 2.2 could look as follows in Listing 2.1, but would obviously not make the data more “machine-readbable” or easier to process.

[18] See Table 3.10 in Section 3.4.5 for a discussion of popular file formats in Open Data portals.

Listing 2.1: Naive conversion of tabular data into RDF

```
@prefix : <http://www.example.org/> .

:c1 rdfs:label "Route".
:c2 rdfs:label "Period".
:c3 rdfs:label "Ref_crossing".
:c4 rdfs:label "Total in EUR 2014".

[:c1 "Central Med"; :c2 "2010-2015", :c3 "285,700"; :c4 "3,643,000,000"].
[:c1 "East Borders"; :c2 "2010-2015"; :c3 "5,217"; :c4 "72,000,000" ].
[:c1 "East Med Land" ; :c2 "2010-2015"; :c3 "108,089" ; :c4 "1,751,000,000"].
[:c1 "East Med Sea"; :c2 "2010-2015" ; :c3 "61,922"; :c4"1,053,000,000"].
[:c1 "West African"; :c2 "2010-2015"; :c3 "1,040"; :c4 "4,000,000"].
[:c1 "West Balkans"; :c2 "2010-2015"; :c3 "74,347"; :c4 "1,589,000,000"].
[:c1 "West Med"; :c2 "2010-2015"; :c3 "29,487"; :c4 "251,000,000"].
```

We would leave coming up with a likewise naive (and probably useless) conversion to XML or JSON to the reader: The real intelligence in mapping such data lies in finding suitable ontologies to describe the properties representing columns $c1$ to $c4$, recognizing the datatypes of the column values, linking names such as “East Med Sea” to actual entities occurring in other datasets and knowledge graphs, etc. Still, the data conversion, pre-processing and cleansing tasks in data processing should not be underestimated: In fact, in a 2017 survey by Kaggle, Google’s Data Science platform, “dirty data” leads as the most common and time-consuming problem for workers in the data science realm.[19]

Within the Semantic Web, or to be more precise, within the closed scope of Linked Data (cf. Section 2.3) this problem and the steps involved have been discussed in depth in the literature [Auer and Lehmann, 2010, Ngomo et al., 2014]. A partial instantiation of a platform which shall provide a cleansed and integrated version of the Web of Linked Data is presented by the LOD-Laundromat [Beek et al., 2016] project: here, the authors present a cleansed unified store of Linked Data as an experimental platform for the whole Web of Linked Data, mostly containing the all datasets of the current LOD cloud, are made available. Querying this platform efficiently and investigating the properties of this subset of the Web of Data is a subject of active ongoing research, despite only linked RDF data has been considered; however, building such a platform for the scale of arbitrary Open Data on the Web – or even only for the data accumulated in Open Data portals – would demand a solution at a much larger scale, handling more tedious cleansing, data format conversion and schema integration problems.

2.3 The Semantic Web & Linked Data

The so-called “Semantic Web” is based on the idea of Tim Berners-Lee [Berners-Lee, 1998] who envisioned an extension of the existing Web, which – additionally to the current human-readable “Web of Documents” – is a “Web of Data” and expresses “information in a machine processable form” [Berners-Lee, 1998]. The Resource Description Framework

[19] https://www.kaggle.com/surveys/2017, last accessed 13/02/2019

(RDF) serves as the core technology for the Semantic Web: In theory, it allows us to model and exchange information based on common schemata, assertions and rules. However, early RDF on the Web was mostly published as either large data dumps or in isolated documents, without links between different RDF sources.

To address this issue Berners-Lee introduced four "Linked Data" design principles [Berners-Lee, 2006] to increase the integration and linkage between published RDF resources:

LDP1: use URIs as names for things;

LDP2: use HTTP URIs so those names can be dereferenced;

LDP3: return useful – herein we assume RDF – information upon dereferencing of those URIs; and

LDP4: include links using externally dereferenceable URIs.[20]

The main idea of these principles is that the URIs used in the RDF data should be HTTP-dereferenceable, and ideally should link to other RDF sources, which in return provide additional information. To encourage publishers – in particular, government data publishers – to provide their data as Linked Data, in such a manner that it is reusable and integrable, a "5 Star Rating" [Berners-Lee, 2006] has been added to the Linked Data principles. In this rating, above all in order to be considered as "open", the dataset has to be published under an open license; we discuss the importance and technical details of licensing in Section 2.4. Subsequently, the scheme awards stars for the use of open formats, standards, and the linkage to other datasets. The original rating, paraphrased from Berners-Lee [Berners-Lee, 2006]:

☆	Available on the web (whatever format) *but with an open licence, to be Open Data*
☆☆	Available as machine-readable structured data (e.g. excel instead of image scan of a table)
☆☆☆	as (2) plus non-proprietary format (e.g. CSV instead of excel)
☆☆☆☆	All the above plus, Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff
☆☆☆☆☆	All the above, plus: Link your data to other people's data to provide context

In many aspects, the Semantic Web has not necessarily evolved as expected, and the biggest success stories so far do less depend on formal logics [Hitzler et al., 2014] than many may have expected, but more on the availability of data. A recent article by Bernstein et al. [Bernstein et al., 2016] takes a backwards look on the community and

[20] That is, within your published RDF graph, use HTTP URIs pointing to other dereferenceable documents, that possibly contain further RDF graphs.

summarizes successes of the Semantic Web such as the establishment of lightweight annotation vocabularies like Schema.org on Web pages – in the spirit of the Linked Data principles – or praising the uptake of large companies such as Google, Yahoo!, Microsoft, and Facebook who are developing large knowledge graphs (cf. Section 2.5), which however, so far mostly are closed.

2.3.1 A Vocabulary for Data Catalogs

When introducing data portals and catalogs as sources of Open Data on the Web (Section 2.1.1) we already mentioned the problem of heterogeneous metadata schemata; different schemata and vocabularies prevent an integration and interlinkage of data as envisioned by the Semantic Web idea. Therefore, the W3C proposed the Data Catalog Vocabulary (DCAT) [Maali and Erickson, 2014] for describing datasets on the Web: DCAT is a W3C metadata recommendation defined in RDF and reuses the Dublin Core Metadata vocabulary.[21]

On a high level the DCAT model distinguishes between *catalogs*, *datasets*, and *distributions* (see Figure 2.4) and corresponds to the high level structure of Open Data portals, outlined in Section 2.1.1. A `dcat:Catalog` represents the catalog or portal and is a "curated collection of metadata about datasets" [Maali and Erickson, 2014]; the DCAT model suggests a set of Dublin Core properties to describe a catalog in detail, e.g. `dct:title`, and `dct:language`. The property `dcat:dataset` is used to add instances of the class `dcat:Dataset` to a catalog: A DCAT dataset is a collection of data, with a set of recommended properties (e.g., title, description, distribution, etc.), available for access or download. The additional class `dcat:CatalogRecord` is optional, and is only used if "a distinction is made between metadata about a dataset and metadata about the dataset's entry" [Maali and Erickson, 2014]. Eventually, the `dcat:Distribution` represents an accessible form of the dataset, e.g., a downloadable CSV file, or an API that provides the data. Listing 2.2 illustrates a concrete DCAT dataset and a corresponding DCAT distribution (example taken from [Maali and Erickson, 2014]).

[21] Dublin Core (DC) is a metadata standard that has been specified by the Dublin Core Metadata Initiative [Weibel et al., 1998]. It contains elements for describing resources that are used primarily for cataloging, archiving and indexing of documents (e.g., in archives, libraries).

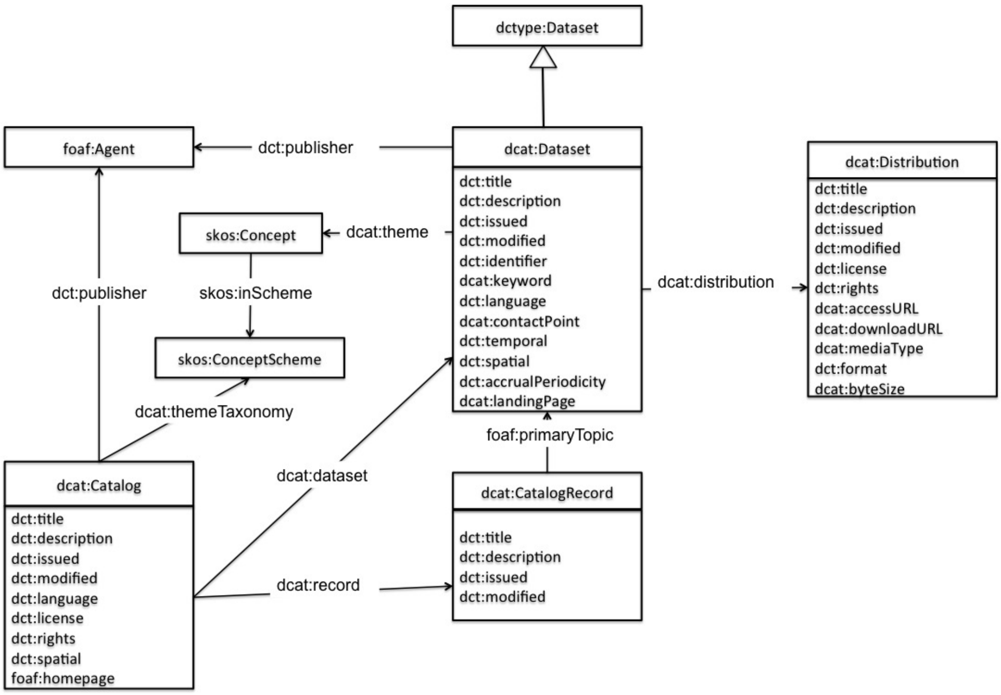


Figure 2.4: The DCAT classes and properties from [Maali and Erickson, 2014].

Listing 2.2: DCAT example

```
:dataset-001
  a dcat:Dataset ;
  dct:title "Imaginary dataset" ;
  dcat:keyword "accountability","transparency" ,"payments" ;
  dct:issued "2011-12-05"^^xsd:date ;
  dct:publisher :finance-ministry ;
  dcat:distribution :dataset-001-csv ;
  .

:dataset-001-csv
  a dcat:Distribution ;
  dcat:downloadURL <http://www.example.org/files/001.csv> ;
  dct:title "CSV distribution of imaginary dataset 001" ;
  dcat:mediaType "text/csv" ;
  .
```

Another example of a DCAT metadata description is given in Figure 2.3: The figure represents the metadata description of the dataset in Figure 2.2 mapped to the DCAT vocabulary. The process of mapping and homogenization of metadata descriptions from different sources will be discussed in Section 3.2.1.

The European Union identified the issue of insufficient description of public datasets and conducted several activities towards metadata standards across European portals: The DCAT application profile for data portals in Europe (DCAT-AP)[22] extends the DCAT core vocabulary and aims towards the integration of datasets from different European data portals. In its current version (1.2) it extends the existing DCAT schema by a set of additional properties. DCAT-AP allows to specify the version and the temporal scope of a dataset; further, it classifies certain predicates as "optional", "recommended" or "mandatory". For instance, in DCAT-AP it is mandatory for a `dcat:Distribution` to hold a `dcat:accessURL`. The European Data Portal[23] (launched in November 2015) which serves as a harvesting portal of 78 European data catalogs, uses and supports the DCAT-AP metadata profile.

2.4 Licensing and Provenance of Data

Publishing data on the Web is more than making it publicly accessible using a suitable representation. When it comes to consuming publicly accessible data, it is crucial for data consumers to be able to assess the trustworthiness of the data as well as being able to use it on a secure legal basis and to know where the data is coming from, or how it has been pre-processed. As such, if data is to be published on the Web, appropriate metadata (e.g., describing the data's provenance and licensing information) should be published alongside with it, thus making published data as self-descriptive as possible (cf. [Heath and Bizer, 2011]).

Current license definitions found in Open Data lack a machine-readable description that would allow automated compatibility checks of different licenses with Open Definition compliant data licenses (cf. Table 2.3). Later in this thesis, in Section 3.4.5, we analyse and discuss the use of licenses on Open Data portals in detail (cf. Table 3.10 for a listing of the most common licenses across the portals).

Table 2.3: Open Definition compliant data licenses [International, 2019]

License
Creative Commons Zero (CC0)
Creative Commons Attribution 4.0 (CC-BY-4.0)
Creative Commons Attribution Share-Alike 4.0 (CC-BY-SA-4.0)
Open Data Commons Attribution License (ODC-BY)
Open Data Commons Public Domain Dedication and Licence (ODC-PDDL)
Open Data Commons Open Database License (ODC-ODbL)

[22] `https://joinup.ec.europa.eu/solution/dcat-application-profile-data-portals-europe`, last accessed 02/12/2019

[23] `http://www.europeandataportal.eu/`, last accessed 02/12/2019

In order to circumvent these shortcomings, different RDF vocabularies have been introduced to formally describe licenses as well as provenance information of datasets, two of which (ODRL and PROV) we will briefly introduce in the next two subsections.

2.4.1 Making Licenses machine-readable

The Open Digital Rights Language (ODRL) [Iannella and Villata, 2017] is a comprehensive policy expression language (representable with a resp. RDF vocabulary) that has been demonstrated to be suitable for expressing fine-grained access restrictions, access policies, as well as licensing information for Linked Data as shown in [Cabrio et al., 2014, Steyskal and Polleres, 2014].

An *ODRL Policy* is composed of a set of *ODRL Rules* and an *ODRL Conflict Resolution Strategy*, which is used by the enforcement mechanism to ensure that when conflicts among rules occur, a system either grants access, denies access or generates an error in a non-ambiguous manner.

An *ODRL Rule* either permits or prohibits the execution of a certain action on an asset (e.g. the data requested by the data consumer). The scope of such rules can be further refined by explicitly specifying the party/parties that the rule applies to (e.g. Alice is allowed to access some dataset), using constraints (e.g. access is allowed until a certain date) or in case of permission rules by defining duties (e.g. a payment of 10 euros is required).

Listing 2.3 demonstrates how ODRL can be used to represent the *CreativeCommons* license `CC-BY 4.0`.

Listing 2.3: `CC-BY 4.0` represented in ODRL

```
<http://purl.org/NET/rdflicense/cc-by4.0>
  a odrl:Policy ;
  rdfs:label "Creative Commons CC-BY" ;
  rdfs:seeAlso <http://creativecommons.org/licenses/by/4.0/legalcode>;
  dct:source <http://creativecommons.org/licenses/by/4.0/> ;
  dct:hasVersion "4.0" ;
  dct:language <http://www.lexvo.org/page/iso639-3/eng> ;
  odrl:permission [
    odrl:action cc:Distribution, cc:Reproduction, cc:DerivativeWorks;
    odrl:duty [
      odrl:action cc:Notice, cc:Attribution
    ]
  ] .
```

2.4.2 Tracking the Provenance of Data

In order to handle the unique challenges of diverse and unverified RDF data spread over RDF datasets published at different URIs by different data publishers across the Web, the inclusion of a notion of provenance is necessary. The W3C PROV Working Group [Lebo et al., 2013] was chartered to address these issues and developed an RDF vocabulary to enable annotation of datasets with interchangeable provenance information. On a high level PROV distinguishes between *entities*, *agents*, and *activities* (see Figure 2.5). A

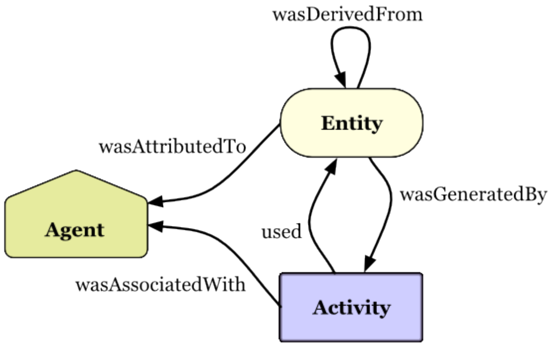


Figure 2.5: The core concepts of PROV. Example taken from [Lebo et al., 2013].

`prov:Entity` can be all kinds of things, digital or not, which are created or modified. Activities are the processes which create or modify entities. An `prov:Agent` is something or someone who is responsible for a `prov:Activity` (and indirectly also for an entity).

Listing 2.4 illustrates a PROV example (all other triples removed) of two observations, where observation `ex:obs123` was derived from another observation `ex:obs789` via an activity `ex:activity456` on the 1st of January 2017 at 01:01. This derivation was executed according to the rule `ex:rule937` with an agent `ex:fred` being responsible. This use of the PROV vocabulary models tracking of source observations, a timestamp, the conversion rule and the responsible agent (which could be a person or software component). The PROV vocabulary could thus be used to annotated whole datasets, or single observations (data points) within such dataset, or, respectively any derivations and aggregations made from Open Data sources re-published elsewhere.

Listing 2.4: PROV example

```
ex:obs123 a prov:Entity ;
   prov:generatedAtTime "2017-01-01T01:01:01"^^xsd:dateTime;
   prov:wasGeneratedBy ex:activity456 ;
   prov:wasDerivedFrom ex:obs789 .

ex:activity456 a prov:Activity;
   prov:qualifiedAssociation [
      a Association ;
      prov:wasAssociatedWith ex:fred ;
      prov:hadPlan ex:rule397 .
   ] .
```

2.5 Knowledge Graphs

The phrase “Knowledge Graph” (KG) has been popularized in recent years, particularly since the launch of the Google Knowledge Graph in 2012. With this, also several other companies announced (and sometimes even published) their Knowledge Graphs. Examples include *social media* Knowledge Graphs such as the Facebook Graph API[24] and the LinkedIn Knowledge Graph,[25] the Lynx project[26] which builds a *legal* Knowledge Graph of heterogeneous compliance data sources (legislation, case law, standards, industry norms and best practices), the Springer Nature SciGraph,[27] AirBnB,[28] Microsoft,[29] etc.; which can be all subsumed as different approaches for “collecting, managing, integrating, publishing, annotating, processing and analysing diverse data using a graph abstraction” [Bonatti et al., 2019].

2.5.1 What is a Knowledge Graph?

In order to find a better grasp of this trend, there have been several different definition attempts of what a Knowledge Graph is; in the following we want to give a non-exhaustive overview:

- Marchi and Miguel [Marchi and Miguel, 1974] defined already in 1974 the phrase “Knowledge Graph” in the context of a teaching-learning process. In their mathematical definition they define a Knowledge Graph as a set of “knowledge units” (the points of the graph) and “prerequisite relation” (the connecting edges). The idea here is that in order to reach a new knowledge unit (in the graph) it is necessary to “know a set of units of knowledge” [Marchi and Miguel, 1974], i.e. to have visited the prerequisite edges of the current point. While this might be one of the earliest uses of the term, it is used in a rather specific context, and is quite different to the current understanding of what a Knowledge Graph is.
- In 2013 Pujara et al. define Knowledge Graphs as “[...] information extraction systems [...], which use a variety of techniques to extract new knowledge, in the form of facts, from the web. These facts are interrelated, and hence, recently this extracted knowledge has been referred to as a knowledge graph.” [Pujara et al., 2013] This definition rather focuses on the process of how the graph is constructed, than on the concrete structure.
- According to Paulheim “any graph-based representation of some knowledge could be considered a knowledge graph [including] any kind of RDF dataset as well as

[24] https://developers.facebook.com/docs/graph-api, last accessed 27/06/2019

[25] https://engineering.linkedin.com/blog/2016/10/building-the-linkedin-knowledge-graph, last accessed 27/06/2019

[26] http://lynx-project.eu/, last accessed 27/06/2019

[27] https://www.springernature.com/de/researchers/scigraph, last accessed 27/06/2019

[28] https://medium.com/airbnb-engineering/contextualizing-airbnb-by-building-knowledge-graph-b7077e268d5a, last accessed 27/06/2019

[29] https://concept.research.microsoft.com/, last accessed 27/06/2019

description logic ontologies" [Paulheim, 2017] and he uses the following characteristics to define Knowledge Graphs: (i) real world entities and their interrelations, (ii) defines a schema for these, (iii) allows potentially arbitrary interrelating entities with each other, and (iv) covers various topical domains. However, in particular the requirement (iv) would restrict a Knowledge Graph only to domain-independent (or at least cross-domain) graphs.

- Färber et al. [Färber et al., 2018] try to formally defined a Knowledge Graph, by using the RDF data model ("We define a Knowledge Graph as an RDF graph" [Färber et al., 2018]). An RDF graph is then defined following the default model [Cyganiak et al., 2014], i.e. as "a set of RDF triples where each RDF triple (s, p, o) is an ordered set of the following RDF terms: a subject $s \in U \cup B$, a predicate $p \in U$, and an object $U \cup B \cup L$." [Färber et al., 2018] This definition obviously excludes any Knowledge Graphs that are not based on RDF.

- In [Ehrlinger and Wöß, 2016] Ehrlinger and Wöß collect and analyze different, more recent, Knowledge Graph definitions (including [Paulheim, 2017, Färber et al., 2018, Pujara et al., 2013]) and conclude from these that a Knowledge Graph "[...] acquires and integrates information into an ontology and applies a reasoner to derive new knowledge." They argue that a Knowledge Graph is "superior and more complex than a knowledge base (e.g., an ontology) because it applies a reasoning engine to generate new knowledge and integrates one or more information sources." [Ehrlinger and Wöß, 2016]

- In the Journal of Web Semantics announced in 2015 a Special Issue on Knowledge Graphs using the following definition:[30] "Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities. They have become a powerful asset for search, analytics, recommendations, and data integration."

- Along similar lines, there is the definition by the Special Issue of the Semantic Web Journal on Knowledge Graphs (2018, Hitzler):[31] "A Knowledge Graph (KG) is a graph-theoretic knowledge representation that (at its simplest) models entities and attribute values as nodes, and relationships and attributes as labeled, directed edges. KGs have emerged as a unifying technology in several areas of AI, including Natural Language Processing and Semantic Web, and for this reason, the scope of what constitutes a KG has continued to broaden."

In the recent Dagsthul seminar on Knowledge Graphs in 2018 [Bonatti et al., 2019] we discussed previous definitions, their advantages and shortcomings. Based on the discussions and current developments we also argue for a looser and more permissive

[30] `http://www.websemanticsjournal.org/index.php/ps/announcement/view/19`, last accessed 27/06/2019

[31] `http://www.semantic-web-journal.net/blog/call-papers-special-issue-knowledge-graphs-construction-management-and-querying`, last accessed 27/06/2019

definition – along the lines of [Ehrlinger and Wöß, 2016, Paulheim, 2017] – and tried to come up with such an inclusive attempt: Generally, a Knowledge Graph can be viewed as "a graph of data with the intent to compose knowledge"; a "graph of data" is not limited to a particular graph model, but rather can be instantiated with a number of models, e.g., directed edge-labelled graphs (such as RDF triples), and property graphs. By "composing knowledge" this definition refers to the very general process of "of extracting and representing knowledge in a manner that enhances the interpretability of the resulting Knowledge Graph" [Bonatti et al., 2019].

2.5.2 History & Examples

In the following we want to strengthen our arguments for a more inclusive view on the concept of "Knowledge Graph" by giving a non-exhaustive collection of milestones and important KG examples which either arose from scientific projects or companies:

1984 **Cyc** [Lenat, 1995], started by Douglas Lenat in 1984, is the longest-lived attempt to encode common sense knowledge in a machine-readable way. The Cyc project is curated, i.e. the facts are hand-coded, and contained more than 2.2 million assertions (facts and rules) with over 900 person-years of effort to construct (according to a 2006 paper [Matuszek et al., 2006]).

2007 The **DBpedia** project [Auer et al., 2007] – in contrast to Cyc – is an automated extraction framework of structured content from the Wikipedia. The first DBpedia dataset was published in 2007; the current version describes 4.58 million things (i.e. extracted Wikipedia articles) and consists of 3 billion facts (RDF triples).[32]

2007 Similar to DBpedia, the **Freebase** project contained data harvested from Wikipedia, but also from other sources such as MusicBrainz [Swartz, 2002]. The project has been acquired by Google and shutdown in 2016. Freebase powered the first version of Google's KG; the data has been transferred to Wikidata in 2012, before shutting down the project [Tanon et al., 2016].

2008 **YAGO** (Yet Another Great Ontology) [Suchanek et al., 2007] again contains structured data extracted from the Wikipedia, WordNet [Miller, 1995a], and GeoNames. In [Färber et al., 2015] a comparative survey can be found that identifies the differences between DBpedia, Freebase and YAGO.

2010 While the above approaches are either based on a manual effort (e.g. Cyc) or on extraction of structured information, the idea of **NELL** (Never-Ending Language Learning) [Mitchell et al., 2018] is to extract information from unstructured web pages. NELL contains an initial set of classes and relations, and continuously crawls the web for new instances of these. In total the system extracted already 2.8 million instances of 1186 different classes and relations.[33]

[32] `https://wiki.dbpedia.org/about`, last accessed 01/07/2019

[33] `http://rtw.ml.cmu.edu/rtw/overview`, last accessed 2019-06-01

2012 **Wikidata** is a collaboratively edited KG with the goal of supporting the Wikipedia, started by the Wikimedia Foundation in 2012. New entries can be entered by any user (as for the Wikipedia) and consist of a label, a description, and several additional properties/statements. Also, regarding the available properties Wikidata follows a folksonomy approach by allowing any user to propose new properties, which have to be discussed/supported by the community to get added to the set of available Wikidata properties.

2012 Later in 2012 **Google** introduced their KG[34] as a tool to enrich and disambiguate search queries. While there is no official documentation on the technologies for the Google KG, it was initially based on Freebase and now harvested various sources, including Wikidata and Wikipedia.

2.5.3 Challenges and Possibilities

A Public FAIR Knowledge Graph: There is a trend of creating domain-independent, large-scale KGs; for instance, companies such as Google, Apple and Microsoft create their KGs to support entity and concept in their service.

The challenge here is if and how such a KG of "everything" can be created. In the 2018 Dagstuhl seminar on Knowledge Graphs [Bonatti et al., 2019] the main concern was that such a mass amount of knowledge should be "open to the public" in a FAIR manner. FAIR [Wilkinson et al., 2016] refers to the published guidelines and initiatives on how to improve Findability, Accessibility, Interoperability, and Reuse of digital assets. While we can find analogies and commonalities between the FAIR principles and the Linked Data principles [Berners-Lee, 2006], the latter are more permissive in terms of underlying technology and how to publish the data.

Of particular interest will be the challenge of Interoperability: There exists already a number of (publicly available) KGs which capture various aspects of the real world, however, these are available in heterogeneous formats and incompatible semantics.

Representing and capturing changes: The current popular KGs such as Wikidata do not track the evolution and changes of entities, such as events, languages, etc. To better fit the dynamic environment of the Web, the KGs themselves need the supporting technology to represent and capture highly dynamic and constantly evolving information [Bonatti et al., 2019].

[34] `https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html`, last accessed 01/07/2019

CHAPTER 3

Monitoring Data Quality on the Web

With the rising number of available resources on the Web, the (meta-)data quality in Open Data portals have been identified as the key barrier [Beno et al., 2017] for a wider adoption and one of the core problems for the overall success of Open Data [Zuiderwijk et al., 2012]. In fact, there have been a number of prior reports that confirm existing quality issues in Open Data [Kucera et al., 2013, Reiche et al., 2014, Umbrich et al., 2015].

For the data to be discoverable by consumers, the publishers need to describe their data in an accurate and comprehensive manner. Missing or incorrect metadata information prevents consumers from finding relevant data for their needs and as a consequence requires a substantial amount of time to (manually) scan the portals and the data itself to locate relevant data sources. Even worse, if a user finds interesting datasets, the data might not be available due to outdated links or might not conform with the format declared in the metadata (e.g., wrong file formats or formats that do not conform with their specifications).

In order to better understand the severity of such quality issues, we periodically measure and assess the quality of information in Open Data portals for various quality dimensions such as the retrievability of the actual data or the existence of contact or license information. We argue that the insights gained from such a large-scale assessment are not only useful to inform data and portal providers about particular problems, but also help to identify how, and at what stage in the publishing life cycle, quality improvement methods need to be applied. It will later on help us to develop tools to (semi-)automatically support the creation of data and its metadata, but also algorithms to automatically improve and repair wrong metadata descriptions.

To this end, we present a quality assessment and evolution monitoring framework for web-based data portal platforms, which offer their metadata in different and heterogeneous models. In particular, we make the following contributions:

(i) We provide a *generic formal model* which can be used to represent data and metadata in web portals and discuss general characteristics and quality metrics independently from the portal software frameworks.

(ii) We define a *set of quality metrics* for the DCAT metadata standard, grouped by five different dimensions and present mappings from metadata of three major Open Data portal software providers to our metadata model.

(iii) We introduce our efficient *quality assessment and monitoring framework* that is able to periodically process hundreds of data portals.

(iv) We *report findings* based on monitoring a set of over 261 Open Data portals. This includes the discussion of general quality issues, e.g. the retrievability of resources, and the analysis of the DCAT specific quality metrics (cf. contribution (ii)).

(v) We present an extension to our framework to directly *compare the different portals' quality assessment and their evolution* over time.

(vi) We *analyse a corpus of CSVs* from the monitored Open Data portals and study the characteristics and properties of the files from a consumers' point of view.

The remainder of this chapter is structured as follows: We propose a generic model for web-based data portals in Section 3.1, our contribution (i). Aligned to contribution (ii), we introduce concrete quality metrics based on DCAT in Section 3.2. We introduce contribution (iii), our QA framework and its implementation, in Section 3.3. We present and discuss contribution (iv), our concrete quality assessment findings, in Section 3.4. In Section 3.5 we present contribution (v), an extension to compare the portals' metadata quality evolution. Section 3.6 discusses our contribution (vi), a large-scale analysis and profiling of tabular data on Open Data portals. Eventually, we discuss related publications and projects in Section 3.7 and conclude with Section 3.8.

3.1 A Formal Model for Web Data Portals

Our proposed model for web-based data portals is inspired by the Streams, Structures, Spaces, Scenarios, Societies (5S) model [Gonçalves et al., 2004, Fox et al., 2012], which describes the components of *digital libraries* (e.g., metadata catalogs, collections, browsing and indexing services) through higher level mathematical objects. In detail, we base parts of our work on the "structure" and the "scenario" concepts, which are used within the 5S model to define a descriptive metadata structure and a set of services for a digital library, respectively.

We base our terminology and formalization on the 5S model the following way: Out of the extensive body of definitions in the 5S model we use the term and definition of *services* to introduce and define a set of services, which are offered by a portal (e.g., via an API request). Further, we use the graph-based definition of *descriptive metadata structures* of Gonçalves et al. [Gonçalves et al., 2004] to formalize metadata descriptions of the available datasets.

3.1.1 Generic Model for Web Data Portals

Let $\mathcal{P}$ denote a corpus of data portals, where $P \in \mathcal{P}$ is a single data portal, accessible via the URL h_P, which holds a set of dataset URLs $D_P = \{d_1, \ldots, d_n\}$ and a set of services $\mathit{Serv} = \{\texttt{list}, \texttt{meta}, \texttt{show}, \texttt{resource}\}$:

$$P = (h_P, D_P, \mathit{Serv}) \tag{3.1}$$

Such data portals further provide *metadata descriptions* of the listed datasets. A metadata description is a structured document holding important contextual information about a dataset. In the following, we use the notation $\mathcal{M}$ to denote the set of all available metadata descriptions over $\mathcal{P}$. Note that exactly one metadata description $m \in \mathcal{M}$ is associated with a dataset URL.

In the context of data portals, a *resource* is any target of an URL, which can be hosted internally (i.e., hosted on the same server as the portal) or externally (i.e., a reference to a distant web or file server). Typically we can access resources via links in the metadata descriptions or using the API of the data portal and usually we encounter downloadable files. We denote the set of all resource URLs occurring over the set $\mathcal{P}$ of data portals as $\mathcal{R}$ and the set of all dataset URLs as $\mathcal{D}$ respectively, where $D_P \subseteq \mathcal{D}$ for all $P \in \mathcal{P}$.

Services. In the following, we define the set of services $\mathit{Serv} = \{\texttt{list}, \texttt{meta}, \texttt{show}, \texttt{resource}\}$. These services are used by our framework to implement the harvesting and quality computation, cf. Section 3.3. We rely on the availability of these services for the automated computation of our metrics. Next, we describe the services in detail.

- `list`: Let `list` be a service that returns the set of all dataset URLs for a given Portal, i.e. formally defined as the function $\texttt{list} : \mathcal{P} \rightarrow 2^{\mathcal{D}}$, where in particular $\texttt{list}(P) = D_P$ for a portal $P = (h_P, D_P, \mathit{Serv})$.
- `meta`: Let `meta` be a service, formalised by a function $\texttt{meta} : \mathcal{P} \times \mathcal{D} \rightarrow \mathcal{M}$, that assigns each dataset URL $d \in D_P$ in a portal $P \in \mathcal{P}$ exactly one metadata description $m \in \mathcal{M}$.
- `show`: Let `show` be a service that provides the set of metadata descriptions for a given data portal P, i.e. $\texttt{show} : \mathcal{P} \rightarrow 2^{\mathcal{M}}$ with $\texttt{show}(P) = \{\texttt{meta}(P, d) \mid d \in D_P\}$.
- `resource`: In general, a dataset can describe and reference multiple resources. Therefore, the service `resource` returns a set of resource URLs for a given dataset URL:
$$\texttt{resource} : \mathcal{P} \times \mathcal{D} \rightarrow 2^{\mathcal{R}} \tag{3.2}$$

Note, that a specific resource URL can be described in various datasets. We can use this service to describe the set of all resource URLs occurring on a data portal P:

$$\bigcup_{d \in D_P} \texttt{resource}(P, d) \subseteq \mathcal{R} \tag{3.3}$$

Usually these services are directly available as HTTP-based RESTful APIs and therefore are accessed via the portal URL h_P, as in the case of CKAN, Socrata and OpenDataSoft. In case one of the services is not directly available as an API, we can implement the services for that particular portal software. For instance, the `resource` service can be implemented by using the `meta` service and extracting the resource URLs from the returned metadata. This flexibility allows us to integrate future portals which are hosted by other content-management software, e.g, HTML based portals without available APIs.

Metadata Descriptions. We assume metadata is organised as (potentially nested) key-value pairs, where the key contains the label of a property describing the respective dataset, and its value contains the corresponding description or numerical value of this property. For instance, on CKAN portals the metadata description of a dataset is accessible via an API service (`meta` or `show`) and the metadata, returned as JSON document, holds references to the actual resources.

We provide a general characterization of a metadata description which is applicable to any occurring concrete metadata instance for data portals. As such, we propose the following tree-based model of a metadata description which is inspired by the graph-based definition of a "descriptive metadata specification" in the 5S model [Gonçalves et al., 2004]:

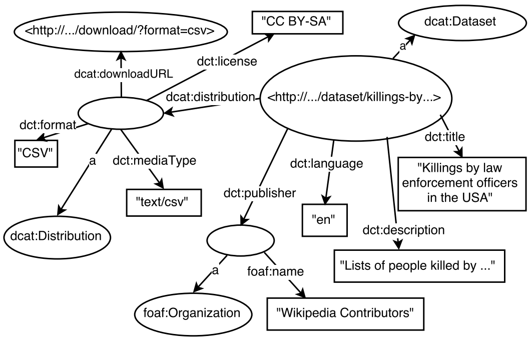
Let a metadata description $m \in \mathcal{M}$ be a a labelled tree $m = (V, E, \mathit{label})$ with the dataset URL as its root where each node $v \in V$ and each edge $e \in E$ can hold a label $\mathit{label}(v)$ (or $\mathit{label}(e)$ respectively), defined by the labelling function *label*. If there is no label specified for some node or edge then the function *label* returns ϵ (denoting an empty string).

The successor nodes of the root can be either internal nodes (i.e., a node with out-degree > 1) or a leaf nodes (also called terminal nodes). In the following, we denote the set of labels over all leaf nodes in a metadata instance m by $\mathit{leaves}(m)$. A path δ in m is a set of consecutive edges from the root to a leaf node. Let $\mathit{leaf}(\delta)$ return the single label of the leaf node of the corresponding path.

We note that in principle this generic metadata model covers any tree-based data structures such as XML, JSON and (acyclic) RDF descriptions – also typically represented nowadays in serialization formats such as JSON-LD [Sporny et al., 2014]. The RDF view intuitively correspond to triples $(n_1, \mathit{label}(v), n_2)$ for each edge $v \in V$ between nodes n_1, n_2.

The root $r \in V$ of a metadata instance m represents a dataset d in a portal and is labeled by the dataset URL. The adjacent edges of root r represent attributes and properties of

```
{
 "datasetid":"killings-by-law-enforcement-...",
 "metas":{
  "publisher":"Wikipedia Contributors",
  "language":"en",
  "license":"CC BY-SA",
  "title":"Killings by law enforcement officers",
  "references":"http://en.wikipedia...",
  "keyword":[
   "killings",
   "law enforcement officers",
   "USA"
  ],
  "description":"Lists of people killed by ..."
 }
}
```



(a) DCAT represented in tree structure.

Figure 3.1: Example mapping of an OpenDataSoft metadata description to DCAT.

the corresponding dataset. The label of an attached leaf node of an edge holds the value of a metadata property and branches at internal nodes describe sub-properties.

3.1.2 DCAT Model Instantiation

In the following we consider the `dcat:Dataset` and `dcat:Distribution` classes of the DCAT model [Maali and Erickson, 2014]. The definition of a `dcat:Catalog` class corresponds to the concept of data portals, i.e., it describes a web-based data catalog and holds a collection of datasets (using the `dcat:dataset` property). A `dcat:Dataset` describes a metadata instance which can hold one or more distributions, a publisher, and a set of properties describing the dataset. A `dcat:Distribution` instance provides the actual references to the resource (using `dcat:accessURL` or `dcat:downloadURL`). Further, it contains properties to describe license information (`dct:license`[1]), format (`dct:format`) and media-type (`dct:mediaType`) descriptions and general descriptive information (e.g, `dct:title` and `dcat:byteSize`).

In the following, we will apply the RDF structure of a `dcat:Dataset` together with its distributions and properties to the tree-shaped concept of a metadata description introduced in Section 3.1.1. We label the root node of the metadata description with the `dct:Dataset` subject (i.e., the dataset URL) and add an edge for each of the properties, linked with a node for the corresponding objects and values, respectively. For instance, the leaf of the path (`dcat:dataset`, `dct:publisher`, `foaf:homepage`) is used to describe provenance information in DCAT. Figure 3.1 (a) displays the tree structure of a DCAT metadata description.

[1] `dct:` denotes the Dublin Core Metadata namespace.

3.1.3 Metrics over General Data Portal Model

Initially, we define the set of all possible *metadata properties* within a metadata description. Clearly, a tree-shaped metadata description consists of a set of *paths* from the root node to the leaves. The sequence of edge labels of these paths describe metadata properties and the corresponding leaves hold the values of these paths. For instance, the path labelled $\langle \texttt{dcat:distribution}, \texttt{dct:format} \rangle$ in Figure 3.1 (b) describes the "format of a distribution".

In the following definitions, let Δ_m be the set of all paths from the root of a single metadata instance m to the leaves. We use δ to denote a single path in a metadata description and write $\mathit{label}(\delta)$ for the sequence of labels on the path. Note that necessarily $|\Delta_m| = |\mathit{leaves}(m)|$ holds.

Path Selector Function: Let $s_K(m)$ be a path selector function over a metadata description m which we assume for simplicity to be defined by a set of keys K, i.e., $s_K(m) = \{\delta \mid \delta \in \Delta_m \text{ s.t. } K \cap \mathit{label}(\delta) \neq \emptyset\}$. if we apply a path selector function with the $K = \{\texttt{dct:distribution}\}$ to the tree-structured DCAT metadata in Figure 3.1 (b) this will return 5 paths of this DCAT metadata description, including for example a path δ_1 with $\mathit{label}(\delta_1) = \langle \texttt{dcat:distribution}, \texttt{dct:format} \rangle$ with $\mathit{leaf}(\delta_1) =$"$\texttt{CSV}$".

Boolean Evaluation Functions over a Path: Let $f(\delta)$ be a boolean function over the leaf of a specific path δ which returns either 0 or 1. For instance $f(\delta)$, we will use the boolean function $\mathtt{nonEmpty}(\delta) = (\mathit{leaf}(\delta) \neq \epsilon)$ to determine if the labelling of some leaf of a path is non-empty.

Another example would be the function $\mathtt{isValidEmail}(\cdot)$ which is used to evaluate a regular expression on the value of the leaf of a given path. Further, we can use such a boolean function for evaluating user-defined functions, e.g. a function $\mathtt{isOpenFormat}(\cdot)$ (cf. details below in section 3.2.3) which returns 1 if the specified value of $\mathit{leaf}(\delta)$ is contained in a predefined set of labels corresponding to open format descriptions. For instance, for the path δ_1 from above $\mathtt{isOpenFormat}(\delta) = 1$ if we assume the value "CSV" among the set of open file formats.

Aggregation of Labels: Finally, for our metrics we will use an aggregation function $\mathtt{agg} \in \{\mathtt{min}, \mathtt{max}, \mathtt{avg}\}$ to specify how to aggregate $f(\delta)$ for all paths $\delta \in s_K(m)$, to denote minimum, maximum and average. For the special case that $s_K = \emptyset$ (i.e., none of the paths in the metadata description is matching the specified selector) we assume that $\mathtt{agg}$ returns 0 for any specific aggregation function, i.e., overall the aggregation always returns a value between 0 and 1.

Quality Metrics over Paths: We now define a basic quality metric over a metadata description m as:

$$\mathsf{Metric}(s_K, f, \mathtt{agg})(m) = \mathtt{agg}(\{f(\delta) \mid \delta \in s_K(m)\}) \tag{3.4}$$

For instance, we will use a OpenFormat quality metric defined as follows:

$$\mathsf{OpenFormat} = \mathsf{Metric}(s_{\{\texttt{dct:format},\texttt{dct:mediaType}\}}, \mathit{isOpenFormat}, \mathit{avg}) \tag{3.5}$$

Combined Quality Metrics: We can also combine several basic metrics again by aggregation. For instance, to calculate the average existence of discovery information in a DCAT metadata description, we use the following *combined metrics*:

$$\mathsf{Discovery} = \mathit{avg}\left(\begin{array}{ll} \{ & \mathsf{Metric}(s_{\{\texttt{dct:title}\}}, \mathit{nonEmpty}, \mathit{max}), \\ & \mathsf{Metric}(s_{\{\texttt{dct:description}\}}, \mathit{nonEmpty}, \mathit{max}), \\ & \mathsf{Metric}(s_{\{\texttt{dct:keyword}\}}, \mathit{nonEmpty}, \mathit{max})\} \end{array} \right) \tag{3.6}$$

Here, we calculate the average existence over results of different DCAT path selector functions. When applying this Discovery metric to the example in Figure 3.1, we observe a total value of 2/3: the first and second metrics (title and description) returns 1 since they exist and are non-empty (i.e., max aggregation yields 1), while the third metric returns 0 since there is no path with the `dct:keyword` property in the figure.

3.2 Metadata Mapping and Quality Dimensions

When investigating the metadata structure of common data publishing frameworks and portals (such as CKAN and Socrata) one observes different metadata schemas and heterogeneity issues. For instance, the Socrata framework describes licensing information by the single metadata key `license`, while in CKAN there are three different keys for specifying the ID, the URL and the name of a license.

This observation highlights the need for a common schema applicable to a range of metadata sources that can be used in order to improve the comparability and simplify the integration of data from different portals. This allows to compute our quality metrics for web data portals independently from their corresponding publishing software and metadata format.

As a first step towards a generalised metadata schema, we propose a manual mapping for metadata schemas observed on CKAN, Socrata and OpenDataSoft portals to the DCAT metadata standard. The proposed mapping is intended as a homogenization of different metadata sources by using the W3C's DCAT vocabulary [Maali and Erickson, 2014]. Our decision in favour of DCAT was influenced by the increasing momentum in terms of integration and implementations of DCAT in existing Open Data systems: CKAN has a plugin for exporting DCAT, Socrata can export DCAT per default and OpenDataSoft is using overlapping metadata key names to DCAT by design. That is, DCAT serves as a least common denominator for describing datasets in various formats and therefore allows us to homogenise metadata retrieved from different publishing frameworks.

3.2.1 DCAT Mapping

In Table 3.1 we introduce our mapping for the different metadata keywords, which is partially derived from the dataset harmonization framework proposed by [Assaf et al., 2015]. The mapping includes metadata keys from Socrata, CKAN and OpenDataSoft mapped to `dcat:/dct:` (Dublin Core Metadata) properties. The bold headers in the table indicate a class (i.e. an RDF subject) within the DCAT model; the part after → represents an RDF property. Blank fields within the table indicate that we were not able to match a corresponding key with the same semantic meaning. Please note, that individual datasets may contain a suitable key, but that we only map default, regularly occurring metadata keys.

For instance, `dcat:Dataset`→`dct:title` denotes an RDF triple (*`dataset`*, `dct:title`, *`title`*) in the resulting mapping, where *`dataset`* is a `dcat:Dataset` and *`title`* is the corresponding mapped value (i.e., a RDF literal holding the value of the mapped metadata key).

The proposed mapping of the keys is mainly based on matching names. For instance, considering the mapping of the OpenDataSoft metadata keys, we can see that all mapped keys use the same key-names as the DCAT vocabulary. If the key-names are not matching (as for most of the CKAN keys), we mainly rely on existing mappings, further explained in Section 3.2.1.

Figure 3.1 displays an application of the proposed DCAT mapping for an OpenDataSoft metadata description. The DCAT mapping is presented as a tree, where oval nodes represent RDF resources and square nodes represent literals. Note that the `dct:license` in the DCAT model belongs to a distribution, while in the original metadata it is attached to a dataset instance. (This holds likewise for the license keys in Socrata and CKAN portals.)

Adapting existing Mappings: In order to make use of the proposed homogenization within our QA framework (Section 3.3) we implemented a mapping algorithm for each of the data management systems covered by Table 3.1.

Regarding the CKAN software we took a closer look at the source code of the DCAT extension for CKAN,[2] currently being developed by the Open Knowledge Foundation. We used the existing mapping of datasets mostly "as is", except for the licenses information which is currently not mapped properly: the original mapping in the extension assumes a license key for each resource in a dataset which does not exist in CKAN datasets.

For Socrata portals, we mainly rely on the pre-existing DCAT output. Additionally, we modify the result so that it conforms to the standardized DCAT model. This means, firstly, we replace non-DCAT with standardised DCAT properties in the result if they

[2] `https://github.com/ckan/ckanext-dcat`, last accessed 03/02/2019. We currently use the code committed on August 13, 2015.

Table 3.1: DCAT mapping of different metadata keys.

DCAT	CKAN	Socrata	OpenDataSoft
dcat:Dataset			
dct:title	title	name	title
dct:description	notes	description	description
dct:issued	metadata_created	createdAt	-
dct:modified	metadata_modified	viewLastModified	modified
dct:identifier	id	id	datasetid
dcat:keyword	tags	tags	keyword
dct:language	language	-	language
dct:publisher	organization	owner	publisher
dct:contactPoint	maintainer, author (-email)	tableAuthor	-
dct:accrualPeriodicity	frequency	-	-
dct:landingPage	url	-	-
dct:theme	-	category	theme
dcat:Distribution			
dct:title	resources.name	-	-
dct:issued	resources.created	-	-
dct:modified	resources.last_modified	-	-
dct:license	license_{id, title, url}	licenseId	license
dcat:accessURL	resources.url	*export URL*[a]	*export URL*[a]
dcat:downloadURL	resources.download_url	-	-
dct:format	resources.format	*export format*[a]	*export format*[a]
dct:mediaType	resources.mimetype	*export mime-type*[a]	*export mime-type*[a]
dct:byteSize	resources.size	-	-

[a] Socrata and OpenDataSoft offer data export in various formats via the API.

are synonymous and secondly, we add provenance and authorship information if it is available in the default metadata.

Regarding the homogenization of OpenDataSoft portals we map the values of the metadata keys as described in Table 3.1.

3.2.2 Concrete DCAT Quality Dimensions

Commonly, data quality is described as “the fitness for use of information” [Bizer and Cyganiak, 2009] and is typically a multidimensional construct. The selection of a proper set of quality dimensions is highly context-specific since their purpose is testing the fitness for use of data for a specific task. In the context of Open Government Data (OGD) portals, this task is to obtain government information about the locality or country in question, for citizens and stakeholders [Kucera et al., 2013].

As such, we propose in the following a set of quality dimensions and metrics based on the available metadata attributes in the DCAT specification. We particularly consider metadata attributes that are relevant for assessing the quality of OGD [Veljković et al., 2014] (cf. Table 3.4), but for the actual assessments we limit ourselves to (i) metrics that are computable in an automated way (cf. Section 3.2.4), and (ii) attributes that are actually available in the data (cf. Table 3.8). An overview of our quality dimensions

Table 3.2: Quality Dimensions on DCAT keys.

Metric		dcat:Dataset	dcat:Distribution
	Existence *Existence of important information (i.e. exist certain metadata keys)*		
Access*	Is there access information for resources provided?		dcat:accessURL dcat:downloadURL
Discovery	Is information available that can help to discover/search datasets?	dct:title dct:description dcat:keyword	
Contact*	Existence of information that would allow to contact the dataset provider.	dcat:contactPoint dct:publisher	
Rights	Existence of information about the license of the dataset or resource.		dct:license
Preservation	Existence of information about format, size or update frequency of the resources	dct:accrualPeriod.	dct:format dcat:mediaType dcat:byteSize
Date	Existence of information about creation and modification date of metadata and resources respectively.	dct:issued dcat:modified	dct:issued dcat:modified
	Conformance *Does information adhere to a certain format if it exist?*		
AccessURL*	Are the values of access properties valid HTTP URLs?		dcat:accessURL dcat:downloadURL
ContactEmail*	Are the values of contact properties valid emails?	dcat:contactPoint dct:publisher	
ContactURL*	Are the values of contact properties valid HTTP URLs?	dcat:contactPoint dct:publisher	
DateFormat	Is date information specified in a valid date format?	dct:issued dcat:modified	dct:issued dcat:modified
License	Can the license be mapped to the list of licenses reviewed by opendefinition.org?		dct:license
FileFormat	Is the specified file format or media type registered by IANA?		dct:format dcat:mediaType

Table 3.3: Quality Dimensions on DCAT keys (cont'd).

	RETRIEVABILITY *Availability and retrievability of the metadata and data*	
Retrievable	Can the described resources be retrieved by an agent?	`dcat:accessURL` `dcat:downloadURL`
	ACCURACY *Does information accurately describe the underlying resources?*	
FormatAccr	Is the specified file format accurate?	`dct:format` `dcat:mediaType`
SizeAccr	Is the specified file size accurate?	`dcat:byteSize`
	OPEN DATA *Is the specified format and license information suitable to classify a dataset as open?*	
OpenFormat	Is the file format based on an open standard?	`dct:format` `dcat:mediaType`
MachineRead	Can the file format be considered as machine readable?	`dct:format`
OpenLicense	Is the used license conform to the open definition?	`dct:license`

and their metrics are listed in Table 3.2 and 3.3. These metrics are classified into five main categories: Existence, Conformance, Retrievability, Accuracy and Open Data fitness of information. All listed metrics focus only on metadata and shall enable an automated and scalable assessment. To put it another way, our research work does not yet include metrics that require to inspect the content of a dataset, and metrics that require a manual assessment are currently out of scope of the study.

Our definition of the Existence dimensions is inspired by other commonly used "completeness" metric [Pipino et al., 2002, Bizer and Cyganiak, 2009]. However, our existence metric differs in the sense that it gives an indication to what extent a certain set of DCAT keys are used (i.e., can be mapped) and contain information, while in other works the completeness is typically defined as the extent to which data is not missing [Pipino et al., 2002]. The existence dimensions (analogous to completeness) can be categorised as contextual [Zaveri et al., 2015] or context-based dimensions [Bizer and Cyganiak, 2009], i.e., as dimensions that "highly depend on the context of the task at hand" [Zaveri et al., 2015].

The Conformance dimension is inspired by the "representational-consistency" dimension which is defined as "the degree to which the format and structure of the information conform to previously returned information" [Zaveri et al., 2015]. However, our conformance dimension differs from consistency in terms of not comparing values to previously returned information, but by checking the conformance of values wrt. a given schema or standard. For instance, the Contact metric is a measure for the existence of contact information, while the ContactEmail metric is a conformance measure which checks if the available contact information is a valid email address.

Our Retrievability and Accuracy dimensions are aligned with existing ones: see accessibility in [Pipino et al., 2002, Umbrich et al., 2015] or availability in [Bizer and Cyganiak, 2009] for retrievability, and [Zaveri et al., 2015, Reiche et al., 2014] for accuracy. The accuracy dimensions – FormatAccr and SizeAccr – refer to the compliance of the actual content of the underlying resources with the metadata. In order to accurately assess these dimensions we need to inspect the actual content of the resource. In [Zaveri et al., 2015] the accuracy is therefore considered as an intrinsic quality dimension, i.e., it assesses if information correctly represents the real world.

The Open Data dimension is based on the Open (Knowledge) Definition.[3] It defines "open data" as an item or piece of knowledge which satisfies the following three requirements: (i) freely accessible as a whole, (ii) provided in a machine-readable and open format, and (iii) openly licensed. While (i) is already covered by the Retrievability dimension, we introduce the OpenFormat, MachineRead and OpenLicense metric to cover the requirements (ii) and (iii).

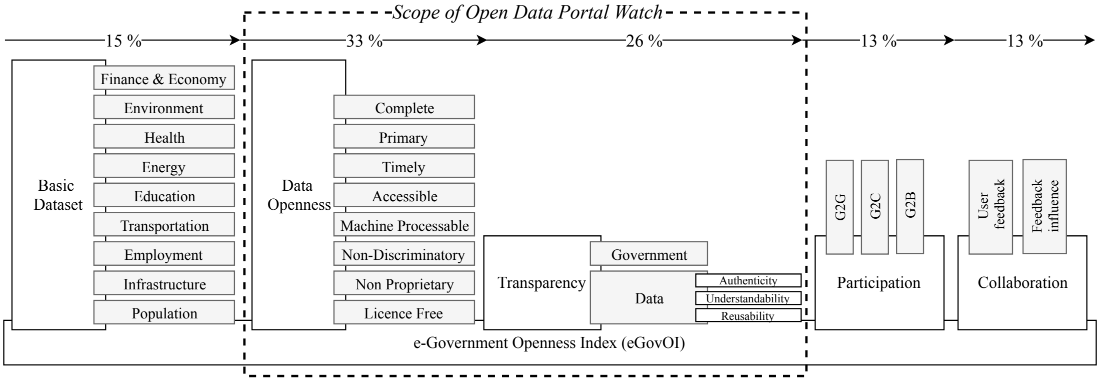


Figure 3.2: The e-Government Openness Index (eGovOI), as displayed in [Veljković et al., 2014].

Data Openness & Transparency Evaluation in the Literature

Evaluating openness and transparency in OGD and e-government depends on multiple dimensions [Bertot et al., 2012, Huijboom and Van den Broek, 2011, Janssen et al., 2012], the main ones being summarized by the e-Government Openness Index (eGovOI) [Veljković et al., 2014] (cf. Figure 3.2). Metadata of open datasets provide a useful basis for evaluating various aspects of such dimensions. For example, high-quality metadata is key for documenting results, so that they can be interpreted appropriately, searched based on what processes were used to generate them, and if they can be understood and used by other investigators [Gil et al., 2011, Sugimoto, 2014]. Unfortunately, in practice, assessing the quality of metadata information is not an easy and straightforward process; one of the major challenges lies in the lack of commonly agreed metadata representations [Zuiderwijk et al., 2012].

In the following, we discuss in greater detail how the proposed dimensions and associated metrics align with the eGovOI's [Veljković et al., 2014] openness and transparency criteria (cf. Figure 3.2). The alignment is displayed in Table 3.4, where the rows correspond to the eGovOI criteria and the columns to our quality metrics. We use a two-level scale (+, ++) to highlight whether our metrics *slightly* or *strongly* contribute to cover the eGovOI criteria.

- *Complete:* According to the eGovOI the completeness is calculated using five features: (i) availability of meta description, (ii) available download, (iii) machine readable, (iv) it is linked to other data, and (v) ease of data accessibility. In this regard, all quality metrics that fall under the EXISTENCE can be used to assess whether all metadata descriptions are available. The MachineRead metric also helps to assess whether a can be considered as machine readable, along with the

[3] http://opendefinition.org/, last accessed 2019-04-26

Table 3.4: Alignment of Open Data Portal Watch quality dimensions and the eGovOI key criteria for openness and transparency (cf. Figure 3.2).

eGovOI metric	Associated reference	Existence						Conformance							Accr		OD		
		Access	Discovery	Contact	Rights	Preservation	Date	AccessURL	ContactEmail	ContactURL	DateFormat	License	FileFormat	Retrievable	FormatAccr	SizeAccr	OpenFormat	MachineRead	OpenLicense
Complete	“all the information required to have the ideal data representation” [Veljković et al., 2014]	++	++	++	++	++	++						++	+	+	+		++	
Primary	“data is as collected at the source, with the highest possible level of granularity, not in aggregate or modified forms” [Group, 2007]					++	++				++		+		+		++	++	++
Timely	“data is made available as quickly as necessary to preserve the value of the data” [Group, 2007]					++	++									+			
Accessible	“timely and accurate decisions requires reliable and relevant information” [Rojas et al., 2014]		++		+			++						++	+		+	+	+
Machine Processable	“data is reasonably structured to allow automated processing” [Group, 2007]					+							+		+	+	++	++	
Non-Discriminatory	“data is available to the widest range of users for the widest range of purposes” [Group, 2007]				++						++								++
Non Proprietary	“data is available in a format over which no entity has exclusive control” [Group, 2007]				+								+		+	+	++	++	
License Free	“data is not subject to any copyright, patent, trademark or trade secret regulation.” [Group, 2007]				++						++								++
Authenticity	“government should publish information about data sources on portal, and provide possibility of reviewing datasets published by a specific data source” [Veljković et al., 2014]			++					++	++									
Understandability	“existence of textual description, searchable tags and links for a dataset” [Veljković et al., 2014]		++																
Reusability	“the 5 Star Open data scale is widely used to evaluate data reusability” [Berners-Lee, 2006]. “government should focus [...] more on open data reusability” [Sieber and Johnson, 2015]					+							+		+	+	++	++	+

accuracy dimension that checks whether the specified file format and size are correct. However, assessing whether "links to other data" exist is currently not supported, which would require to parse the content for links.

- *Primary:* The primary criterion is partially covered by the Open Data dimension, i.e. if the file format is conform with an open or machine readable format. However, we cannot assess if the data is published in the original/primary format or a transformation or aggregation operations have been performed prior to publishing. Indeed, this would require knowledge about the publishing process of the data provider.

- *Timely:* This criterion is partially covered by Preservation and Date: the former checks if there exists any update frequency information within the metadata, and the latter if any creation or modification date about the metadata and underlying datasets is provided. To achieve a sufficiently accurate assessment of dataset timeliness, a continuous monitoring and content inspection process of the resources would be required (as discussed in the following section, cf. "Freshness").

- *Accessible:* The AccessURL metric reports if the dataset can be directly downloaded by a client without any authentication. However, this metric currently does not cover scenarios in which a data consumer would need to manually invoke a download link.

- *Machine Processable & Non-Proprietary:* The MachineRead and OpenFormat metrics assess if the provided data formats can be considered as non-proprietary and machine processable (e.g., using JSON or CSV rather than an unstructured text file). Also relevant wrt. to this criteria is the FileFormat metric which checks if the file format or media type is registered by the Internet Assigned Numbers Authority (IANA).[4]

- *Non-Discriminatory & License Free:* Providing third parties with data in a usable form, without any restriction and for free, is assessed through the OpenLicense metric that checks if the provided data license is considered to be an open license according to the opendefintion.org. To cope with specific licensing situations (e.g., a license specific to a country policy), the Rights metric complements OpenLicense by identifying whether any licensing information has been provided within the metadata.

- *Authenticity:* Our metrics for contact information (Contact, ContactEmail, ContactURL) partly cover how authentic the data publisher is and if it is possible to contact the publisher. Any further authenticity check would require manual interaction.

[4] `http://www.iana.org/assignments/media-types/media-types.xhtml`, last accessed 2019-04-26

- *Understandability:* The understandability criteria is hard to assess in an automated manner and, as such, is not covered by our metrics. Nevertheless, the Discovery metric assesses the existence of keywords, titles and descriptions within the metadata, and can therefore serve as an indication if the content of a dataset is described. However, only a manual assessment would clearly determine for whom and to what extent the description of a dataset is understandable.

- *Reusability:* The reusability criterion is partially covered by our metrics. We currently do not inspect the content of the published data regularly, thus it is impossible to assess whether a dataset has been published following the 5 star Linked Data principles [Berners-Lee, 2006]; this would indeed require to inspect the content for links and verify that these links point to existing data. However, by assessing the machine readability of the published data formats, we already cover the first 3 principles of the 5 star model.

Freshness of Resources as a Quality Dimension

Timeliness [Pipino et al., 2002], or *Freshness* [Neumaier and Umbrich, 2016], is a measure of how sufficiently up-to-date a dataset is for a certain task (e.g., live timetables or current weather data). However, it is hard to automatically understand the time dimension from the metadata description of a dataset, e.g., to distinguish between static data vs. real-time data. Therefore, we consider the following Freshness dimension separately from the above metrics that are purely defined on (the availability and content) certain metadata fields.

Currently, Open Data portals do not often provide any information on how up-to-date the sources, the users and data providers are dealing with, really are. Some portals provide meta data fields to specify the update frequency of a data source. However, there is no guarantee that the change frequency information is correct and that the data source follows the specified change ratio. In order to create a freshness metrics that would solve the mentioned issues, one first has to learn the change history of a data source which ideally contains the exact time of the content change, and then apply a heuristic to estimate the next likely change time in order to estimate how up-to-date a data source is.

There are two possibilities to collect the change time for a given resource: i) push-based approaches for which the data publishers provides change notifications (in the metadata) and ii) pull-based approaches for which one periodically checks if a data source has changed. Only portal providers can have access to the former push-based information, either if the data providers upload a new version of a data source to the portal or if specific metadata fields, in the existing metadata, are edited. For all other use cases, one has to rely on the latter, pull-based approach, to collect change information.

3.2.3 Automated Assessment of Quality Metrics

To calculate the proposed metrics, we use the formulae introduced in Section 3.1.3. In general, the metrics are assessed by calculating the average (i.e. by using the aggregation function avg) over the set of corresponding DCAT properties. The star (*) besides a metric in Table 3.2 and 3.3 indicates that we use the max function to aggregate the values, which basically means that one positive evaluation is sufficient.[5]

Existence

To calculate the existence value for a specific quality metric we use the boolean evaluation function $\mathit{nonEmpty}$ from Section 3.1. The Access* and Contact* quality metric in Table 3.2 are defined by using the max aggregation function, e.g.:

$$\mathsf{Access*} = \mathsf{Metric}(s_{\{\texttt{dcat:accessURL},\texttt{dcat:downloadURL}\}}, \mathit{nonEmpty}, \mathit{max}) \tag{3.7}$$

The other existence metrics are defined using the avg aggregation. Discovery is calculated using the combined metrics as already introduced in Section 3.1.3 and the Rights metric is defined using a single DCAT property:

$$\mathsf{Rights} = \mathsf{Metric}(s_{\{\texttt{dct:license}\}}, \mathit{nonEmpty}, \mathit{avg}) \tag{3.8}$$

Conformance

The conformance metrics are assessed by using boolean evaluation functions which are either implemented using regular expressions or by specific functions for mapping licenses and file formats.

- *Using Regular Expressions.* In our conformance evaluation we use regular expressions to validate URLs, email addresses, and date formats of the AccessURL*, ContactEmail*, ContactURL* and DateFormat metrics, respectively. For instance, we calculate the AccessURL* metric of the dataset in Figure 3.1 by applying a regular expression for URLs to the value of the `dcat:downloadURL` property:

  $$\mathsf{AccessURL*} = \mathsf{Metric}(s_{\{\texttt{dcat:accessURL},\texttt{dcat:downloadURL}\}}, \mathit{isValidUrl}, \mathit{max}) \tag{3.9}$$

  For the DCAT metdata in Figure 3.1 the metric evaluates to 1 since `dcat:downloadURL` describes a valid URL.

- *License Conformance.* To validate the metadata description of a given license information we use a list of licenses provided by the Open Definition.[6] This

[5] We introduce this exception because for certain keys (e.g. the `dcat:accessURL` and `dcat:downloadURL`) the existence/availability of a value for one of these keys already provides the desired information.

[6] `http://licenses.opendefinition.org/licenses/groups/all.json`, last accessed 2019-04-26

list contains details about 109 different licenses including their typical ID, URL, title and an assessment if they are considered as ”open“ or not. The license information of a dataset in CKAN can be described with three different CKAN keys, namely `license_id`, `license_title` and `license_url`. In Socrata and OpenDataSoft there is only one `license` key which describes the license.

In our framework we implemented the user-defined function as a license matching heuristic which tries to match a dataset license to one of the licenses in the predefined list by performing the following steps. Firstly, we try to perform the match using the `license_id` value, if available. If this check fails we use next the `license_title`, which is matched either against the ID or title in the Open Definition license list. We perform this additional ID match because we observed that in several cases the datasets contain the license ID in the license title field. If this check also fails, we use as a fall back solution the `license_url` value for the match. Once a match was successful we consider a license as compliant.

- *File Format Conformance.* Regarding the conformance of file formats (FileFormat) we check if the normalised description (i.e., we remove leading dots and use lower case strings) is a format or a media type registered by the Internet Assigned Numbers Authority (IANA).[7]

Open Data

The assessment of openness and machine readability of licenses and file formats is based on specific boolean functions (cf. Section 3.1.3).

- *Format Openness.* Regarding the OpenFormat we use a `isOpenFormat`(·) function which checks for containment in a predefined set of confirmed open formats:

  *ascii,audio/mpeg,bmp,cdf,csv,csv.zip,dbf,dvi,geojson,geotiff,gzip,html,iati,ical,ics,jpeg*2000,
  json,kml,kmz,mpeg,netcdf,nt,ods,pdf,pdf/a,png,psv,psv.zip,rdf,rdfa,rss,rtf,sparql,
  sparql web form,svg,tar,tiff,tsv,ttl,txt,wms,xml,xml.zip,zip

  The formula used for calculating the format openness is already introduced in Section 3.1.3.

- *Machine-Readability of Formats.* Likewise, we defined a set of machine-readable file formats for the MachineRead metric:

  cdf,csv,csv.zip,esri shapefile,geojson,iati,ical,ics,json,kml,kmz,netcdf,nt,ods,psv,psv.zip,rdf,rdfa,
  rss,shapefile,shp,shp.zip,sparql,sparql web form,tsv,ttl,wms,xlb,xls,xls.zip,xlsx,xml,xml.zip

[7] `http://www.iana.org/assignments/media-types/media-types.xhtml`, last accessed 2019-04-26

The aforementioned collection of open and machine-readable formats are mainly based on a manual evaluation of file formats by the OpenDataMonitor project.[8]

- *Open Data fitness of Licenses.* We confirm the openness of the license (OpenLicense metric) per dataset by evaluating how the specified license is assessed in the list of licenses provided by the Open Definition (same list as in license conformance above). We decide on the openness of a license based on the above introduced license mapping heuristic, i.e., we use a boolean function *isOpenFormat* which returns 1 if we can map a license to the Open Definition list and the license is suitable to publish Open Data (according to this list).

  Please note the case that our metric reports only on the confirmed licenses. It might be that the non-confirmed licenses are also adhering to the Open Definition.

Retrievability

We calculate the Retrievability dimension by defining the boolean function `retr` using the HTTP status code of a GET request:[9]

$$\texttt{retr}(x) = \begin{cases} 1 & \text{if } \mathrm{GET}(x) = 2xx \\ 0 & \text{else} \end{cases} \tag{3.10}$$

Based on this boolean function we define the Retrievable dimension as follows:

$$\mathsf{Retrievable} = \mathsf{Metric}(s_{\{\texttt{dcat:accessURL},\texttt{dcat:downloadURL}\}}, \texttt{retr}, max) \tag{3.11}$$

Accuracy

The accuracy dimension reflects the degree of how accurately the available metadata values describe the actual data. In Table 3.3 we introduced two accuracy metrics for DCAT metadata keys: FormatAccr and SizeAccr. Most commonly, one defines different distance functions for the relevant metadata keys, e.g. a function which compares and calculates the distance between the value of the `dcat:byteSize` key and the actual size of the resource.

A possible indicator for the size of a resource is the `content-length` field in the HTTP response header. However, we observed a considerable number (∼22%) of resources with missing `content-length` information. Also, if available, this information could also refer to the compressed version and not the actual file size. Therefore, the reliable calculation of the SizeAccr metric requires the download and inspection of all referenced resources. Likewise for the file format we have observed in our experiments that even if the `content-type` header is available it is partially inconsistent with the real file

[8] `https://github.com/opendatamonitor/odm.restapi/blob/master/odmapi/def_formatLists.py`, last accessed 2019-10-16

[9] Note that we automatically follow redirects (i.e. 3xx status codes) and mean here the HTTP return code after such redirects.

formats (e.g., misconfigured web servers). As such, it is necessary to download and inspect the content to determine the real content-length, encoding and file format of a resource. However, and in order to keep our framework scalable (without the need to download all resources) we currently exclude these accuracy measures in our evaluation.

Freshness

In theory, we can use the change history of a resource to estimate how likely it is that the given resource is up-to-date. The change history of a resource contains the points in time at which changes to the content are known. In [Neumaier and Umbrich, 2016] we discuss and compare different freshness estimators based on the three possible scenarios when collecting the the change history from the Open Data portals. In the following we list the different change history scenarios:

1. *Push-based change history:* The first scenario assumes that the data provider pushes *change notification* to the portal, by either uploading a new version to the portal or by editing specific metadata fields in the description of the dataset for a resource. In that scenario, the change history will contain the complete set of all change times.

The second and third scenario assumes that the push-based change history of a resource is not accessible. It is either because we are not the portal provider or because the data is stored externally or the data provider does not provide metadata updates. In such case, we need to actively monitor the metadata of a resource for change information, upon available, or monitor the resource for content changes directly. In [Li et al., 2015], the authors identified two categories of *pull-based change history based on sampling*:

2. *Age sampling:* The second scenario assumes that an agent has access to the latest change time of a resource, also referred to as the *age of a resource.* Such age information can be either collected from specific fields in the metadata or HTTP response information.[10]

3. *Comparison sampling:* The third scenario relies upon monitoring the actual content of a resource and detect changes by *comparing two versions.* In contrast to the age-sampling approach, the third scenario only enables us to detect if there are any changes in comparison to the last sampling point.

3.2.4 Potential Shortcomings of the Automated Assessment

Shortcomings of the Conformance & Open Data Assessment:

[10] We refer to the `last_modified` header field which should be returned upon a HTTP GET or HEAD request. Yet, not all portals or HTTP servers provide such last-modification date information.

- Our License metrics are based on an automated mapping of the available license descriptions of the datasets to a list of Open Data conformant licenses by opendefinition.org. This clearly is a shortcoming of our assessment with respect to recall: we rely on the completeness of the list by opendefinition.org and therefore potentially miss conformant licenses not listed there. Our main argument for using opendefinition.org is that we can rely on a trusted source and get the information in a structured way. It is not in our expertise and not our aim to assess the openness of certain licenses ourselves.

- Similarly, our OpenFormat and MachineRead metrics are based on a fixed list that we adapted from the OpenDataMonitor project[11] and therefore might miss some file formats that can be considered machine-readable or open.

Shortcomings of the Retrievability Assessment:

- Most of the file hosts detect and block excessive requests of a single client. Therefore, in order to get a realistic retrievability assessment, the implementation has to meet some politeness policies. While we implemented basic crawling-politeness rules, i.e. shuffling of domains and waiting times between the requests, we cannot be sure if a negative respond is due to our crawling or due to other causes (e.g., in case of a 503 "Service Unavailable" server response).

- As already discussed in Section 3.4.5, we observed datasets with links to external landing pages, and no direct downloads of the data. In such cases our metric would just measure the retrievability of this Web page, and not of the actual resource.

- The Retrievable metric is based on the status code of a HTTP GET request. In an earlier version of the framework we implemented this metric using HTTP HEAD requests,[12] however, due to the high number of host not supporting HEAD, the HTTP GET is required.

3.3 Open Data Portal Watch Framework

The overall architecture of "Open Data Portal Watch", our quality assessment and evolution monitoring framework for Open Data (Web) portals, is shown in Figure 3.3 and comprises of four main building blocks, where each block contains a set of components:

- Input: The input block consists of several components to provide various iterators for data items that are processed and/or analyzed.

[11] `https://project.opendatamonitor.eu/`, last accessed 2019-10-20

[12] `https://tools.ietf.org/html/rfc2616#section-9.4`, last accessed 2019-10-18

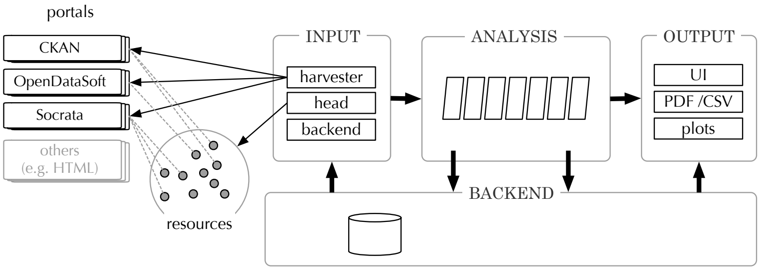


Figure 3.3: Open Data Portal Watch architecture

- Analysis: The data items provided by the input block are then piped through the analysis block which consists of a set of processing and analyzer components that can be chained into a processing pipeline.

- Backend: Both the input and analysis blocks interact with the backend unit in order to store or retrieve raw data or results.

- output: The components in the output block interact with the backend and analysis blocks and can be used to provide results and information in various formats (e.g., as CSV files or as JSON data for the UI).

3.3.1 Architecture

In the following, we discuss the components of each block in more detail.

Input: We implemented three different modules to access and retrieve data:

1. *Harvester.* The first component is called the *harvester.* It accesses the online data portals and retrieves all metadata about the datasets. Our framework currently provides three different harvester modules to invoke the specific service functions for the differently portal software (CKAN, Socrata, OpenDataSoft). The challenge we faced is that the service function for the same portal software might react very different across the portals or are not activated for every portal. Also temporary network or server errors can occur and need to be taken care of.

   In our harvesting component, initially, we invoke the `show` service of the portal to directly download the metadata descriptions of the datasets. Ideally this requires only one HTTP GET operation. However, we observed in practice and for the CKAN portals that it is more stable to combine the `show` function with pagination (i.e., not retrieving all metadata descriptions of a portal at once) which results in more requests but less data to generate on the server and to transfer for each request. It turned out, that pagination is extremely beneficial for larger portals with more than 1000 datasets.

Since we encountered server timeouts for some portals using the `show` service, we additionally make use of the `list` service of the portals: we retrieve the list of all dataset URIs and request the metadata description for each single dataset URI (using the `meta` service of the portal). Note, that this single processing highly influences the runtime of our harvesting algorithm. In order to avoid getting locked out by the server due to an excessive amount of HTTP requests, we wait for a short amount of time before executing the next query on a specific portal (cf. web crawling politeness [Najork and Heydon, 2002]). It is worth noting that using our implementation it is possible to process multiple portals in parallel.

2. *Head.* The second component performs HTTP HEAD requests on the resource URLs described in the datasets of the portals to check their availability and to gather more information The list of all unique resource URLs is extracted and stored in the database during the analysis of the harvested datasets. The header information which is retrieved is stored in the backend and analysed in the ANALYSIS block.

3. *Backend.* The third component of the INPUT module is used to supply the analysis block with data from the database instead of data from the portals and resources, respectively. For instance, this component can be used to recalculated quality dimensions for already stored datasets.

Backend: Our backend system is a Postgres (Version 9.4) database instance which makes use of the native JSON type feature to store schema-less information for datasets, resources and the portal metadata. For instance, we store the header information from the HTTP HEAD lookups in the resources table as JSON. In total, we have four main tables:

- One table to store basic information about each portal, such as the URL, API URL and the software).
- One table to store basic properties (e.g., number of datasets and resources) and the aggregated quality metrics for each portal and snapshot.
- One table to store the harvested information about each datasets for each portal and snapshot.
- One table to store all unique resources and the information from the HEAD lookups and the datasets and portals they are described in.

We further partition the dataset and resource table by snapshot for performance reasons.

Analysis: The components in our analysis block can be grouped into three categories: First, a set of components to calculate basic statistical information about the occurrence and distribution of various metrics, such as the number of datasets, resources, response code distribution, frequency count for licenses, formats, organisations, etc. Secondly, a set of quality assessment components, including our DCAT mapping, which calculates the previously introduced DCAT quality dimensions. Eventually, we implemented a set of components that interact with the backend in order to store the raw harvested data, the resource headers and the results from the quality assessment analyzers.

We pipe the retrieved datasets directly through our analysis block to calculate all measurements "on the fly". Since the portals can be treated independently, we process them in parallel. The retrieved datasets for each portal and snapshot are in addition stored/archieved in our backend system. This allows us on the one hand to share the collected portal snapshots with other researchers and on the other hand to re-compute metrics, or compute possible new quality metrics for already collected snapshots. In addition, the archived snapshots can be further exploited to analyse changes and modifications to the metadata which we plan to address in future work.

3.3.2 Server Error Handling

We implemented several strategies to cater for and prevent possible data loss caused by "server errors" during the harvesting process for a portal. If a portal is unavailable we restart the metadata harvesting process at a later stage.

Further, we occasionally observed server or timeout errors while invoking the `show` service due temporary server overload which might be caused by fetching potentially large sets of metadata descriptions. In that case, we re-invoke the `show` service with decreasing pagination size and increasing the wait time.

In order to trace possible server errors (but also bugs in our code) we store the debug and error logs for each harvested snapshot.

3.3.3 Data & Efficiency evaluation

One of the main requirements of our framework is to be able to periodically monitor the portals which depends on the time elapsed to harvest the metadata of all portals in the system. The present experiments are based on the active monitoring of 261 portals once a week. Please note, that the monitoring and harvesting process is influenced by external factors which cannot be assured to scale in all possible cases. For instance, if a data portal does not support the download of multiple metadata descriptions via pagination (cf. Section 3.3.1), we have no other alternative than to send a request for each single description (potentially even in a patient way, additionally respecting typical politeness delays between requests [Harth et al., 2006]).

Figure 3.4 plots the time elapsed to fetch all portals for an time period of 11 snapshots. The plot shows that our framework fetches the vast majority (>95%) of the portals in 10

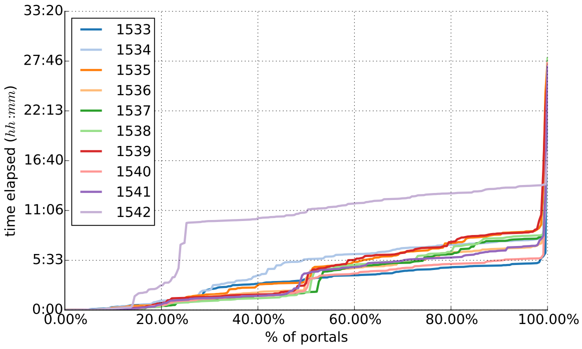


Figure 3.4: Elapsed time for harvesting process for the last 11 snapshots.

Table 3.5: Number of portals and processing time per snapshot

snapshot	1533	1534	1535	1536	1537	1538	1539	1540	1541	1542
$\lvert\mathcal{P}\rvert$	239	239	239	239	239	239	239	240	256	256
not available	8	10	11	8	8	9	8	8	13	13
fetched	231	229	227	231	229	228	231	230	243	243
fetch aborted	0	0	1	0	2	2	0	2	0	0
time (hh:mm)	27:29	28:03	27:48	26:41	27:35	28:05	26:01	27:33	27:09	17:33

to 12 hours and fetches the remaining individual portals in a total of about 27 hours.

In addition, Table 3.5 lists for each snapshot the total number of portals in our system ($\lvert\mathcal{P}\rvert$) and for how many of these portals we could successfully harvest all dataset descriptions.

As we can see, we had to terminate the fetch process for a maximum of 2 portals for the snapshots 1537, 1538, and 1540. In fact, the two responsible portals are huge CKAN portals for which we had to harvest the datasets one by one using the `meta` service since the `show` service was temporarily not available. Please also note that we have currently between 8 to 13 portals in the system for which we could not start the harvesting process, either because the respective portals were offline or returned API errors at the time of access.

3.4 Metadata Quality Assessment

In this section, we present the findings of our quality assessment for 261 Open Data portals for the snapshot of the fourth week of February 2016.

Table 3.6: Top-5 and bottom-5 portals, ordered by datasets.

domain of portal URL	Origin	Software	$\|\mathcal{D}\|$	$\|\mathcal{R}\|$
open.canada.ca	Canada	CKAN	244948	1163911
data.gov	US	CKAN	162351	763049
ckan.gsi.go.jp	Japan	CKAN	147955	147953
data.noaa.gov	US	CKAN	65915	475330
geothermaldata.org	US	CKAN	56391	62136
data.salzburgerland.com	Austria	CKAN	6	34
www.criminalytics.org	US	Socrata	6	-
bistrotdepays.opendatasoft.com	France	OpenDataSoft	4	-
www.opendatanyc.com	US	Socrata	2	-
ckanau.org	Ecuador	CKAN	1	2

3.4.1 Overview of portals

At the time of this analysis, our system held a total of 261 portals, of which are 148 using the CKAN software, 102 the Socrata software and 11 are powered by OpenDataSoft. The full list of all current portals is available on the web-interface of our framework.[13] In total, the 261 portals attribute to 1.1M datasets which describe 2.1M unique resources. Table 3.6 lists the top and bottom 5 portals with respect to the number of datasets. It is worth noting that 4 out of the top-5 portals are based in North America.

We collected the list of portals from various sources. One source is the list of customers on the homepage of the portal software providers (e.g. Socrata,[14] OpenDataSoft,[15] and CKAN[16]). Another source of portal URLs stems from the dataportals.org service which lists in total 431 Open Data publishing sites, out of which 125 are CKAN portals. Further, the OpenDataMonitor project also provided a list of 217 portals, including 52 CKAN portals.[17]

Table 3.7: Distribution of number of datasets over all portals.

$\|D\|$	<50	$<10^2$	$<5\times10^2$	$<10^3$	$<5\times10^3$	$<10^4$	$<5\times10^4$	$<10^5$	$>10^5$
$\|\mathcal{P}\|$	73	21	75	30	36	11	9	3	3

Table 3.7 lists the distribution of portals regarding their number of datasets. The table cells in Table 3.7 should be interpreted as intervals: for instance, in the 3rd column we can see that 75 portals hold between 100 and 500 datasets.

[13] https://data.wu.ac.at/portalwatch/portalslist

[14] https://opendata.socrata.com/dataset/Socrata-Customer-Spotlights/6wk3-4ija, last accessed 19/09/2018

[15] https://www.opendatasoft.com/customers/, last accessed 19/09/2018

[16] https://ckan.org/about/instances/, accessed 2018-09-19

[17] http://project.opendatamonitor.eu/, last accessed 26/04/2019

One can observe that the majority of ∼65% of the portals contains less than 500 datasets. The largest two portals are Canada's Open Government data catalog (`open.canada.ca`) consisting of ∼245k datasets followed by the official U.S. government data portal data portal `data.gov`.

Mapping and Usage of DCAT Metadata Fields: To allow a better understanding of which metadata attributes could be mapped to the DCAT vocabularies we present in Table 3.8 the densities of the attributes, i.e. the fraction of datasets for which a specific attribute is available. This allows us show and discuss the actual use of the attributes by the portals. An interesting observation is the average use of the topical/categorical attributes `dcat:keyword` and `dcat:theme`: 45% of all datasets provide at least one keyword, and only 28% use the `dcat:theme` attribute to classify the dataset; even worse for the spatio-temporal attributes `dct:spatial` (0.45) and `dct:temporal` (0.12).

Actually, most of the portals, however, use some kind of high level taxonomies – or at least tags – to categorise their datasets. As we will discuss in Section 4.1.1, the problem here is that CKAN offers no default field for categorisation of datasets, and that there is no common agreement on which additional metadata fields to use. The portals implement their own solution which would require manual mappings.

The fields `dct:spatial`, `dct:temporal`, and `dcat:theme` from Table 3.8 are not covered by our introduced metrics (cf. Section 3.2.2); this is justified on the basis of the low usage (below 50%) of these attributes across the portals.

Table 3.8: The density of selected attributes, i.e. the fraction of datasets for which the attribute is available; the attributes are ordered from left to right by overall density. The table lists the aggregated values for all portals, for the largest portals in the system (at the time of the snapshot), and for the Austrian `data.gv.at`.

Portal	`dct:title`	`dct:modified`	`dcat:accessURL`	`dct:description`	`dct:issued`	`dct:publisher`	`dct:format`	`dct:license`	`dcat:keyword`	`dct:spatial`	`dcat:theme`	`dct:temporal`
`all portals`	0.99	0.99	0.94	0.90	0.89	0.89	0.85	0.71	0.61	0.45	0.28	0.12
`open.canada.ca`	1	1	0.99	1	1	1	0.99	0.99	0	0.86	0	0
`data.gov`	1	1	0.96	0.92	1	0.99	0.86	0.70	0.42	0.76	0.42	0
`ckan.gsi.go.jp`	1	1	1	0	1	1	0.97	0	0.98	0.08	0	0
`data.noaa.gov`	1	1	0.89	0.73	1	1	0.84	0	0	0.97	0	0
`data.gv.at`	1	1	0.99	0.97	1	1	0.99	0.93	0.99	0	0	0

3.4.2 Retrievability

The results of our dataset and resource retrievability analysis are summarized in Table 3.9. We grouped the response codes by their first digit; *others* indicate socket or connection timeouts. As expected, nearly all datasets could be retrieved without any errors (∼98%). The 8026 datasets that could not be retrieved responded with a 403 Forbidden HTTP status code, indicating that an access token is required to retrieve the information.

A slightly different picture can be observed regarding the retrievability of the content of the actual resources. Out of a total of 2.6M resource values (i.e., values of the `dcat:accessURL` or `dcat:downloadURL` properties) appearing in 1.1M dataset descriptions, 2.1M are unique distinct values. We performed lookups on the valid URLs among these, resulting in the response code distribution in Table 3.9. Around 78% of these resources are accessible returning in a response code of 2xx. An slightly alarming observation is that 308k described resources (∼15%) returned a response code of 4xx, indicating that the resources is not available. A closer inspection of these results revealed that 176k resource URLs, hosted on Socrata portals, return a 400 code with the error message "*HEAD is not supported*". 14k resources (∼7%) caused some socket or timeout exception upon the lookup (indicated with others). In general, the number of exceptions should interpreted with caution since the unavailability of the content of a URL might be temporary due to internal server errors or network problems. In future work we plan to distinguish between persistent and temporary errors by considering the evolution of the URL's retrievability.

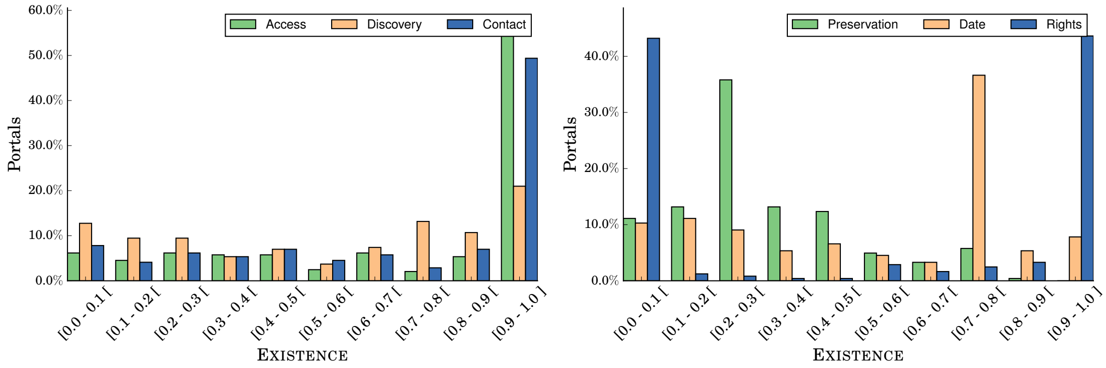
Table 3.9: Distribution of response codes.

		2xx	4xx	5xx	others
$\vert\mathcal{D}\vert$	1146435	1138246	8026	0	163
$\vert\mathcal{R}\vert$	2102778	1641098	308531	14410	138739

3.4.3 Existence

Next, we discuss the results of the metrics for the existence quality dimension which are displayed in the bar chart in Figure 3.5. In general, the metrics are rather equally distributed. A slightly concerning result is the existence of access information: only 50% of the portals have an Contact value over 0.9. For instance, the missing information does not allow data consumers to contact the publishers, e.g., to get more information about the data or to report errors.

Similarly, only 50% of the portals have a Rights value over 0.9 (i.e., there exists licensing information) and furthermore, about 45% of the portals do not provide any licensing information at all. This absence of license and rights information is extremely worrying considering that one core requirements for Open Data is that the data is published under an license which allows the open use of the content.



(a) Access, Discovery, and Contact information.

(b) Preservation, Date, and Rights.

Figure 3.5: Measurements for the Existence dimension. The y-axis gives the percentage of the portals; the x-axis gives the score for the respective metric, binned in an interval of 0.1.

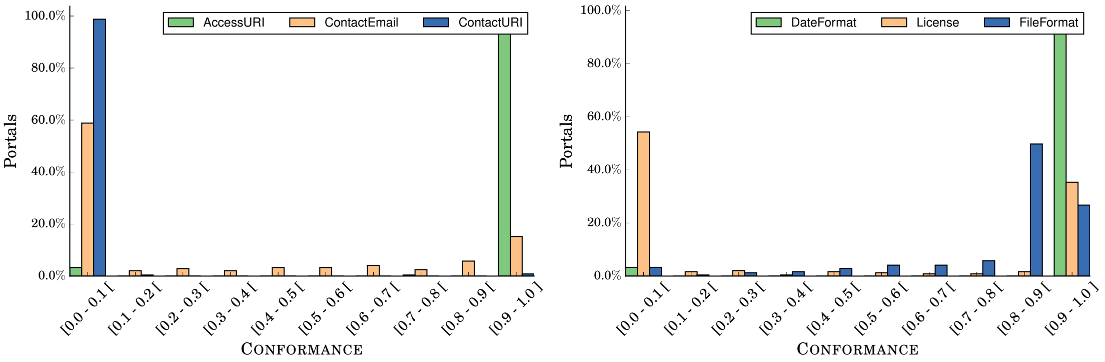
3.4.4 Conformance

Our results about the various metrics in the conformance quality dimension are shown in the bar chart in Figure 3.6. The left figure in Figure 3.6 shows the conformance distribution for the AccessURI, ContactEmail and ContactURI metric. Considering the conformance of access URIs (i.e., if the resource references are valid URIs), we observe that over 95% of the portals have an average AccessURI conformance value of over 0.9, indicating that the values are syntactically valid URLs.

Regarding the conformance of available contact information, we discover that only a small subset consisting of 20% of the portals have an average ContactEmail value of over 0.9 and about 60% of the portals do not really contain any valid email information. Regarding the appearance of URLs, we observed an average URL contactability for almost all portals of less than 0.1 (with one portal in the range 0.1 – 0.2, and only 2 portals with a value over 0.9, namely `data.overheid.nl` and `data.graz.gv.at`). This results show that there are basically no URLs or email addressed provided for contacting the publisher or maintainer of a dataset and a user would need to manual search for such information based on the provided text values.

The right chart in Figure 3.6 shows the remaining conformance metrics for DateFormat, License and FileFormat. Interestingly, for only $\sim$ 40% of the portals we were able to map almost all licenses (a License value over 0.9) to the list of licenses reviewed by opendefinition.org. The majority of the remaining portals have a value of less than 0.1. This shows, that there is more (manual) work necessary to be able to automatically identify the license information.

Regarding the specified file formats, we can see that about 80% of the portals have a FileFormat value of over 0.8, i.e., that for these portals almost all file format description are using format identifiers which are registered by the Internet Assigned Numbers



(a) Access and Contact information. (b) Date, License and File Format information.

Figure 3.6: Measurements of the Conformance dimension. Again, the bar charts indicate the percentage of portals (y-axis) with their respective score for a specific metric (x-axis).

Authority.[18] The DateFormat conformance for the occurring date descriptions is in general very high.

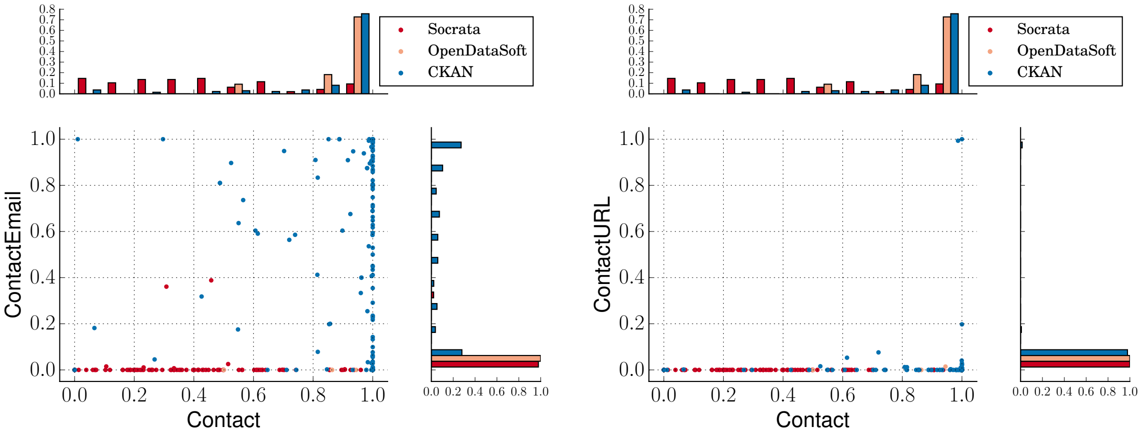
Overall, we can conclude that the majority of the portals have a low contactability value which bears the risk that data consumers stop using dataset if they cannot contact the maintainer or author (e.g., regarding the re-use if the license is not clearly specified or in case of any data related issue). Further, we have to admit that an automated identification of licenses is very hard to achieve, and that a better source of license IDs and licensing information is required to get better license conformance results.

Conformance vs. Existence: In the following we discuss the relation between existence and conformance values for different metrics for all portals in form of scatter plots. We further categorise the portals according to their publishing software.

The scatter plots in Figure 3.7 shows the relation between the Contact-existence and the ContactEmail and ContactURI conformance metrics, respectively. We can see in both plots that in contrary to CKAN portals, the OpenDataSoft and Socrata portals (red and yellow coloured) do not provide contact information which are valid email or URL addresses. Further, we can also see that the existence of any contact information is rather high for OpenDataSoft and CKAN portals and equally distributed for Socrata portals.

Figure 3.8 displays the Preservation and FileFormat metrics grouped by the portal software. Interestingly, the file format conformance based on the IANA registration is in general rather high. Drilling deeper, we see that almost all Socrata portals have a FileFormat value of about 0.8. The reason for this is that the Socrata software represents the actual data in 6 different formats with their respective media types (e.g. CSV, JSON, XML, RDF) and out of these file formats and media types the values `application/excel` and

[18] `http://www.iana.org/`, last accessed 2015-11-02



(a) Contact vs ContactEmail.

(b) Contact vs ContactURI.

Figure 3.7: Existence and conformance of contact information.

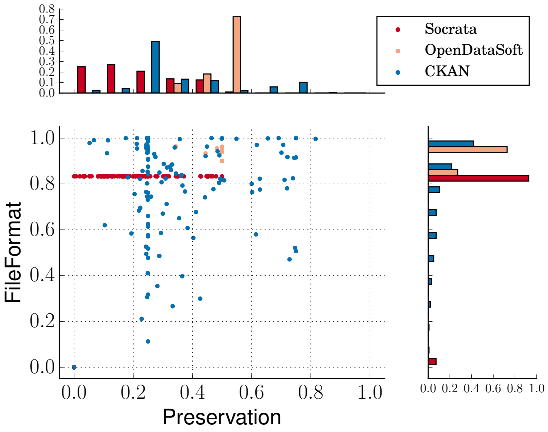


Figure 3.8: Preservation vs FileFormat.

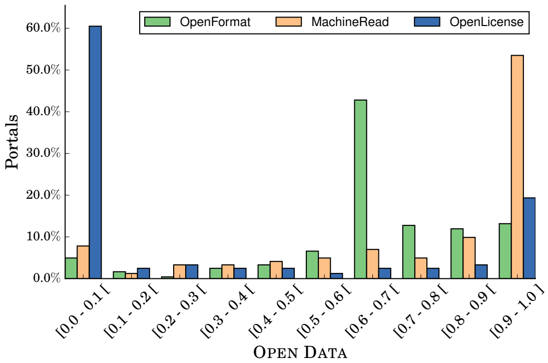


Figure 3.9: Openness of Licenses and Formats.

application/xml+rdf are not registered by the IANA; resulting in a conformance values of 10/12.

Noticeable in this plots is the Preservation value of 0.25 (x-axis) for a high percentage of CKAN portals. The reason for this observation is that most of the datasets in CKAN portal provide preservation information which can be mapped to only one of the four DCAT keys, namely dct:format, resulting in an average value of 1/4. We observe a similar result for the OpenDataSoft portals with the majority of the portals showing an preservation value between 0.5 – and 0.6.

Table 3.10: Most frequent formats and licenses.

license_id	$\lvert\mathcal{D}\rvert$	%	$\lvert\mathcal{P}\rvert$		format	$\lvert\mathcal{R}\rvert$	%	$\lvert\mathcal{P}\rvert$
ca-ogl-lgo	239662	32.3	1	1	**HTML**	491,891	25	74
notspecified	193043	26	71	2	PDF	182,026	9.2	83
dl-de-by-2.0	55117	7.4	7	3	**CSV**	179,892	9.1	108
CC-BY-4.0	49198	6.6	84	4	XLS(X)	120,703	6.1	89
us-pd	35288	4.8	1	5	**XML**	90,074	4.6	79
OGL-UK-3.0	33164	4.5	18	6	**ZIP**	50,116	2.5	74
other-nc	27705	3.7	21		...			
CC0-1.0	9931	1.3	36	11	**JSON**	28,923	1.5	77
dl-de-by-1.0	9608	1.3	6		...			
Europ.Comm.[a]	8604	1.2	2	16	**RDF**	10,445	0.5	28
others	80164	10.8			others	813,915	41.4	

[a] http://open-data.europa.eu/kos/licence/EuropeanCommission

3.4.5 Openness

A crucial requirement for opening up datasets is that the published resources and file formats adhere to the open definition: everybody should be allowed to *use, re-use and modify* the information which should be provided in an *open and machine-readable* format.

Table 3.10 displays on the left hand the top-10 used licenses and the number of datasets they are specified in, and on the right hand the top formats and the number of unique resources together with their number of portals they appear in. Bold values indicate that the license or format is considered as *open* by our metric. Note, that all these numbers are based on available metadata information of the datasets and can potentially be higher due to varying spellings, misspellings, and missing metadata. Therefore, these numbers should be considered as a lower bound for the respective formats.

By looking at the top used formats in Table 3.10, we can see that most of them are covered by *open formats* but in the top-10 only XML and CSV can be considered as *machine-readable*. A surprising observation is that ∼9% of all the resources are published as PDF files, and 25% of the resources are labelled as HTML. This is remarkable, because strictly speaking PDF and HTML cannot be considered as Open Data formats: while PDFs may contain structured data (e.g. in tables) there are no standard ways to extract such structured data from PDFs – or general-purpose document formats in general. Therefore, PDFs – as well as HTML documents – cannot be considered as easily machine-readable, nor as a suitable way for publishing Open Data.[19]

[19] Although there are works on extracting structured information from PDFs (e.g. tabular data within PDFs [Yildiz et al., 2005]), this topic is complementary to the scope of this thesis.

Why are there PDF and HTML documents? To better understand our results we want to discuss – based on our experience – the four main explanations for the high number of PDFs and HTML resources on the data portals:

Specification: We observe several data portals where the documentation and specification of the data is published as additional PDF files. For instance, the datasets at the Canadian `open.canada.ca` are largely accompanied by two PDFs describing the attributes in French and English.[20]

Transparency: Some of the large governmental data portals merge their "Open Data" and "Open Government" approaches. This results in portals where we find large numbers of text documents such as public reports, studies, parliamentary and senate decisions, etc. To give an example: at `transparenz.hamburg.de` ∼50% of the datasets are PDFs, resulting from the transparency law of the city of Hamburg.

Landing Pages: Another noteworthy observation is the use of additional landing pages for datasets: several portals do not offer direct downloads of the data, but only links to external landing pages – registered as "HTML" – where the actual data can be downloaded. This, for instance, explains the high number of HTML resources at the UK's data.gov.uk (∼25% of the datasets). Clearly, these additional landing pages are a problem for an automated processing of resources.

Multiple Formats: Eventually, a common pattern for publishing Open Data is serving the dataset in multiple formats; for instance, besides publishing the same content as CSV and XLS tables, there is also a PDF containing the table.

As we also see in Table 3.10, RDF does not appear among the top-15 formats for Open Data publishing.[21] This underlines the previously stated hypothesis that – especially in the area of Open Government Data – openly available datasets on data portals are mostly not published as RDF or Linked Data.

Also, we can see that JSON does not appear among the top ten formats in terms of numbers of published data resources on Open Data portals. Still, we include those main formats in our discussion, as

- particularly JSON and RDF play a significant role in metadata descriptions;
- JSON is the prevalent format for many Web APIs;
- RDF, as we saw, is apart from the Linked Data cloud prevalent in Web pages and crawls through its increasing support as an annotation format by popular search engines.

[20] See `https://open.canada.ca/data/en/dataset/0f45b62a-b4f6-4eaa-ad84-2fca80108d0b` as an example.

[21] The numbers for the RDF serializations JSON-LD (8 resources) and TTL (55 resources) are vanishingly small.

Country-specific Licenses vs. Creative Commons: Regarding the used license IDs in Table 3.10, we observe that the *confirmed*[22] open licenses in the top-10 cover only ∼17% of all datasets; more than 50% of the monitored datasets announce somehow in the metadata some kind of license information. Further, we notice that some of the more frequent licenses are only used in very few portals. For instance, the first ranked "ca-ogl-lgo", the Canadian Open Government License[23], is a share-alike license used throughout the Canadian Open Data portal `open.canada.ca`. It is worth mentioning in this context that some of the governmental data portals issue licenses for all datasets of the portal at a single place; for instance, the UK's `data.gov.uk` states that "all content is available under the Open Government Licence v3.0, except where otherwise stated" (cf. OGL-UK-3.0 in Table 3.10) in the footer of the page.

In Figure 3.9 the distribution of the Open Data assessments are displayed: the average OpenFormat, MachineRead, and OpenLicense values per portal. We can see that around 20% of the portals have a high average of confirmed *license openness* value of over 0.9. There is also a large group of around 60% of the portals for which we could only confirm an average license openness per dataset of less than 0.1.

Considering file format information, we observe a high *machine readability* with around 60% of the portals having an average value of over 0.9. In contrast, the average values for the OpenFormat metric spread more or less equally from 0.1 to 0.9, with a peak of about 40% of the portals for the values 0.6 - 0.7.

Overall, we could not confirm for the majority of the portals that their datasets provide an open license and their resources are available in open formats. However, the machine readability of formats yields marginally better results. These results indicate that we need to further investigate methods to address the unconfirmed licenses and formats.

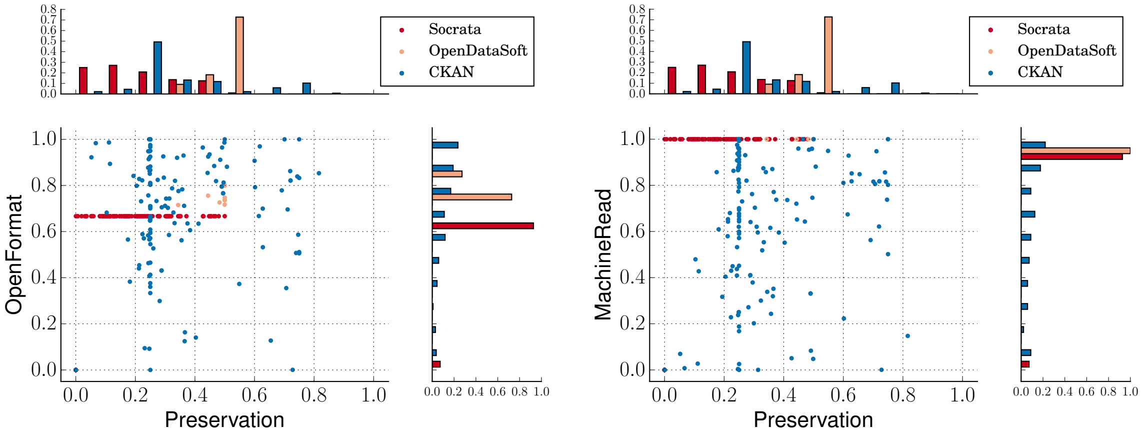
Existence and Open Data Fitness of Formats: The scatter plot in Figure 3.10 displays the Preservation-existence values vs. the confirmed openness (a) and machine-readability (b) values of file format descriptions.

On the left plot we can observe a consistent OpenFormat value around 0.66 for the Socrata portals. This results from the 6 possible format representation the Socrata software offers by default. Similarly, we see a high percentage of OpenDataSoft portals with a OpenFormat value in the range between 0.7 – 0.8. In contrast, the values for CKAN portals spread across the whole spectrum.

Considering the machine-readability of the formats on the right plot, we see that all offered file formats for Socrata and OpenDataSoft portals are identified as machine-readable, whereas average values for the CKAN portals are equally distributed.

[22] We have to stress again that our assessment is based on a list of licenses provided by the Open Definition (cf. Section 3.2). Obviously, we could assume that the country-specific Open Data licenses (e.g., the German CC-BY license on the third rank) comply with the Creative Commons and the Open Definition, however, for an automated assessment we have to rely on a curated and machine-readable list.

[23] `http://open.canada.ca/en/open-government-licence-canada`, last accessed 13/02/2019



(a) Preservation vs. OpenFormat.

(b) Preservation vs. MachineRead.

Figure 3.10: Existence vs. Open Data fitness of format descriptions.

We notice from the various scatter plots that Socrata and OpenDataSoft portals show very homogeneous results with respect to our DCAT conformance and openness dimensions. This is expected because both frameworks provide no flexibility to publish data in arbitrary formats. In fact, both systems require to upload the data in tabular representations. In contrast, the CKAN software is highly extensible and does not put any restrictions on the file formats which results in more heterogeneous quality values.

3.4.6 Freshness Estimation

We empirically study the metadata of 4.4m resources of our corpus of data portals, of which are 2.2m distinct resource URLs, and investigate which of the change history scenarios – *pull-based* history, i.e., data publishers provide change notifications vs. *push-based* history, i.e., one periodically checks if a data source has changed (cf. Section 3.2.3) – can be applied to estimate the freshness for a given resource. Additionally, we study how many resources are available for a freshness estimation for each portal framework separately.

Resource hosting: The first study is to compute the ratio of local vs external sources to estimate to how many portals we could apply only the push-based freshness estimators. Since Socrata and OpenDataSoft portals host all of their resources locally (and therefore change notification should be available), in this evaluation, we will focus only on the data of the CKAN portals. To compute if a resource is locally stored at the portal server, or not, we (i) match the domain name of the resource URL to the domain name of portal URL and (ii) inspect if the metadata key-value pair '*url_type*': '*upload*' is present in the metadata of the resource, indicating that the resource was uploaded to the portal.

The results of analysing 130 CKAN portals, describing 3.8m resources (with 2m distinct resource identifiers), are listed in Table 3.11; portals are grouped by their local vs external ratio. Only a small amount of 9 CKAN portals host all of the described resources locally, while the majority of the resources belong to 54 portals with a local to external ratio between 0 and 25%. Another 40 portals have a local to external resource ratio between 25% and 99.99%. This indicates, that push-based freshness estimators are not sufficient for CKAN portals and pull-based freshness need to be applied in addition to cover all resources.

Table 3.11: Ratio of local vs. external resources on CKAN.

ratio	0	< 0.25	< 0.5	< 0.75	< 1	1
$\|p\|$	27	54	9	7	27	9
% of $\|r\|$	5.76%	88.48%	0.38%	0.05%	1.12%	4.21%

Availability of change time information: Next, we inspect the metadata of the datasets and the resource HTTP header responses for change time information about the resources. In the CKAN portals, we scan the metadata for the "`last_modified`" field, in Socrata portals for the "`rowsUpdatedAt`" field and in OpenDataSoft metadata for the "`modified`" field. We performed a lookup on valid URLs to collect their HTTP response headers to find out if the resource was available (response code between 200 and 400). Please note, that we perform HTTP HEAD lookups, if applicable, to retrieve the necessary information. Since Socrata portals do not implement the HTTP HEAD protocol, we performed HTTP GET operations and cancel the connection after receiving the first content bytes.

The results of this study are summarised in Table 3.12. An interesting observation is the ratio of total vs distinct resource identifiers; we see that the resources in CKAN portals occur in more than one portal. We also see the very low number of local resources in CKAN portals (cf. Table 3.11).

Regarding the change time information in the metadata (**metadata field**), we observe that nearly all datasets in Socrata and OpenDataSoft portals provide these information with values. This stands in contrast to observations about CKAN portals. Here a very small percentage of datasets provide actual time information about changes in the metadata, of which most datasets describe external resources. This results indicate that the majority of the CKAN portal data publishers do not yet update the metadata information of their datasets if the resource content changed.

Inspecting the **HTTP response headers** of a total of 3.1m resources reveals that ∼66% of the CKAN, 100% of the Socrata and 0% of the OpenDataSoft resources have the `last_modified` header field.

Overall, our empirical evaluation of the portals reveals that applying only push-based freshness estimators is suitable for Socrata and OpenDataSoft portals; all their resources are locally stored. For CKAN portals there is a need for applying additional pull-based

Table 3.12: Change information in metadata and resources.

	CKAN	**Socrata**	**OpenDataSoft**
resources	4049851	181548	8757
distinct	2116940	165966	8757
local	227204	181548	8757
metadata field			
exists	3884657	175332	8742
with age value	146230	175332	8742
external with age value	130587	0	0
HTTP response headers			
URLs	4049851	181548	8757
HTTP lookups	3097665	67198	7958
with age value	1936612	67198	0
no age information	122612	5331	15

estimators: A direct inspection of the HTTP header information and downloading the content of the files is required, since not all metadata provide change time information.

3.5 Comparison of Metadata Quality

Comparing the quality of Open Data is not a straightforward process because it implies to take into consideration (i) multiple quality dimensions whose quality may vary from one another, and (ii) various Open Data end-users who – *depending on their role/needs* – may have different preferences regarding the importance of each dimension. To address this Multi-Criteria Decision Making (MCDM) problem we use the Analytic Hierarchy Process (AHP) to integrate various data quality dimensions as well as end-user preferences. We compare the Open Data portals indexed in the Open Data Portal Watch based on the previously introduced quality dimensions and metrics.

3.5.1 AHP-based Comparison of Portals

AHP, originally introduced by [Saaty, 1996], has the advantage of organizing critical aspects of the problem in a manner similar to that used by the human brain in structuring the knowledge, i.e. in a hierarchical structure of different levels, namely: the overall goal, the criteria (potential sub-criteria), and the alternatives. The MCDM ranking problem of our study is broken down into the hierarchical structure, which consists of four distinct levels:

- *Goal level:* to assess and rank the monitored Open Data portals in terms of published metadata quality;

- *Criteria & sub-criteria levels:* respectively correspond to the quality dimensions and sub-dimensions given in Table 3.2 and 3.3;
- *Alternative level:* the alternatives correspond to the set of monitored portals.

Given the criteria hierarchy, AHP does perform the following computation steps to obtain the final ranking of alternatives with respect to the overall goal:

1. Compare each element in the corresponding level and calibrate them on the numerical scale.
2. Perform calculation to find the maximum eigenvalue, consistency index (CI), consistency ratio (CR), and normalised values;
3. If CI and CR are satisfactory, then decision/ranking is done based on the normalised eigenvalues.

Given the AHP structure, several computational steps are performed to obtain the final ranking of alternatives with respect to the overall goal. Nonetheless, in view of the thesis' scope, we decided not to detail such computational steps here, but the reader can refer to [Kubler et al., 2016] to obtain more details. Indeed, even though the referenced paper focuses only on metrics specific to the CKAN software, the computational steps related to AHP remain unchanged.

In the end, after applying the AHP approach, each portal will be ranked amongst the portals assessed and compared in the study, and more specifically each portal will obtain a ranking (i) per quality dimension (i.e., Existence, Conformance, Retrievability, Accuracy, and Open Data) as illustrated in Table 3.2 and 3.3, as well as (ii) an aggregated one that results from the aggregation of these four rankings and the personal preferences specified by the end-user.

3.5.2 Portal Ranking Example

First, we want to analyze the portal rankings without prioritizing any quality dimension. Figure 3.11 gives insight into the quality comparison results, where the x-axis refers to the 259 portal indices (the indices can be found in Appendix A.2) and the y-axis to the relative quality score obtained after applying AHP. It can be observed that portals 67 (`data.gov.uk`) and 107 (`data.ottawa.ca`) have the highest scores when having all criteria equal in importance. Given, for instance, that an end-user is particularly interested in portals located in Brazil, since she/he is carrying out a study on the quality of Open Data portals managed by Brazilian institutions/organizations, the plot seems to highlight that portal 22 (i.e., dados.recife.pe.gov.br) has the best quality among the five Brazilian portals; cf. the red highlighted results in Figure 3.11.

Let us assume now the end-user wants to give higher priority to the "Open Data" quality dimension (e.g., extreme importance over the other dimensions at level 2). To

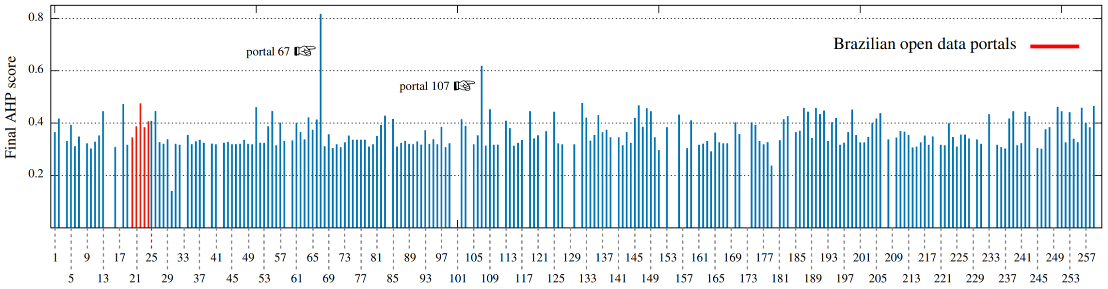
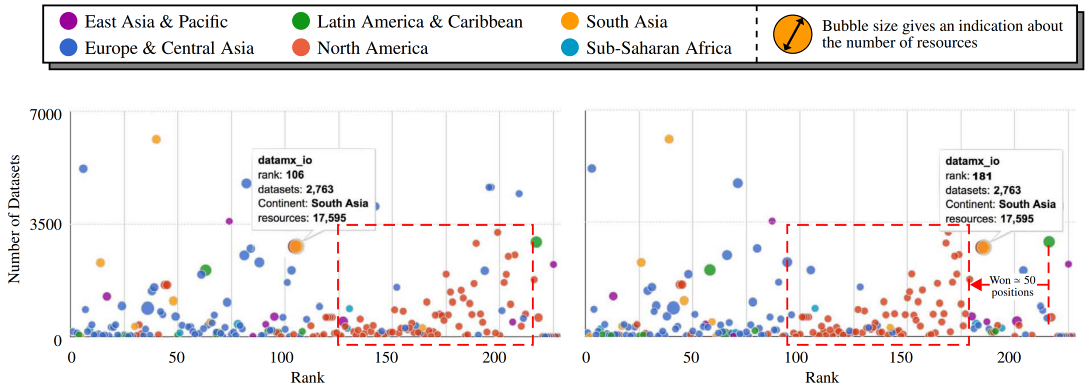


Figure 3.11: Final AHP scores (y-axis) obtained by the 259 available Open Data portals (y-axis) for week 1 (2017) – Criteria of equal importance. Portal number as listed in Appendix A.2.



(a) Equivalence between all (sub)-quality dimensions/metrics

(b) Prioritization of "OpenData" -related dimensions/metrics

Figure 3.12: Evolution of ranking vs. datasets at week 1 (2017) having different user preferences about the importance of criteria.

bring to light how the final portal ranking can be affected by enduser references, we propose to compare the first and second scenarios (i.e., equivalence between criteria vs. prioritization of open data-related metrics) taking a slightly different view in Figure 3.12. Each bubble refers to one specific portal (the bubble's color having been chosen according to the continent where the city portal is located/hosted), the x-axis refers to the portal indices (from 1 to 259), the y-axis to the number of datasets held by each portal for the selected week, and the bubble size to the number of resources (the bigger the bubble, the higher the number of resources). An interesting finding is that, for equivalent preferences (see Figure 3.12 (a)), data portals located in North America occupy the bottom of the rankings (most of them being ranked between 130–220), while the same set of portals won ∼50 positions when prioritizing the Open Data dimensions (see Figure 3.12 (b)). Even though it appears that most of the portals from the other continents remain better, this shows that the licensing on portals that have slipped down the overall rankings is less well managed than the ones located in North America. Overall, the results/rankings must be carefully studied and interpreted depending on the specified preferences.

3.5.3 Portal Evolution over One Year

This section discusses the evolution of data portals both regarding the portal rankings (a portal can win or lose positions from week to week) and the resources held by each portal (datasets and/or resources can be deleted or added on portals).

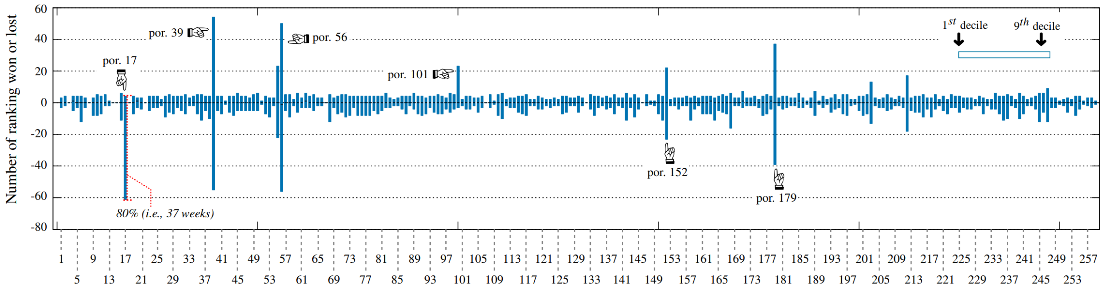


Figure 3.13: Overview of the deviation of portals' ranking from one week to another. Portal number as listed in Appendix A.2.

Figure 3.13 provides an overview of the ranking evolution in the form of a decile boxplot (the 1st and 9th decile being displayed). The x-axis still refers to the portal indices (1 to 259), while the y-axis refers to the number of ranks that each Open Data portal won or lost on a weekly basis. For example, looking at portal 17, in 80% of the cases (i.e., during 37 weeks out of 47) it lost from 1 to 61 positions (see 1st decile's value) and won up to 4 positions (see 3rd decile's value). As a result, the portal lost more than 61 positions during 5 weeks and, similarly, won more than 4 positions during 5 weeks. Although we implemented a mitigation strategy[24] to avoid a "yo-yo" effect when portals become inaccessible from one week to another (i.e., winning and loosing a high number of ranks), we observe that a few portals such as portals 39, 56, and 179 (cf., Figure 3.13) are nonetheless affected by this effect.[25] However, this effect is observed only for 6 portals out of the 259, which does not call into question the findings of our study. After investigation, the deviation of portals 17 and 100 is due to the addition or deletion of datasets/resources. Looking at such deviation patterns can help us to better understand the reasons of an upgrade or downgrade of a portal. Overall, and as a general comment, it can be stated that the ranking of the vast majority of portals does not evolve much (between 1 to 10 positions), which reflects to some extent the fact that governmental organizations do not pay sufficient heed in upgrading their portal's datasets.

[24] We consider the last available values related to all criteria, while downgrading the portal's accessibility dimension.

[25] This is due to the fact that these portals are accessible but no datasets are available for the monitored week (may be due to maintenance operations), thus impacting on the other dimensions and leading to their downgrading in the final ranking.

3.6 CSV Quality & Characteristics

The following section discusses a data corpus consisting of tabular Open Data sources and studies the characteristics and properties of the files from a consumers' point of view. Our metadata reports already showed that the CSV (comma-separated values) format is the predominant format in the Open Data landscape. The main reason is the simplicity and independence of this format: it stores tabular data in plain text where each line of the file is a data record. Each record consists of one or more fields which are separated by a delimiter, typically a comma.

Comma-separated values as a file format has been used for decades for interchanging database information between machines and pre-dates the Open Data initiative. Historically, the CSV format developed out of a need for such an exchange format without an initial formalization; therefore, there are many variations in use and there is not a single, fully specified, "CSV" format. In 2005 the IETF (Internet Engineering Task Force) published a first attempt to standardize CSV [Shafranovich, 2005]; the document defines the associated MIME type "`text/csv`" and proposes a strict dialect and implementation:

- Each record is located on a line, delimited by a line break.
- The use of a header line (appearing as the first line of the file) is optional, however, "the presence or absence of the header line should be indicated via the optional `header` parameter of this MIME type" [Shafranovich, 2005].
- The fields of a record are separated by a comma and each line contains the same number of fields.
- A field may be enclosed in double quotes so that commas and line breaks can be used within fields. If double quotes are used in already double-quotes enclosed fields, these must be escaped by another double quote (e.g., the field "b""b" contains one double quote within the two b's).

However, one can observe many variants of this specification, and nowadays CSV stands more for "character-separated-values". This work studies and analyses the characteristics of 200K CSV files listed in 232 Open Data portals, attributing to a total file size of 413 GB. To the best of our knowledge, this is the first large-scale study of Open Data CSV files. Section 3.6.1 introduces the statistics of our data corpus, detailing the availability of the files per portal and reporting on the file size distribution. Next, Section 3.6.2 details the amount of files that can be parsed as CSV files (using straightforward heuristics to detect delimiter characters, comments and header lines) and reports about typical CSV dialects and table shapes.

3.6.1 Data Corpus

The experiments in this chapter are based on a data corpus collected from 232 Open Data portals, to study the characteristics of Open Data CSV files. The list of data portals and

Table 3.13: Data corpus statistics

	All	CKAN	Socrata	OpenDataSoft
Resources	3 571 085	3 436 288	125 514	9283
Valid http	3 558 823	3 424 227	125 514	9082
Unique	1 995 742	1 898 804	109 650	9082
Labelled as CSV	200 939	185 946	18 275	2166
HTTP 200 OK	141 738	126 776	12 864	2098
Parsed as CSV	104 826	95 249	10 612	1918

Table 3.14: Download statistics.

	OK	NA	Connection	SSL	URI	Sever	Content	IP	Intern
count	141 738	44 838	2350	1386	959	651	613	530	132
	73.36%	23.21%	1.22%	0.72%	0.50%	0.34%	0.32%	0.27%	0.07%

their metadata descriptions was extracted from the Open Data Portal Watch framework (see Section 3.3), using the snapshot of the second week of May 2016.

Corpus statistics: We parsed 950 117 dataset descriptions and extracted 1 995 742 distinct resources. In total, 200 939 unique resources are annotated as a "CSV" files or contain either the tokens "*.csv*" in the URL or the token "csv" in the query parameters (assuming that the URL defines the export format to be CSV). 141 694 files were successfully downloaded of which 104 826 are considered to be CSV files (cf. Section 3.6.2). Table 3.13 additionally groups the corpus statistics by the underlying portal framework.

Download statistics: We tried to downloaded the 200 939 resources and stored the HTTP response header information. In total, we successfully downloaded 141 694 documents (73 %) with a total of 413 GB content. In Table 3.14 we show the distribution of received HTTP Status codes and exceptions which are grouped into typical error classes. A set of 44 838 (23 %) documents are not available any more for download, more details about this in the following.

The majority of the downloaded content has a per-file size of less than 100 kB, the biggest file had a file size of 25 GB (cf. Figure 3.14).

Table 3.15 provides an overview of the reported content-types in the HTTP Response headers with status code 200, which were identified to contain actual CSV content. Interestingly, we see that the header content-type cannot really be considered as an indication if the file is in CSV format.[26]

[26] Note, that in this work we did not decompress zipped files (e.g., application/zip content-type), which could potentially contain CSVs.

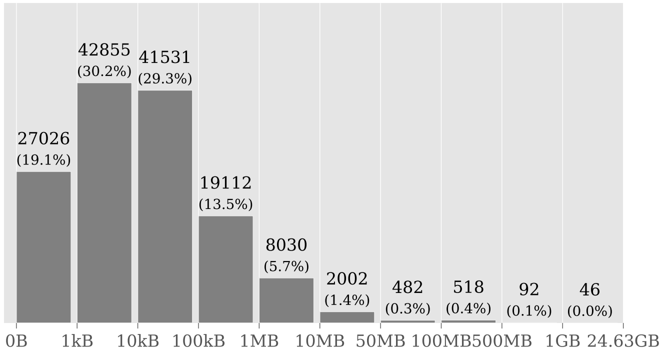


Figure 3.14: File size distribution

Table 3.15: Header content-type distribution

№	MIME-TYPE	COUNT	
1	application/octet-stream	63 350	(44.71 %)
2	text/csv	49 451	(34.90 %)
3	application/zip	9636	(6.80 %)
4	text/html	8118	(5.73 %)
5	text/plain	2825	(1.99 %)
6	application/csv	2453	(1.73 %)
7	application/vnd.ms-excel	2423	(1.71 %)
8	text/x-comma-separated-values	752	(0.53 %)
9	text/xml	582	(0.41 %)
10	application/x-zip-compressed	496	(0.35 %)

Domain statistics: Next we group the 200 939 extracted resource URIs (which were labelled as CSVs) by their domain and analyzed the percentage of retrievable resources per domain. In total we discovered 2301 unique domains. The 10 domains with the most resources are listed in Table 3.16 together with their ratio of retrievable resources. To our surprise, we see three domains with the retrievability ratio of less than 10 percent. To get the full overview, we plotted the retrievability for each of the 2301 domains in Figure 3.15. Another view is provided as a dot plot in Figure 3.16. The domains are binned by their availability ratio (y-axis). Next, for each bin, we ordered the domains by their size and split them into 10 equal size groups (ten dots on the x-axis). Each dot represents the average size of 1/10 of the domain for that bin. The two plots show that there exists a couple of smaller domains for which the files are not available, but also some larger domains (e.g. the dot representing ten domains with an average availability of 0.4 -0.5 and an average document number of 3000, cf. Figure 3.16).

Table 3.16: Top-10 domains and their retrievability ratio.

№	DOMAIN	COUNT	RETR
1	cdn1.sdlabs.ru	44 558.0	(100.00 %)
2	ec.europa.eu	10 075.0	(0.04 %)
3	www20.statcan.gc.ca	7946.0	(99.87 %)
4	www.gov.uk	6108.0	(49.31 %)
5	www.e-stat.go.jp	6067.0	(0.00 %)
6	opendata.socrata.com	5544.0	(98.70 %)
7	webarchive.nationalarchives.gov.uk	3700.0	(96.05 %)
8	www.landesdatenbank.nrw.de	2585.0	(75.51 %)
9	aimis-simia.agr.gc.ca	2489.0	(100.00 %)
10	www.statistik.sachsen.de	2289.0	(11.93 %)

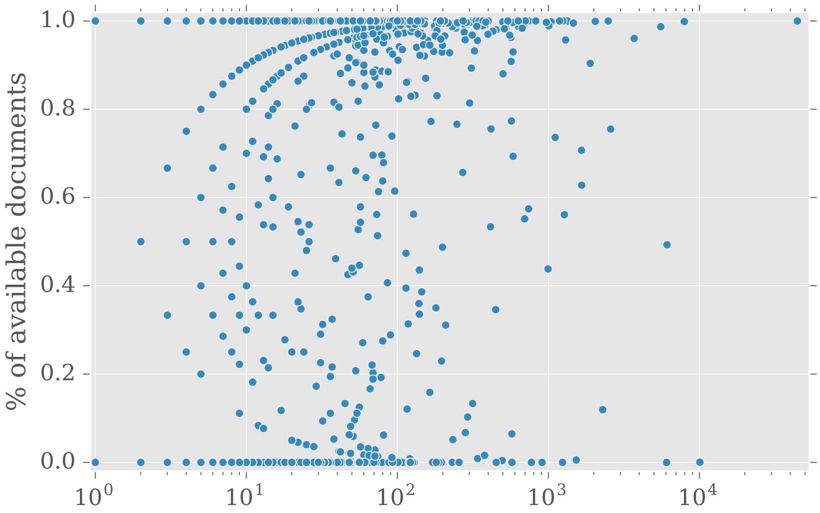


Figure 3.15: Retrievability of resources on domains; each dot represents a domain and the horizontal axis gives the number of files per domain.

3.6.2 Parsed CSV Files

This section reports on the challenges in parsing our corpus using standard Python libraries. As a first step, we tried to identify the correct encoding of the file using the file-magic library (which uses the underlying Unix file library).[27] Next, we assumed the properly encoded content is CSV-like and guessed the delimiter, line ending and used

[27] http://pubs.opengroup.org/onlinepubs/9699919799/utilities/file.html, last accessed 03/07/2019

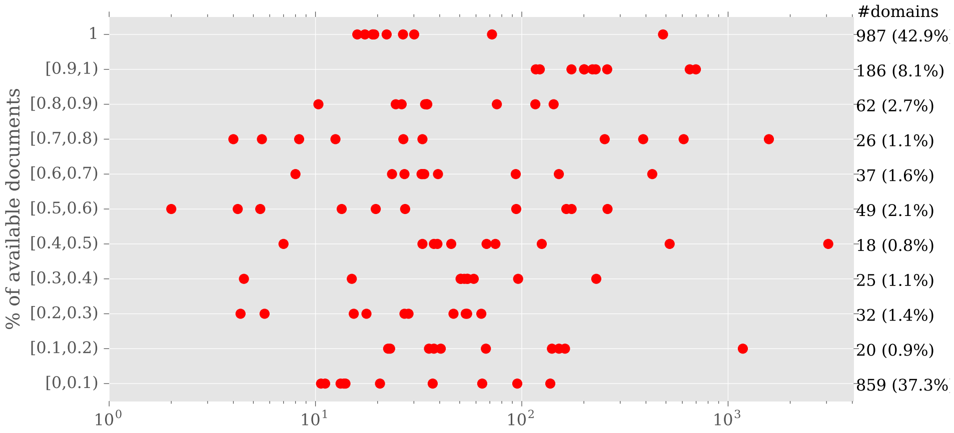


Figure 3.16: Retrievability of resources on domains as dot-plot. Each dot represents the average over 1/10 of the total elements in the bin; the horizontal axis gives the number of files per domain.

quotation char using the heuristics in the standard CSV library.[28]

In order to properly detect the shape of tables in CSV documents we checked if the documents contained any preceding comment lines or consisted of multiple tables. For instance, consider Table 3.17 (found on an Italian Open Data portal[29]). Table 3.17 displays a CSV document with leading comment lines (typically indicated by rows where only the first field contains a value). It also consists of multiple tables within a single CSV file: the document holds three tables which are separated by repeated line breaks. We also observed variations of multi-tables where the number of delimiters remain consistent for each line (i.e., the tables are separated by empty records/cell values).

Comment lines, multi-tables and multiple header rows potentially occur in documents produced by spreadsheet applications such as Microsoft Excel. With the help of such a tool it is easy to see if there are multiple tables within one sheet; cf. Table 3.18 which displays the content of Table 3.17 in Microsoft Excel. Next, we discuss our heuristics to detect preceding comment rows, multiple header rows, and multiple tables in a single CSV document.

Header detection heuristic: Typically, CSV files provide header rows to describe a specific column and increase the readability of the content. However, even in the RFC specification [Shafranovich, 2005] the use of an header row is optional. In certain cases we observed multiple header rows in a document. In particular, we observed this when exporting a spreadsheet with merged cells to CSV (see for instance, line 3 in Table 3.18).

[28] `https://docs.python.org/2/library/csv.html#csv.Sniffer`, last accessed 03/07/2019

[29] `http://dati.veneto.it/dataset/fb289cb7-ba04-4eb8-a2cf-11a8a0fb0e3c/resource/72e9df90-a114-40a7-9889-b907064e2e39/download/c0201030concinquinantecoinfo.csv`, last accessed 06/06/2017

Table 3.17: Example CSV document.

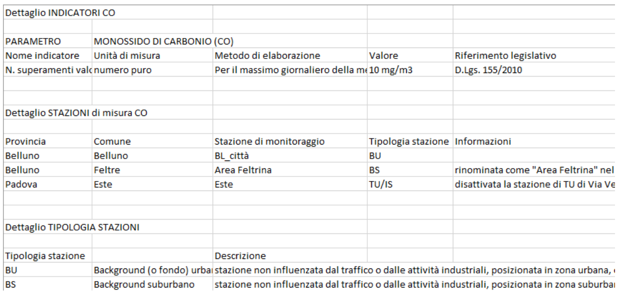
```
Dettaglio INDICATORI CO;;;;;

PARAMETRO;MONOSSIDO DI CARBONIO (CO);;;;
Nome indicatore;Unita di misura;Metodo...
N. superamenti valore limite protezion...

Dettaglio STAZIONI di misura CO;;;;;

Provincia;Comune;Stazione di monitorag...
Belluno;Belluno;BL_citta;BU;;
Belluno;Feltre;Area Feltrina;BS;"rinom...
Padova;Este;Este;TU/IS;disattivata la ...
```

Table 3.18: Viewed with spreadsheet.



Dettaglio INDICATORI CO				
PARAMETRO	MONOSSIDO DI CARBONIO (CO)			
Nome indicatore	Unità di misura	Metodo di elaborazione	Valore	Riferimento legislativo
N. superamenti valc	numero puro	Per il massimo giornaliero della m	10 mg/m3	D.Lgs. 155/2010
Dettaglio STAZIONI di misura CO				
Provincia	Comune	Stazione di monitoraggio	Tipologia stazione	Informazioni
Belluno	Belluno	BL_città	BU	
Belluno	Feltre	Area Feltrina	BS	rinominata come "Area Feltrina" nel
Padova	Este	Este	TU/IS	disattivata la stazione di TU di Via Ve
Dettaglio TIPOLOGIA STAZIONI				
Tipologia stazione		Descrizione		
BU	Background (o fondo) urba	stazione non influenzata dal traffico o dalle attività industriali, posizionata in zona urbana,		
BS	Background suburbano	stazione non influenzata dal traffico o dalle attività industriali, posizionata in zona suburba		

An absolutely accurate detection of header rows is impossible, due to the lack of syntactic description of CSVs. In order to get an idea how the CSV format is used in terms of missing headers and multi-headers, we implemented a simple heuristic header detection algorithm:

- If there are any numeric values in the first row we assume no header in respective CSV.[30]

- If the first two rows consist of strings only, followed by some numeric cell values in the next rows, we assume a (two row) multi-header.

- Else, we consider the first row as the default header. This is for instance the case if all cells in all the first rows consist of string values.

Multi-table and comment line detection heuristic: Similar simple heuristics are used to detect comment lines and multi-tables. The algorithm considers as comment lines all lines at the beginning of a file with zero or one delimiter. This heuristic was developed based on a manual inspection of randomly selected CSV files.

Regarding multi-table detection, the developed heuristic first parses the files and builds groups for consecutive lines with the same column number (e.g., ($rows$,$cols$)). For instance, a table with 20 rows and 10 columns would be represented as one single group (20,10), while a multi-table with the first table having 20 rows and 10 columns and the second table having 10 rows and 5 columns would result in the groups ((20,10), (10,5)). The algorithm considers a CSV file containing multiple tables if there exists more than one group with more than one row and different cell/column numbers. We limit the number of possible tables in a document to three.

[30] Obviously, there are cases where numeric values in the first row can make sense: for instance, consider a transposed table which holds the header values in the first column instead of the first row, or a table describing data for different years, using the years as headers. However, with this pragmatic approach we aim for an indicator for the use of header rows.

3.6.3 Corpus Results

In order to convert our corpus of CSV files to a consistent representation, i.e., a single header table of regular shape without comment lines, we applied the introduced heuristics. Out of a total of 141 738 files, which are marked as CSV in the metadata, we successfully parsed 104 826 CSV documents. As possible delimiters we allowed the following characters: , \t ; # : | ˆ . The Python standard CSV library also uses a single whitespace character as a possible delimiter. We excluded this choice since it is an extremely uncommon delimiter with an high rate of incorrectly guessed files. Table 3.19 shows the results of this parsing process in which 73.9 % of the downloaded files could be successfully parsed as a table (or multi-table with less than 5 tables). The majority of the valid CSV files contain a single table. The main parsing error was that the delimiter could not be detected, followed by compressed (zip) files, which we did not handle and parse in this study.

Table 3.19: Overall parse process statistics

	COUNT	
parsed	141 738	100 %
without errors	104 826	73.9 %
single tables	102 210	97.5 %
two tables	2279	2.2 %
three tables	337	0.3 %
with errors	36 912	26.1 %
ignored (zip files)	9925	26.88 %
too many tables	1717	4.6 %
no delimiter	19 511	52.8 %
others	5759	15.6 %

CSV dialect: Table 3.20 shows the distribution of the detected delimiter in total and grouped by the underlying portal software. The most common delimiter symbol is the comma ($\sim$ 70%) followed by the semi-colon ($\sim$ 8%). In addition to these two delimiters there are also 1153 tab-separated-value files and only a minute proportion of files using other delimiter symbols (e.g., | and #).

The OpenDataSoft software integrates and displays tabular data in the framework and allows the export of these tables in different formats. Surprisingly, the OpenDataSoft framework mainly returns semicolon-separated files when exporting CSV. A possible reason is that OpenDataSoft is mainly deployed in France (7 out of 11 monitored portals) where semicolon is the commonly used delimiter (e.g., Excel saves a spreadsheet as semicolon separated file under French location settings).

In order to further look into the deviating use of delimiter in different countries we

Table 3.20: Distribution of delimiter

	All	CKAN	Socrata	OpenDataSoft
,	96 580	83 063	13 515	2
;	11 011	9123	1	**1887**
\t	1153	1152	1	-
:	251	202	27	22
|	194	194	-	-
#	62	61	1	-
ˆ	3	3	-	-

grouped the portals by their origin location. In Table 3.21 we list the top 5 countries and their use of comma and semicolon delimiters. Beside the French portals also in German and Austrian portals there are more semicolon-separated files. In principle, it can be assumed that this is highly influenced by the use of comma as the decimal mark.

Table 3.21: Use of semicolon vs. comma in top-5 countries

	comma (,)	**semicolon (;)**
RUS	45 181	175
GBR	17 590	8
USA	16 331	148
DEU	377	4551
AUS	3655	10

Multi-tables & comment lines: By applying our multi-table detection algorithm we observed 102 210 CSV files containing a single tables, 2279 files with two and 337 with three tables (cf. Table 3.19), which results in a total of 107 779 tables.

In Table 3.22 we list the tables with a certain number of detected comment lines and header rows. As the results show, the majority of the tables have no comment line and one header row. Around 11k documents have no detectable header row, 92 970 have one header row and 3002 tables contain two header rows according to our heuristic.

Columns-Rows shape: Table 3.23 contains the descriptive statistics for the row and column counts (excluding tables with more rows/columns than the 95% quantile). This covers 88% of the tables. We can see that the average Open Data CSV table has around 379 rows and 14 columns. Figure 3.17 shows the distribution of the tables for various row/column shapes (rows are binned). Surprisingly, we see a large number of tables with exactly one 1 row and different columns. A manual inspection of randomly selected

Table 3.22: Comment and header rows

	HEADER ROWS		
COMMENT LINES	0	1	2
0	11 065	86 846	1251
1	279	3565	638
2	66	706	105
3	26	231	27
4	78	352	108
5	293	1270	873

files with 1 row indicates that these are exports from Socrata portals with test data.[31]

Table 3.23: Statistics about number of rows and columns (max 95% quantile)

Statistics	COUNT	MEAN	STD	MIN	25%	50%	75%	MAX
rows	94 958	379	1091	1	5	25	146	8684
columns	94 958	14	7	2	8	19	20	26

Readability of headers: In order to get better insights into the readability of the header values we analyzed the structure and composition of the CSVs' headers. The results in Table 3.24 are based on 92 970 CSVs where we detected a single header row, consisting of a total of 1.7M values. In Table 3.24 we distinguish between values consisting of *multiple* words, values containing *underscores*, values written in *camel case* and the remaining values, which we assume consist of a *single* word.

Table 3.24: Composition of header values.

Header	Count	
Total (single row header)	1 735 807	
Underscore	707 558	(40.7%)
Single word	578 088	(33.3%)
Multiple words	302 474	(17.4%)
Camel Case	147 687	(8.5%)
WordNet Mapping	186 531	(10.7%)

Interestingly, about 50% of the inspected header values (855k) were composed of camel case and underscore separated words. This fact suggests that many of these headers

[31] To give an example: One of those "test data" files is `https://performance.smcgov.org/api/views/mncj-7pjs/rows.csv?accesstype=download`

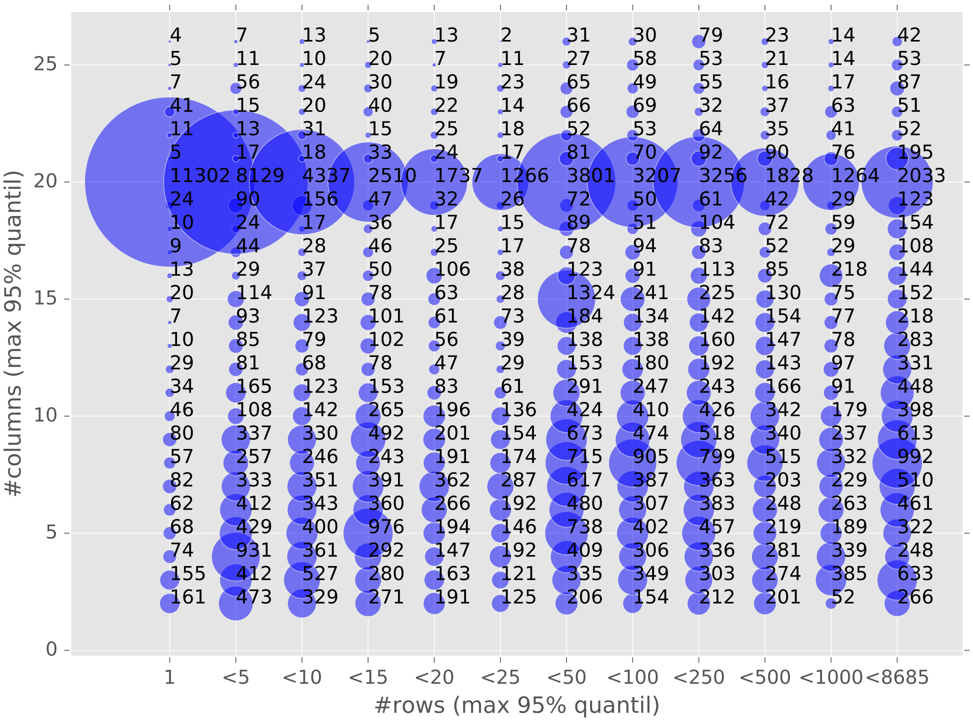


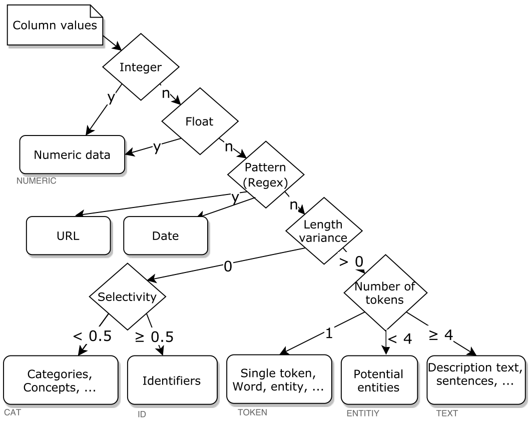
Figure 3.17: Row (binned) vs. column count, including number of tables

spring from dumps of relational databases and therefore some of the headers might also exist as labels in other tables. However, this stringing together of words impedes an automated mapping and linkage to potential entities and concepts.

Additionally, we experimentally investigated the readability of the headers by a simple mapping of the values to WordNet [Miller, 1995b], a lexical database of English words grouped into sets of synonyms. Prior to the WordNet lookup we split the headers on underscores and camel case and apply an automated stemming (i.e., reducing words to their word stem). This resulted in 186 531 mappings ($\sim$ 11%, cf. Table 3.24).

Column Types: To get an idea of the distribution of general data types in the columns, we applied a basic type classification heuristic, cf. Figure 3.18a:

- We check if all values of a column are integers or decimals (i.e., floating point numbers), including the comma as a decimal mark, and classify the column as `NUMERIC`.
- We use a regular expression to detect valid `URLs`.



(a) Column type heuristic

NUMERIC	687 896
ID	626 568
TOKEN	411 976
ENTITY	327 223
TEXT	158 956
DATE	62 589
URL	40 586

(b) Column type distribution

Figure 3.18: Column classification heuristic and results over CSV corpus.

- We parse for DATE values using the Python dateutil package.[32]
- If a column does not consist of URLs or dates we compute the length variance of the string values, i.e., the varying length of the values. If there is no variance, we additionally consider the selectivity of the values. In case the values are rather unique (we use a selectivity of 0.5 as threshold) we classify the column as ID-column (identifiers).
- Eventually, if we observe higher length variance of the values, we distinguish the types TOKEN, ENTITY, and TEXT based on the number of words. Here, we use 4 words as a threshold to consider a column as "potentially entities"-column or as a column holding description text and sentences.

Figure 3.18b lists the results of this column classification heuristic over our corpus. Surprisingly, we see that the majority of the columns are containing either numeric or id data.

3.7 Related Work

Data quality assessment and improvement methodologies are widely used in various research areas such as relational databases, data warehouses, information or process management systems [Strong et al., 1997, Jarke and Vassiliou, 1997], but also to assess the quality of Linked Open Data. To gain a deeper insight into current approaches for assessing the data quality of Linked Open Data, we refer to the work by Zaveri et al. [Zaveri et al., 2015], which provides a comprehensive literature survey. We note, however, that the focus of this survey differs from our work in the sense that the quality dimensions

[32] https://pypi.python.org/pypi/python-dateutil, last accessed 30/04/2019

cover Linked Data specifically, rather than Open (Government) Data in general. For instance, the transparency and openness aspects of our work are not fully covered by [Zaveri et al., 2015].

Over times, different application and research areas established various measures and techniques to assess the quality of data and services and for keeping up with the increasing complexity of the tasks [Zhu et al., 2012]. Batini et al. [Batini et al., 2009] published a detailed and systematic description of methodologies to assess and improve data quality. Generally, the different methodologies involve various phases starting from the definition of quality metrics, the measurement, an analysis phase and possible improvements with small differences how feedback loops are integrated.

3.7.1 Related Efforts on Metadata Quality Assessment

Pipino et al. [Pipino et al., 2002] discuss a set of data quality dimensions and their subjective- and objectiveness on a very general level. Similarly, in [Wang and Strong, 1996] the authors provide two comprehensive surveys: a survey of data quality attributes and a survey of data quality dimension. Wang et al. grouped the dimensions into four different information quality aspect and built a conceptual framework (i.e., a hierarchy) of data quality: (i) intrinsic, (ii) contextual, (iii) representational, and (iv) accessibility. For instance, the intrinsic quality aspect holds the "believability", "accuracy", "objectivity" and "reputation" dimensions. Michnik and Lo [Michnik and Lo, 2009] further refine and extend the four-aspect approach of [Wang and Strong, 1996], e.g., by introducing sub-categories. While these efforts discuss quality dimensions on a very general level, we discuss the concrete computation and automated assessment of quality metrics based on the DCAT metadata schema.

In contrast to our approach, in [Margaritopoulos et al., 2008] the authors present the application of "logic rules" to assess the quality of metadata. They identify three different types of rules: *rules of inclusion*, *rules of imposition*, and *rules of restriction*. The definition of these rules is based on dependencies and relations of resources. For instance, applying a *rule of restriction* to a resource's metadata field means that the values "[...] must include the values of the same metadata field or records of related resources".

This rule-based approach can be considered as an automated metadata quality assessment. While there are already various approaches of automated metadata quality evaluation [Hughes and Kamat, 2005, Najjar et al., 2003], also by using simple statistical evaluations [Greenberg et al., 2001, Moen et al., 1998, Wilson, 2007], a manual evaluation is often unavoidable and therefore very common [Greenberg et al., 2001, Moen et al., 1998, Wilson, 2007].

Regarding quality assessment within the 5S model (cf. Section 3.1) Gonçalves et al. [Gonçalves et al., 2007] discuss quality dimensions and measures for each of the 5S main concepts. Further, the authors provide an example evaluation of the dimensions and discuss the practical utility of the proposed quality model. In relation, Moreira et al. [Moreira et al., 2007] presented 5SQual, a quality assessment tool built upon the 5S model which automatically assesses and evaluates eight different quality dimensions. While these quality assessment approaches focus on the concepts of the 5S model, we focus in our work on the quality of metadata descriptions in data catalogs.

3.7.2 Related Work on Open Data Quality

When looking into related work on data quality, to the best of our knowledge, not much work is published for QA in Open Data. However, in recent years, several projects addressed the Open Data domain and we identify projects which deal with the quality of Open Government Data (aligned to Barack Obama's Open Government Directive [Orszag, 2009]) and aim to assess and compare the state of Open Data across different countries [World Wide Web Foundation, 2015, Bertot et al., 2012]. Further, we identify recent projects which try to assess the progress and spreading of Open Data in the private sector, e.g. the Open Company Data Index,[33] a report by the World Bank Institute which aims to register open corporate data and provides an aggregated quality assessment per country. In the following we highlight projects which propose various metrics to evaluate the (meta-)data quality within Open Data catalogues.

In relation to data quality assessment in Open Government Data catalogues, such as `data.gov` or `data.gv.at`, recent work by Kučera et al. [Kucera et al., 2013] discusses quality dimensions and requirements of such catalogues. The authors list and discuss relevant quality dimensions (*Accuracy*, *Completeness*, *Consistency* and *Timeliness*) but, unfortunately, the work is short of detail in some respects.

More related to the actual data quality assessment is the Global Open Data Index project[34] and the Open Data Monitor project.[35] Both projects define a set of Open Data specific quality metrics and rank various countries by their state of Open Data. However, while the Global Open Data Index is based on a manual expert evaluation, the Open Data Monitor project mainly uses dimensions which can be assessed automatically (e.g., the completeness of metadata). In principle, to the best of our knowledge, all of the above-mentioned efforts either rely on a manual evaluation of their quality dimensions and therefore do not provide an automated assessment as we do with our framework; or the projects do not deal with heterogeneous metadata and therefore do not provide a generic and large-scale quality analysis of metadata Open Data portals.

Complementary to the quality assessment approaches, the OPQUAST project [de Dona et al., 2012] proposes a checklist for publishing Open Data, including questions related to quality aspects. This checklist is very extensive and the questions reach from general questions about the data catalog (e.g., "*The concept of Open Data is explained*") to in-detail questions about specific metadata keys and available meta-information (e.g., "*It is possible to obtain information regarding the level of trust accorded to the data*").

[33] `http://registries.opencorporates.com/`, last accessed 2019-10-30

[34] `http://index.okfn.org`, last accessed 2019-10-30

[35] `http://www.opendatamonitor.eu/frontend/web/index.php`, last accessed 2019-10-30

Most closely related to the quality assessment efforts in this thesis are [Ochoa and Duval, 2009, Braunschweig et al., 2012, Reiche et al., 2014]. The authors discuss a set of quality metrics for metadata in digital repositories, including a detailed description, definition and evaluation of the metrics. [Reiche et al., 2014] also identified the need for an automatic quality assessment and monitoring framework to better understand quality issues in Open Data portals and to study the impact of improvement methods over time. The authors developed a prototype of such a framework which is unfortunately now offline.[36]

Although [Reiche et al., 2014] influenced the herein presented metrics and framework, we extended the work of Reiche et al. in terms of generalised and useful quality metrics in the context of Open Data (e.g., by adding a contactability and open format metric), in terms of the extent of monitored data portals and in terms of a continuous monitoring of these portals.

3.7.3 Alternative Efforts on Modeling Digital Catalogs

Various efforts already exist to study the formal theory of digital libraries. On the one hand, there is most prominently the 5S model, which we already mentioned in Section 3.1. In contrast to the 5S model, the DELOS Reference Model [Agosti et al., 2006, Candela et al., 2007] models a digital library by using the following six main concepts: content, user, functionality, quality, policy, and architecture. The DELOS model is formulated as an entity-relationship model and the structure is mainly hierarchical. The aforementioned concepts represent high level containers, e.g. the "content" concept holds the *Resource* entity and the "quality" concept holds the *Quality Parameter* entity and these two entities are related: a quality parameter is evaluated on a resource. [Agosti et al., 2007] describes and compares the DELOS Reference Model to the 5S Framework. In particular, it compares the quality aspects for the 5S model described in [Gonçalves et al., 2007] with the quality aspects of the DELOS model [Candela et al., 2007].

In [Ferro and Silvello, 2013] the authors propose "NESTOR", a formal model for digital archives. The formally defined model is based on nested sets, where subset relations correspond to different hierarchies within an archive. Further, the authors use the proposed model to map and extend the 5S model [Gonçalves et al., 2004].

Regarding existing efforts towards homogenised metadata descriptions for data catalogs, Assaf et al. [Assaf et al., 2015] propose HDL, an harmonised dataset model. HDL is mainly based on a set of frequent CKAN keys. On this basis, the authors define mappings from other metadata schemas, including Socrata, DCAT and Schema.org. Our work in Section 3.2 is partially influenced by and based on the work by Assaf et al. [Assaf et al., 2015].

[36] `http://metadata-census.com`, last accessed 2015-03-06

3.7.4 Related Studies on Analysing Corpora of Tabular Data

Most existing work on analysis and statistics over corpora of tabular data use HTML/Web tables which are extracted by crawling the Web or a specific domain (e.g., tables found on Wikipedia) [Ritze et al., 2015, Ritze et al., 2016, Crestan and Pantel, 2011, Wang et al., 2012, Hassanzadeh et al., 2015, Lehmberg et al., 2016, Eberius et al., 2015b].

The largest attempt regarding the analysis of tabular data is to the best of our knowledge the Web Data Commons (WDC) project.[37] It presents statistics of 233 million Web tables with a total size of 165 Gigabyte. In order to generate these statistics, the project tried to detect the orientation of these tables, the header rows and entity columns, and extracts some context data of the tables. Ritze et al. [Ritze et al., 2015, Ritze et al., 2016] use the WDC Web tables corpus in order to run additional analysis and to explore the potential of linking this corpus to the DBpedia knowledge base.

Crestan and Pantel [Crestan and Pantel, 2011] performed a large scale analysis of Web tables on a crawl of the Web and proposed a table type taxonomy, e.g., HTML formatting table, horizontal and vertical listings of entities, or enumerations. This automated taxonomy uses HTML tags and "Lexical features" for the categorization and therefore cannot be applied to CSV syntax.

Wang et al. [Wang et al., 2012] present approaches to "understand" Web tables in terms of schema and entities. To this end, the paper describes an header detection (based on HTML tags and formatting) and entity column detection algorithm. Again, these algorithms are tailored to Web tables and only partially applicable to CSVs.

Related to our analysis, in [Ermilov et al., 2013] the authors formalize a canonical form of tabular data consisting of a single header row and corresponding data rows and define three deviation levels from this canonical form: table level (e.g., metadata/comments are embedded), header level (e.g., header missing), data level (e.g., empty cells or rows). Similar to our data, the underlying corpus for the analysis of [Ermilov et al., 2013] is tabular Open Data (100 randomly selected CSVs).

A complementary work – also related to our analysis – is the large-scale study by Chen and Cafarella on Microsoft Excel files [Chen and Cafarella, 2013]: the authors obtained over 400k spreadsheets from 50k distinct domains from a publicly available Web crawl. This work focuses on extracting the relational data of spreadsheets, by applying several preprocessing and extraction steps. The authors particularly focus on extracting hierarchical attributes, i.e. spreadsheets with a second attribute row (or column) which relates to a group of sub-attributes. While the authors showed that these hierarchical attributes are very common in Excel spreadsheets (32.5% of an inspected sample [Chen and Cafarella, 2013]), we observed multi-header-rows (which are potential hierarchical attributes) in only 3% of the files. Eventually, based on the results of [Chen and Cafarella, 2013] we gain the following two insights: (i) Excel spreadsheets are often of small and manageable size, designed for human consumption only, while our report shows that CSVs range from

[37] http://webdatacommons.org/webtables/2015/relationalStatistics.html

manually created tables of small size for human consumption, to (very) large tables such as database exports or sensor measurements; (ii) the methods for extracting attribute hierarchies are applicable to a small subset of our corpus, i.e. the multi-header CSVs – which in fact are most likely just CSV exports of Excel sheets. Therefore, the approach by Chen and Cafarella is clearly related and should be considered in future CSV analyses.

3.8 Critical Discussion and Future Directions

There are metadata quality issues that could disrupt the success of Open Data: inadequate descriptions or classifications of datasets directly affect the usability and searchability of resources. In this chapter we have proposed a set of objective quality metrics (based on the W3C metadata schema DCAT) to monitor the quality of Open Data portals in a generic and automated way to assess the severity of these issues. Moreover, we have introduced a generic abstraction of web-based data portals for the purpose of integrating a large amount of existing data portals in an extensible manner. Further, we have implemented and deployed an Open Data portal monitoring and quality assessment framework – the "Open Data Portal Watch" platform[38] – that monitors our metrics in weekly snapshots of the metadata from over 261 Open Data portals. Our core findings and conclusions of this monitoring can be summarised as follows:

- Slightly alarming, we were able to perform HTTP HEAD lookups on only 78% of the resource URLs without any errors or restrictions – which seems to indicate deficits even in terms of proper implementation of standard HTTP features.
- We observed that there is a gap between the common definition (i.e., the Open Definition[39]) and the actual state of Open Data regarding the use of machine-readable open formats and the existence of license specifications and compliance to actual open licenses.
- The majority of the datasets do not provide machine-readable contact information (e.g. in the form of a valid email address or URLs): missing provenance information – in our opinion – involves the risk of intransparency and impaired usefulness of datasets and would allow valuable user feedback.

To get a more in-depth look at the quality and characteristics of the actual resources at these portals, also a corpus of 200k tabular Open Data resources has been analysed. Our experiments highlight some very specific characteristics of tabular Open Data and the resulting challenges for an automated processing. The core findings of this comprehensive CSV analysis can be summarised as follows:

[38] http://data.wu.ac.at/portalwatch/

[39] http://opendefinition.org/, last accessed 2019-06-24

- 200k (10 %) of the resources are labelled as CSV, of which only 100k files can be actually parsed.
- Only 50 % of the actual CSV files specify the correct format in the HTTP response header.
- Only 10 % of the header values in single header tables could be mapped to entries in the English WordNet dictionary.
- The majority of the CSV files use the correct comma (,) delimiter.
- An average CSV Open Data file contains 365 rows and 14 columns. However, 10 % of the tables have only one row, possibly indicating that these are dummy/test tables.
- 50 % of the columns in our tables contain either numerical values or IDs (same length values, low selectivity).

Technical problems arising from different formats or different CSV dialects can usually be solved by application-specific programming; however, the cost is often prohibitive. Adherence to a few (CSV publishing) rules would improve the accessibility of Open Data and allow for easier knowledge discovery and analysis, which after all is one of the main purposes of Open Data.

In later parts of this thesis – particularly in Chapter 5 when we focus on semantic labelling algorithms – we perform large-scale, automated, processing of CSVs. In the following we want to summarize the main challenges which complicate this process:

Number of columns: CSV files with much more than a few dozen columns become very difficult to handle in an exploratory mode of analysis.

Number of header lines: Interactively, any number of header lines can be dealt with easily, but for an automated processing a variable number of header lines can be challenging to interpret and annotate.

Choice of names in headers: Ideally the headers would provide meaningful descriptions of the content of the data columns.

Multiple tables in one file: This adds an additional level of complexity without appreciable benefit; putting each table into a separate CSV file facilitates access to the data.

The presented report on the quality and characteristics of CSV files is the largest study of this kind. An interesting future direction would be an empirical study on how the observed insights and challenges practically impact the data consumers, and to draw some explicit conclusions about what the findings mean for the data consumers.

Overall, our analysis reveals a number of insights into the shapes and content of Open Data portals, however, we also identified a number of shortcomings in our approach which potentially hint at future work: Currently, our Open Data Portal Watch framework monitors and archives all metadata descriptions from a list of data portals; we have downloaded a corpus of all CSV files for our experiments, but at the moment we cannot guarantee full downloads of all files and sizes in a regular manner.

As for future work, we prioritize (i) the evaluation of more portals and datasets, including other software frameworks and HTML-based data portals, and portals from other domains such as the Data Science portal *Kaggle*,[40] (ii) an efficient monitoring of the actual resource content for the evaluation of metadata accuracy and more extensive corpus characteristics, and finally, (iii) further refinement of the openness metrics regarding various licenses and formats.

A complementary use of our monitoring system is the (semi-)automated generation and correction of missing and wrong metadata, e.g., by suggesting values for certain missing metadata fields, or by automatically checking the consistency of existing fields in comparison to actual values. In the course of the ADEQUATe project[41] we integrated such a tool into the Austrian data portal `data.gv.at`: It reports back any data quality issues to the data publisher and even provides improved metadata descriptions at the portal. We will focus on such an improvements and enrichments of metadata in the next chapter.

[40] `https://www.kaggle.com/`

[41] `https://adequate.at/`

CHAPTER 4

Lifting Data Portals to the Web of Data

In the previous section we showed the issue of heterogeneous metadata schemata across existing data portals and catalogs. Also the W3C identified this problem and proposed an RDF vocabulary: The metadata standard DCAT [Maali and Erickson, 2014] (Data Catalog Vocabulary) describes data catalogs and corresponding datasets. It models the datasets and their distributions (i.e., the actual published data in different formats) and re-uses various existing vocabularies such as Dublin Core terms, and the SKOS vocabulary.

However, currently only a limited number of (governmental) Open Data portals use this DCAT standard as their metadata schema. Further, DCAT is not always directly applicable to the specific schema extensions and dataset publishing practices deployed by particular data portals. For instance, while DCAT describes the *distributions* of datasets as different downloadable representations of the same content (i.e., the dataset in different file formats), we observe in CKAN data portals various different aggregations of datasets: a dataset might be grouped by dimensions such as spatial divisions (e.g., data for districts of cities) or grouped by temporal aspects (e.g., same content for different years). This means that for certain datasets a mapping to the DCAT vocabulary is not straightforwardly possible and some extensions might be needed to accommodate for such common practices.

In order to tackle the aim of better integration of datasets on several fronts, we describe in this chapter how to expose the metadata descriptions as Linked Data in an homogenized representation, and how to we can automatically improve the descriptions of the actual data.

While mapping data portals' metadata to DCAT – as described in Section 3.2 – is certainly a step towards a better interlinking of open datasets, the major search engines do not yet integrate this information for enriching their search results. To enable an integration of data catalogs into the knowledge graphs of search engines (such as Google's Knowledge Graph[1]) we further discuss how to publish the DCAT metadata descriptions using Schema.org's Dataset vocabulary.[2] In fact, Google's recent Dataset Search project [Brickley et al., 2019] harvests our published Schema.org descriptions and can be found via their search interface.[3]

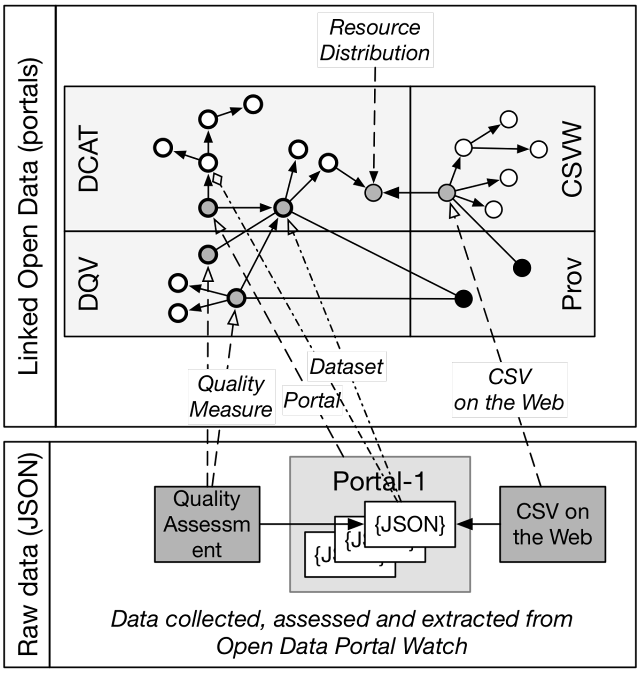


Figure 4.1: The collected dataset descriptions get re-exposed and enriched by quality measures and additional metadata.

Figure 4.1 displays the different vocabularies and components that we discuss in this chapter. The collection of the dataset descriptions (as JSON documents) and mapping to the DCAT vocabulary is already described in the previous Chapter 3. The framework computes a quality assessment for each dataset, and (in case of CSV resources) additional CSV metadata. We model this additional information using W3C vocabularies (DQV [Debattista et al., 2016] and CSVW [Pollock et al., 2015]) and connect the data to the DCAT [Maali and Erickson, 2014] representations. Since all this data is automatically generated by our framework, we also add provenance information to the dataset description, the quality measurements and the CSV metadata.

Overall, in this chapter we present the following concrete contributions:

[1] `https://developers.google.com/knowledge-graph/`, last accessed 28/05/2019

[2] `https://schema.org/Dataset`

[3] `https://toolbox.google.com/datasetsearch`, last accessed 28/05/2019

(i) In Section 4.1 we describe *the process of re-exposing the data* collected by the Open Data Portal Watch framework (see Section 3.3): The output formats include a subset of W3C's DCAT with extensions and Schema.org's Dataset-oriented vocabulary.

(ii) Further, we present in Section 4.1.1 an analysis of the exposed metadata reporting which metadata descriptions can be easily/straightforwardly mapped (and which not, respectively). We illustrate *issues and challenges when mapping CKAN metadata* to the DCAT vocabulary, and highlight potential *design issues/improvements in the target vocabularies* (i.e., W3C's DCAT and Schema.org's Dataset vocabulary).

(iii) We *enrich the integrated metadata by the quality measurements* of the Portal Watch framework available as RDF data using the Data Quality Vocabulary (Section 4.2).

(iv) We present heuristics to further *enrich descriptions of tabular data by auto-generating additional metadata* using the W3C CSV on the Web vocabulary (Section 4.3).

(v) We use the PROV ontology to *record and annotate the provenance* of our generated/published data; the details are described in Section 4.4.

(vi) All the integrated, enriched and versioned metadata is *publicly available as Linked Data* at `https://data.wu.ac.at/portalwatch/`. A SPARQL endpoint allows to query the generated RDF, as described in Section 4.5.

(vii) Finally, we enable *historic access* to the original and mapped dataset descriptions using the Memento framework [de Sompel et al., 2013], cf. Section 4.5.

We discuss related work in Section 4.6 conclude this chapter in Section 4.7. The experiment and results in this chapter are based on a snapshot of the portals monitored by the Portal Watch framework in April 2017.

4.1 Metadata Models for Data Portals

The DCAT model offers three main classes: `dcat:Catalog`, `dcat:Dataset` and `dcat:Distribution`. The definition of a `dcat:Catalog` corresponds to the concept of data portals, i.e., it describes a web-based data catalog and holds a collection of datasets (using the `dcat:dataset` property). An instance of the `dcat:Dataset` class describes a metadata instance which can hold one or more distributions, a publisher, and a set of properties describing the dataset. A `dcat:Distribution` instance provides the actual references to the resource (using `dcat:accessURL` or `dcat:downloadURL`). Further, it potentially provides properties to describe license information (`dct:license`), format (`dct:format`) and media-type (`dct:mediaType`) descriptions and general descriptive information (e.g, `dct:title` and `dcat:byteSize`).

The Portal Watch framework maps the harvested metadata descriptions from CKAN, Socrata and OpenDataSoft portals to the DCAT vocabulary (as defined and described in

Table 4.1: Mapping of main classes and missing properties

DCAT	**S**chema.org
`dcat:Catalog`	`schema:DataCatalog`
`dcat:Dataset`	`schema:Dataset`
`dcat:Distribution`	`schema:DataDownload`
`dcat:frequency`	?

detail in Section 3.3). For instance, the values of the CKAN metadata fields *title*, *notes*, or *tags* get mapped to DCAT using the properties `dct:title`, `dct:description`, and `dcat:keyword` which are associated to a certain dataset instance.

Our framework generates Schema.org compliant dataset descriptions by mapping the DCAT descriptions to Schema.org's dataset vocabulary. This mapping is implemented based on a W3C working draft:[4] the three main DCAT classes `Catalog`, `Dataset`, and `Distribution` are mapped to the Schema.org classes `DataCatalog`, `Dataset`, and `DataDownload` (cf. Table 4.1) and the mapping covers all core properties specified in the DCAT recommendation except for `dcat:frequency`.

All mapped dataset descriptions for all weekly harvested versions (hereafter referred to as *snapshots*) are accessible via the Portal Watch API (`https://data.wu.ac.at/portalwatch/api`):

`/portal/{portalid}/{snapshot}/dataset/{datasetid}/dcat`
: This interface returns the DCAT description in JSON-LD for a specific dataset. The parameter `portalid` specifies the data portal and `datasetid` the dataset. The parameter `snapshot` allows to select archived versions of the dataset: the parameter has to be provided as *yyww* specifying the year and week of the dataset, e.g., 1650 for week 50 in 2016.

`/portal/{portalid}/{snapshot}/dataset/{datasetid}/schemadotorg`
: Analogous to the above API call, this interface returns the Schema.org mapping for a dataset as JSON-LD, using the same parameters.

Additionally, we publish the Schema.org dataset descriptions as single, crawl-able, web pages, listed at `https://data.wu.ac.at/odso` (and `https://data.wu.ac.at/odso/sitemap.xml` as access point for search engines, respectively). These Schema.org metadata descriptions are embedded within the HTML pages, following the W3C JSON-LD recommendation.[5]

[4] `https://www.w3.org/2015/spatial/wiki/ISO_19115_-_DCAT_-_Schema.org_mapping`, last accessed 28/05/2019

[5] `https://www.w3.org/TR/json-ld-syntax/#embedding-json-ld-in-html-documents`, last accessed 28/05/2019

4.1.1 Challenges and mapping issues

DCAT mapping available on CKAN portals: There exists a CKAN-to-DCAT extension[6] that defines mappings for CKAN datasets and their resources to the corresponding DCAT classes `dcat:Dataset` and `dcat:Distribution` and allows to access the DCAT metadata via the CKAN API. However, in general it cannot be assumed that this extension is deployed for all CKAN portals: we were able to retrieve the DCAT descriptions of datasets for 93 of the 149 active CKAN portals monitored by Portal Watch.

The CKAN software allows portal providers to add additional metadata fields to the metadata schema. When retrieving the metadata description for a dataset via the CKAN API, these keys are included in the resulting JSON under the metadata fields "`extras`". However, it is not guaranteed that the DCAT version of the CKAN metadata contains these extra fields. Depending on the version and the configuration of the export-extension we observed three different cases:

Predefined mapping: In recent versions of the extension the portal provider can define a mapping for certain CKAN fields to a specific RDF property. For instance, a CKAN extra field `contact-email` (which is not by default part of the CKAN schema and is not defined in the extension's mapping) could be mapped to an RDF graph using the property vcard:hasEmail from the vCard ontology, e.g.:

```
<https://example.com/example-dataset>
  dcat:contactPoint [
    vcard:hasEmail "example@email.com"
  ] ;
```

Default mapping: A pattern for exporting all available extra metadata keys which can be observed in several data portals is the use of the `dct:relation` (Dublin Core vocabulary) property to describe just the label and value of the extra keys, e.g.:

```
<https://example.com/example-dataset>
  dct:relation [
    rdfs:label "contact-email" ;
    rdf:value "example@email.com"
  ] ;
```

No mapping: The retrieved DCAT description returns no mapping of these keys and the information is therefore not available.

In order to avoid these different representations (and potentially missing information) of extra metadata fields, we do not harvest the DCAT mappings of the CKAN portals but rather the original, complete, JSON metadata description available via the CKAN API and apply a (refined) mapping to DCAT at our framework.

[6] `https://github.com/ckan/ckanext-dcat`, last accessed 28/05/2019

Use of CKAN *extra* metadata fields: We analysed the metadata of 749k datasets over all 149 CKAN portals and extracted a total of 3746 distinct extra metadata fields. Table 4.2 lists the most frequently used fields sorted by the number of portals they appear in. The most frequent key is `spatial`, appearing in 29 portals. Most of these cross-portal extra keys are generated by widely used CKAN extensions. The keys in Table 4.2 are all generated by the harvesting[7] and spatial extension.[8] We manually selected mappings for the most frequent extra keys if they are not already included in the mapping; the selected properties are listed in the "DCAT key" column in Table 4.2. In case of an `?`, we were not able to choose an appropriate DCAT core property.

Table 4.2: Most frequent extra keys

Extra key	**Portals**	**Datasets**	Mapping
`spatial`	29	315,652	`dct:spatial`
`harvest_object_id`	29	514,489	`?`
`harvest_source_id`	28	486,388	`?`
`harvest_source_title`	28	486,287	`?`
`guid`	21	276,144	`dct:identifier`
`contact-email`	17	272,208	`dcat:contactPoint`
`spatial-reference-system`	16	263,012	`?`
`metadata-date`	15	265,373	`dct:issued`

Table 4.3: Extra keys appearing in multiple portals

Portals	1	2	$3-9$	$10-19$	≥ 20
Extra keys	2,189	1,356	168	28	5

Table 4.3 lists the number of keys occurring in multiple portals: 33 of all extra keys occur in more than 10 of the 149 CKAN portals, which mainly originate from popular CKAN extensions. The majority of the extra keys occur only in one or two portals.[9]

Modelling CKAN datasets: The CKAN software allows data providers to add multiple *resources* to a dataset description. These resources are basically links to the actual data and some additional corresponding metadata (e.g., format, title, mime-type).

This concept of resources relates to *distributions* in DCAT [Maali and Erickson, 2014]. A DCAT distribution is defined the following way: "*Represents a specific available form*

[7] `https://extensions.ckan.org/extension/harvest/`, last accessed 28/05/2019

[8] `https://docs.ckan.org/projects/ckanext-spatial/en/latest/`, last accessed 28/05/2019

[9] The high number of keys occurring in two portals is potentially due to the fact that many portals harvest datasets, i.e. the metadata descriptions, of other portals (see the number of portals using the harvesting extension in Table 4.2).

of a dataset. Each dataset might be available in different forms, these forms might represent different formats of the dataset or different endpoints. [...]"[10] This means that distributions of a dataset should consist of the same data in different representations. We applied the following two heuristics in order to find out if CKAN resources are used as distributions, i.e., if CKAN resources represent the same content in different formats:

- *Title similarity:* We compared the titles of resources of a dataset using Ratcliff-Obershelp string similarity used in the Python difflib library.[11] In case any two resource-titles have a measure of higher than 0.8 (with a maximum similarity of 1) we consider the resources as "distributions". For instance, two resources with titles "air-temperature.csv" and "air-temperature.json" most likely contain the same data in CSV and JSON format.
- *Formats:* We looked into the file formats of the resources and report the number of datasets where all formats differ or some formats appear multiple times (e.g., a dataset consisting of two CSVs which indicates different content in these files).

Out of the 767k CKAN datasets about half of them hold more than one resource (cf. Table 4.4). Out of these 401k multi-resource datasets, for 140k datasets all corresponding file formats are different, which we can take as a strong indication that these are *distributions* of the datasets. Using string similarity we encountered similar titles for at least two resources in 261k out of the 401k multi-resource datasets.

Table 4.4: Distributions vs. resources in CKAN datasets

Total	> 1 **resources**	**All diff. formats**	**Similar titles**
767,364	401,054	140,140	261,939

These numbers indicate that there is no common agreement on how to use resources in CKAN. On the one hand there is a high number of datasets where resources are published as "distributions" (see *all diff. file formats* and *similar titles* in Table 4.4) while on the other hand the remaining datasets group resources by other aspects (see *multi-appearance*); e.g., a dataset consisting of the resources "air-temperature-2013.csv", "air-temperature-2014.csv", "air-temperature-2015.csv".

4.2 The Data Quality Vocabulary

Beside the regular crawling and monitoring of data portals, the Portal Watch framework performs a quality assessment along several quality dimensions and metrics. These dimensions and metrics are defined on top of the DCAT vocabulary, which allows us to treat and assess the content independent of the portal's software and metadata schema.

[10] `https://www.w3.org/TR/vocab-dcat/#class-distribution`, last accessed 28/05/2019

[11] `https://docs.python.org/2/library/difflib.html`, last accessed 28/05/2019

This quality assessment is performed along several dimensions: (i) The *existence* dimension consists of metrics checking for important information, e.g., if there is contact information in the metadata. (ii) The metrics of the *conformance* dimension check if the available information adheres to a certain format, e.g., if the contact information is a valid email address. (iii) The *open data* dimension's metrics test if the specified format and license information is suitable to classify a dataset as open. The formalization and implementation details can be found in Section 3.1.

The W3C's Data Quality Vocabulary (DQV) [Debattista et al., 2016] is intended to be an extension to the DCAT vocabulary. It provides classes to describe quality dimensions, metrics and measurements, and corresponding properties. We use the DQV to make the quality measures of the Portal Watch framework available as RDF and to link the assessment to the dataset descriptions. Figure 4.2 displays an example quality assessment modelled in the DQV. The italic descriptions (e.g., *dqv:QualityMeasurement* and *dqv:Metric*) denote the classes of the entities, i.e., the *a*-relations. The measurements of a dataset are described by using a blank node (cf. `_:bn`) and the `dqv:value` property to assign quality measures to the datasets.

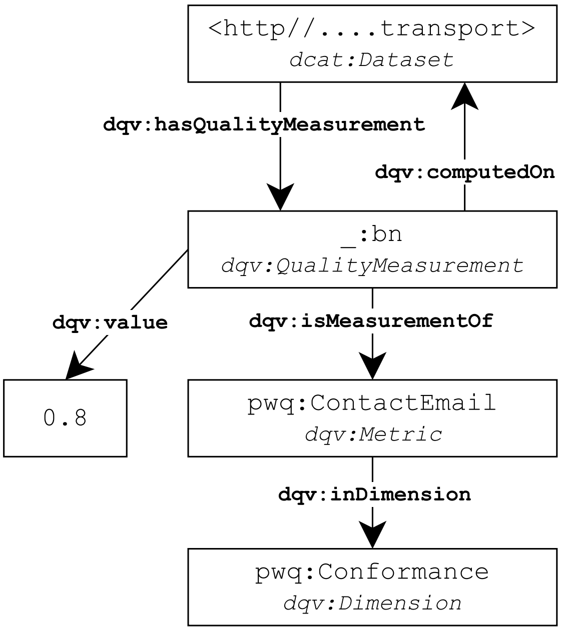


Figure 4.2: Example quality measurement using DQV

API access to the measurements: The DQV results can be retrieved by using the following API or by querying the SPARQL endpoint (see Section 4.5):

```
/portal/{portalid}/{snapshot}/dataset/{datasetid}/dqv
```

Analogous to the previous APIs (see Section 4.1), this interface returns the DQV results in JSON-LD for a specific dataset, requiring the parameters `portalid`, `datasetid` and `snapshot` (specifying the year and week of the dataset).

4.3 Describing CSVs on the Web

The W3C's CSV on the Web Working Group[12] (CSVW) proposed a metadata vocabulary that describes how CSV data (comma-separated-value files or similar tabular data sources) should be interpreted [Pollock et al., 2015]. The vocabulary includes properties such as primary and foreign keys, datatypes, column labels, and CSV dialect descriptions (e.g., delimiter, quotation character and encoding).

We use this W3C vocabulary to further describe CSV resources in our corpus of data portals. Therefore, we select all resource URLs which use `CSV` as their file format in the dataset description. We try to retrieve the first 100 lines of each of these CSVs and apply the following methods and heuristics to determine the dialect and properties of the CSVs:

- We use the "Content-Type" and "Content-Length" HTTP response header fields to get the media type and file size of the resource. Note, that both of these fields might contain not accurate information in some cases, since some servers send the content length of the compressed resource and also use the compression's media type (e.g., `application/gzip`).
- We use the Python magic package[13] to detect the file encoding of the retrieved resource.
- We slightly modified the default Python CSV parser by including the encoding detection and refining the delimiter detection (by increasing the number of sniffed lines and modifying the preferred delimiters); the Python module is online available.[14]
- We heuristically determine the number of header lines in a CSV file by considering changes to datatypes within the first rows. For instance, if we observe columns where all entries are numerical values and follow the same pattern – including the first row – we do not consider the first row as a leading header row.[15]
- We perform a simple datatype detection on the columns of the CSVs: we distinguish between columns which contain numerical, binary, datetime or any other "string" values, and use the respective XSD datatypes[16] `number`, `binary`, `datetime` and `anyAtomicType`.

[12] `https://www.w3.org/2013/csvw/wiki/Main_Page`, last accessed 28/05/2019

[13] `https://pypi.python.org/pypi/python-magic/`, last accessed 28/05/2019

[14] `https://github.com/sebneu/anycsv`, last accessed 28/05/2019

[15] Obviously, there are cases where this heuristic may fail. Our intention here is that this "guessed" information already might be of value for an user.

[16] `https://www.w3.org/2001/XMLSchema`, last accessed 28/05/2019

This acquired information is then used to generate RDF which is compliant to the CSVW metadata vocabulary [Pollock et al., 2015]. Figure 4.3 displays an example graph for a CSV resource. The blank node _:csv represents the CSV resource which can be downloaded at the URL at property csvw:url. The values of the properties dcat:byteSize and dcat:mediaType are values of the corresponding HTTP header fields. The dialect description of the CSV can be found via the blank node _:dialect at property csvw:dialect and the columns of the CSV are connected to the _:schema blank node (describing the csvw:tableSchema of the CSV).

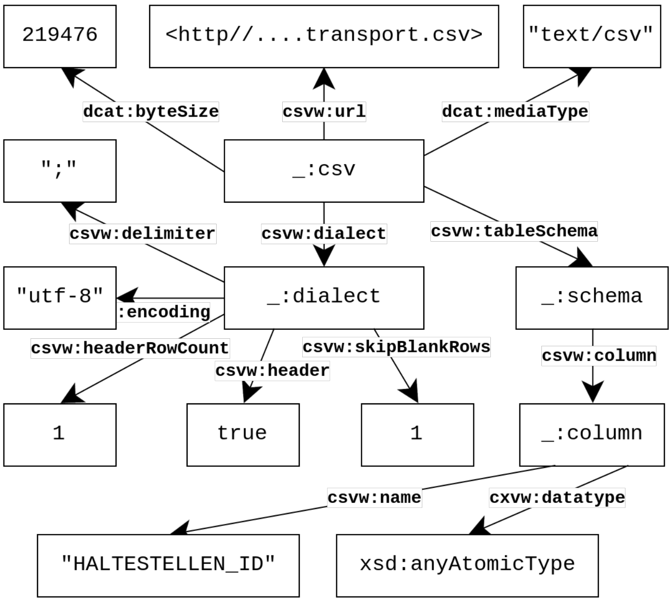


Figure 4.3: The generated CSVW metadata for an example CSV file.

In our experiment we retrieved a total of 222k URLs with “CSV” in the dataset’s metadata description from the Open Data Portal Watch. Out of these, we successfully parsed and generated the CSVW metadata for 153k files. For 44k files we were not able to parse the file and read the first lines. Possible reasons are that the files are not in the described format (e.g., compressed) or our parser was not able to detect/guess the delimiter of the CSV table. The remaining download URLs are either malformed URLs, or resulted in connection timeout and server errors.

4.4 Provenance Annotation

Apart from generating mappings, quality measurements and enrichments of the metadata alone, in order to make data traceable and allow users to judge the trustworthiness of data, it is import to record the provenance of our generated/published data. There are several approaches to address this issue for RDF. A lightweight approach could use different Dublin Core properties to refer from a dataset to entities/agents (i.e., our system) which published the resources, e.g., by using properties such as `dc:publisher`. However, the DCAT metdata descriptions already use these Dublin Core properties and therefore such additional annotations would interfere with the existing dataset descriptions.

The PROV ontology [Lebo et al., 2013], introduced in Section 2.4.2, is a more flexible approach which provides an ontology to annotate all kinds of resources with provenance information and allows tracking of provenance of resource representations.

To add provenance information to our generated RDF data we define a `prov:SoftwareAgent` (a subclass of `prov:Agent`) with URI `<https://data.wu.ac.at/portalwatch>`, cf. Figure 4.4. Since our Portal Watch framework generates weekly *snapshots* of portals, i.e., weekly versions of the datasets of a data portal, and also assesses the quality of these fetched datasets, we associate such a snapshot with a `prov:Activity` which generated the DCAT representation of the dataset and the respective quality measurements. The measurements were computed on the DCAT dataset descriptions which is modelled using the `prov:wasDerivedFrom` property.

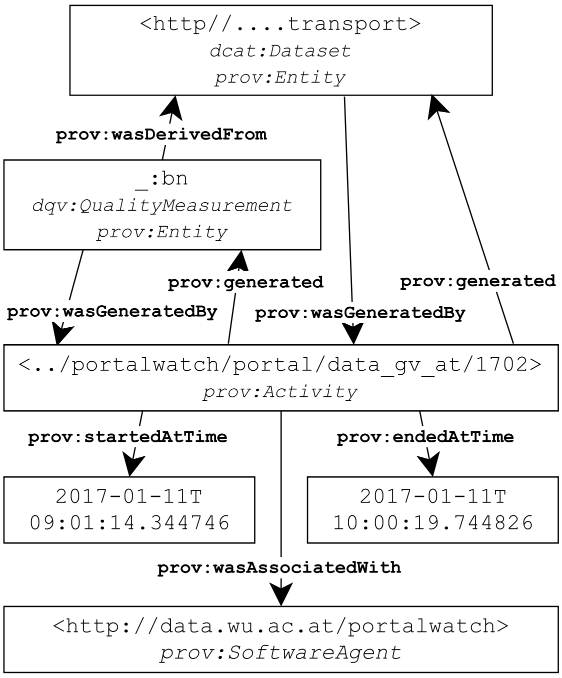


Figure 4.4: Provenance annotation to quality measurement

Regarding the (heuristically) generated CSVW metadata, we annotate all `_:csv` resources (cf. Section 4.3) as `prov:Entity` and associate them with a `prov:Activity` with URI `<https://data.wu.ac. at/portalwatch/csvw/{snapshot}>` for a corresponding snapshot. These activities represent the weekly performed metadata/dialect extraction on the CSVs. Additionally, we add the triple `_:csv` - `prov:wasDerivedFrom` - *CSV-url*, to indicate that the CSVW-metadata entities were constructed based on the existing CSV resources.

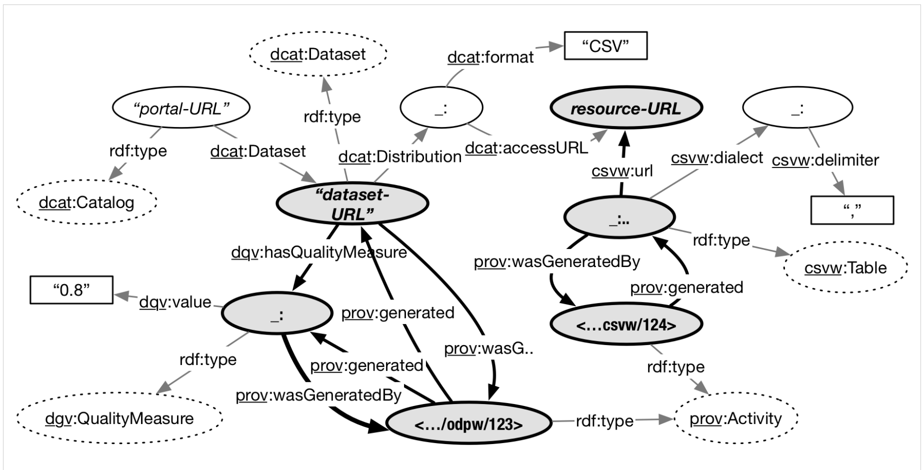


Figure 4.5: Properties and nodes which enable connections between generated datasets

4.5 Data Access & Client Interfaces

This section describes how the generated RDF data is connected and how we enable access to this data. In the previous sections we described four different datasets: (i) the homogenized representation of metadata descriptions (using the DCAT vocabulary), (ii) quality measurements of these descriptions along several dimensions, (iii) additional table schema and dialect descriptions for CSV resources, and (iv) provenance information for the generated RDF data.

In the example graph in Figure 4.5 bold edges and bold nodes represent the properties and resources which connect these four generated datasets. The corresponding classes for the main entities are depicted using dashed nodes.

In the following we introduce the public SPARQL endpoint for querying the generated data and the implemented Memento APIs which provide access to the archived datasets by using datetime negotiation.

SPARQL Endpoint

We make the mapped DCAT metadata descriptions and their respective quality assessments available via a SPARQL endpoint located at the Portal Watch framework (https://data.wu.ac.at/portalwatch/sparql). Currently, we loaded three snapshots of the generated data in the RDF triple store (week 2, 3, and 4 in 2017), where each snapshot is published as a named graph. These snapshots consist of about 120 million triples each. However, the numbers are varying because we observe server errors for certain portals and therefore we are not able to harvest the same number of dataset descriptions every week. The underlying system is OpenLink Virtuoso.[17]

[17] https://virtuoso.openlinksw.com/, last accessed 28/05/2019

In order to describe the quality metrics and dimensions of the Portal Watch framework we define URLs which refer to the respective definitions (using the pwq namespace).

Exploring datasets The SPARQL endpoint allows users to explore and search datasets across data portals and find common descriptions and categories.

For instance, the query in Listing 4.1 returns all portals in the Portal Watch system which use *transportation* as a keyword/tag (in total 31 portals).

```
select distinct(?p)
where {
  ?p dcat:dataset ?d.
  ?d a dcat:Dataset.
  ?d dcat:keyword "transportation".
}
```

Listing 4.1: All portals holding transportation data

Metadata quality comparison and aggregation The SPARQL endpoint also allows to compare and filter datasets across different portals, and the aggregation of quality metrics on different levels.

For instance, the query in Listing 4.2 lists an aggregation of the ContactEmail quality metric (see Section 3.3 for definitions) for the organizations (i.e. publisher) on the Austrian data portal data.gv.at.

```
SELECT ?orga AVG(?v)
WHERE {
  <https://data.gv.at> dcat:dataset ?d .
  ?d dct:publisher ?orga .
  ?d dqv:hasQualityMeasurement ?m .
  ?m dqv:isMeasurementOf pwq:ContactEmail .
  ?m dqv:value ?v .
}
GROUP BY ?orga
```

Listing 4.2: Average *ContactEmail* metric per organization

Memento framework

In order to enable a standardized access of the harvested and archived dataset descriptions of the Portal Watch framework we use the HTTP-based Memento framework [de Sompel et al., 2013]. We implemented *pattern 2* of the specification, "*A Remote Resource Acts as a TimeGate for the Original Resource*", which we detail in the follwing.

Initially, we introduce the following terms specific to the Memento framework which we use in our system:

Original Resource (URI-R): The Original Resource is a link to the resource for which our framework provides prior states. In our implementation this URI-R is the landing page for a dataset description at a specific data portal. For instance, the URI-R *uri-r*[18] is an available dataset description at the Austrian data portal `data.gv.at`.

Time Gate (URI-G): The TimeGate URI for an URI-R is a resource provided by our Memento implementation that offers *datetime negotiation* in order to support access to the archived version of the original resource. The URI-G for the a specific dataset is available at <`https://data.wu.ac.at/portalwatch/api/v1/memento/{portalid}/{datasetid}`> using the internal portal-ID and the dataset's ID; e.g., *uri-g*[19] for the above dataset.

Memento: A Memento for an URI-R is a resource which provides a specified prior state of the original resource. The Memento for the a dataset description is available at <`https://data.wu.ac.at/portalwatch/api/v1/memento/{date}/{portalid}/{datasetid}`>, where `date` follows the pattern `YYYY<MM|DD|HH|MM|SS>` (the parameters within < and > are optional). The Memento for a specific given `date` is defined as the closest available version *after* the given date. For instance, the archived version for the example dataset *uri-r* can be accessed at *uri-m*[20]; this URI returns the archived dataset description closest after January 1 2017.

In our implementation we offer these Mementos (i.e., prior versions) with explicit URIs in different ways: (i) we provide access to the original dataset descriptions retrieved from the data portals' APIs (e.g., *uri-m* which returns the archived JSON metadata retrieved from a CKAN data portal), (ii) the dataset description mapped to the DCAT vocabulary (using the suffix `/dcat` for the URI-T and Memento resources), or Schema.org vocabulary (using suffix `/schemadotorg`), serialized as JSON-LD, and (iii) the quality assessment results in the DQV vocabulary (using suffix `/dqv`), serialized as JSON-LD.

Datetime negotiation: The Memento framework specifies a mechanism to access prior versions of Web resources based on the level of HTTP request and response headers. It introduces the "Accept-Datetime" and "Memento-Datetime" HTTP header fields and extends the existing "Vary" and "Link" headers [de Sompel et al., 2013]. In order to support datetime negotiation within our Memento implementation we implemented these headers for the available URI-G and Memento resources.

[18] *uri-r*: <`https://www.data.gv.at/katalog/dataset/add66f20-d033-4eee-b9a0-47019828e698`>

[19] *uri-g*: <`https://data.wu.ac.at/portalwatch/api/v1/memento/data_gv_at/add66f20-d033-4eee-b9a0-47019828e698`>

[20] *uri-m*: <`https://data.wu.ac.at/portalwatch/api/v1/memento/data_gv_at/20170101/add66f20-d033-4eee-b9a0-47019828e698`>

Our framework implementation follows a 200 negotiation style: a request to the TimeGate URI of a resource has a “200 OK” HTTP status code and already returns the requested Memento. To indicate that our TimeGate URIs are capable of datetime negotiation the “Vary” header includes the “accept-datetime” value (cf. Listing 4.4). Since the original dataset descriptions, i.e., the URI-Rs, are hosted by remote servers we cannot support Memento-compliant HTTP headers for these resources.

In order to retrieve a archived version, a request to the TimeGate of a resource can include the “Accept-Datetime” HTTP header. This header indicates that the user wants to access a past state of the resource. If this header is not present, our implementation will return the most recent version of the resource (i.e., the most recent archived dataset description). Otherwise, the response to this request is the closest version of the resource after the transmitted datetime header value, i.e, the corresponding Memento. For instance, in Listing 4.3 a request to *uri-g* is issued including an “Accept-Datetime”.

```
HEAD /portalwatch/api/v1/memento/data_gv_at/add66f20-d033-4eee-b9a0
   -47019828e698 HTTP/1.1
Host: data.wu.ac.at
Accept-Datetime: Sun, 01 Jan 2017 10:00:00 GM
```

Listing 4.3: Request Datetime Negotiation with *uri-g*

The response header to such a datetime negotiation request with the URI-G of a resource includes the “Memento-Datetime” header which expresses the archival datetime of the Memento. Further, it includes the “Content-Location” header which explicitly directs to the Memento URI, i.e., to the distinct URI of the archived resource. The “Link” header contains URI-R with the “original” relation type (the link to the original dataset description) and URI-G with the “timegate” relation type. These header fields are also included in all Memento URIs’ response headers, e.g., also in the header of *uri-m*.

Listing 4.4 shows the HTTP response header to the request to *uri-g* in Listing 4.3. This header includes the crawl time of the archived dataset in the “Memento-Datetime” header and provides a direct link to the Memento in the “Content-Location” header. The “Link” header includes the reference to the original dataset at the data portal.

4.6 Related Work

Similar to DCAT, the VoID vocabulary [Alexander et al., 2011] is an early approach (2011) published by the W3C as an Interest Group Note. VoID – the Vocabulary for Interlinked Datasets – is an RDF schema for describing metadata about linked datasets: it has been developed specifically for data in RDF representation and is therefore complementary to the DCAT model and not fully suitable to model metadata on Open Data portals (which usually host resources in various formats) in general.

In 2019 Kremen and Necaský [Kremen and Necaský, 2019] introduced a semantic government vocabulary (SGoV) for Open Government Data. The vocabulary allows rich

```
HTTP/1.0 200 OK
Content-Type: application/json
Memento-Datetime: Sun, 25 Dec 2016 23:00:00 GMT
Link: <https://www.data.gv.at/katalog/dataset/add66f20-d033-4eee-b9a0
    -47019828e698>; rel="original",
     <https://data.wu.ac.at/portalwatch/api/v1/memento/data_gv_at/
        add66f20-d033-4eee-b9a0-47019828e698>; rel="timegate"
Vary: accept-datetime
Content-Location: https://data.wu.ac.at/portalwatch/api/v1/memento/
    data_gv_at/20161226/add66f20-d033-4eee-b9a0-47019828e698
Content-Length: 11237
Date: Mon, 16 Jan 2017 16:30:21 GMT
```

Listing 4.4: Response from *uri-g* to request of Listing 4.3

and in-depth annotations of governmental institutions, political parties, members of a party, etc. We would encourage governmental data publishers to evaluate and consider the proposed vocabulary for annotation and publishing.

In 2011 Fürber and Hepp [Fürber and Hepp, 2011] propose an ontology for data quality management that allows the formulation of data quality, cleansing rules, a classification of data quality problems and the computation of data quality scores. The classes and properties of this ontology include concrete data quality dimensions (e.g., completeness and accuracy) and concrete data cleansing rules (such as whitespace removal). In total the ontology consists of about 50 classes and 50 properties. It allows a detailed modelling of data quality management systems, and might be partially applicable and useful in our system and to our data. However, we decided to follow the W3C Data on the Web Best Practices and use the more lightweight Data Quality Vocabulary for describing the quality assessment dimensions and steps.

In the last years there has been a trend towards the use of the DCAT vocabulary in public sector data portals. For instance, the European Data portal[21] – a harvesting portal of European public sector information – offers all the harvested data using the DCAT-AP standard and provides a SPARQL endpoint to query these descriptions. However, so far the data at this portal is not really "linked", i.e., there no links to external resources and no cross-references between datasets, and there are no rich annotations of the public institutions, publishing agencies and geographic entities (such as [Kremen and Necaský, 2019]) and no annotations of the datasets' content (such as CSVW metadata [Pollock et al., 2015]).

In 2018 Google launched their Google Dataset Search engine[22] [Brickley et al., 2019] which is currently based on harvesting metadata descriptions in Schema.org from data portals. This means the search engine relies on data providers describing their metadata

[21] `https://www.europeandataportal.eu/`, last accessed 01/07/2019

[22] `https://toolbox.google.com/datasetsearch`, last accessed 01/07/2019

in Schema.org. However, Schema.org metadata is not yet widely-used across data portals; so the search engine also harvests our re-published Schema.org mappings and offers them via their search interface. The CSV schema and metadata descriptions – as generated and published by our framework – are not included in Schema.org and not part of the Google Dataset Search.

A recent approach by Tygel et al. [Tygel et al., 2016] tries to establish links between Open Data portals by extracting the tags/keywords of the dataset descriptions and merging them (using translations, and similarity measures) at a tag server, where they provide unique URIs for these tags. The tags are further described using relations such as `skos:broader`, `owl:sameAs` and `muto:hasMeaning`. We will investigate how and if we can use this service to connect our generated RDF data to these tag descriptions.

4.7 Critical Discussion and Future Directions

In this chapter we have described vocabularies and methods to automatically improve and (re-)expose dataset descriptions as Linked Data. We have realized the described methods as an extension of the Portal Watch system, and processed the metadata of the 261 indexed data portals. In detail, we provide metadata descriptions in DCAT and Schema.org, we annotate datasets with quality measurements using the W3C's Data Quality vocabulary, and we further enrich the dataset descriptions by automatically generated metadata for CSV resources such as the column headers, column datatypes and CSV delimiter. Also, in order to ensure traceability of the published RDF data, the mapped/generated dataset descriptions and respective measurements include provenance annotations. To allow access to archived versions of the dataset descriptions, the Portal Watch framework offers APIs based on the Memento framework, i.e. time-based content negotiation on top of the HTTP protocol. We see the following next steps and potential improvements:

Generate richer metadata: An important part of publishing Open Data is the attached license. An appropriate license can be crucial for consumers. However, as discussed in Section 2.4.1, current license definitions on Open Data portals lack a machine-readable description. An important part that is missing in our work is this formal representation of the datasets' license (using for instance the ODRL language [Iannella and Villata, 2017]). Such machine-readable annotations would also allow to express and check fine-grained access restrictions and policies [Steyskal and Polleres, 2014].

Further, we see the potential to improve the CSV analysis and generated CSVW metadata. For example, the column datatypes of the CSVW metadata are based on the XSD datatype definitions. These types are hierarchically defined (e.g., a *positive integer* is also a *integer*, is also a *decimal*). More advanced heuristics can be applied to the values in order to generated more fine grained datatypes. For instance, the specification allows to define patterns for date(time) columns which could be automatically detected by such an heuristic.

Representing snapshots as historical data: In the Portal Watch framework a weekly snapshot of the monitored portals is stored together with the quality assessments. In the triple store, the generated RDF is then stored for each snapshot as a new named graph. However, one might be interested in asking queries such as *"How regular does the metadata of this this dataset change?"*, *"When did the last change to a certain metadata field occur?"*, or *"How did the quality of a dataset evolve over time?"*; the current data model is not sufficient (or not practicable) for such issues.

Also we have to deal with scalability issues considering the currently produced number of generated triples. The Portal Watch framework monitors and archives (in a relational database) the metadata descriptions for ∼250 portals already for about one year. Assuming that also the previous snapshots consist of about 120 million triples per snapshot for the archived versions, we could very roughly estimate the number of total triples to 6 billions (50 weeks × 120M triples). If we also assume that we want to keep up our service in the future and that the number of datasets and portals will further increase, we have to investigate on how we can store the data efficiently while maintaining the services to retrieve and use the data.

There are already several ongoing approaches which try to cope with these issues: In [Fernández et al., 2019] Fernandez et al. benchmark existing RDF archiving and stroage techniques along several aspects such as storage space efficiency, retrieval functionality, and performance of various retrieval operations. The authors identify three main archiving strategies for RDF: (i) storing *independent copies* for each version corresponds to our current approach of different named graphs for each snapshot. To address the scalability issue of this strategy, (ii) *change-based approaches* compute and store the deltas between versions. Alternatively, (iii) in *timestamp-based approaches* each triple is annotated with its temporal validity.

A recent approach by Fionda et al. [Fionda et al., 2016] proposes a framework for querying RDF data over time by extending SPARQL. This extension inherits temporal operators from Linear Temporal Logics, e.g., *PREVIOUS*, *ALWAYS*, or *EVENTUALLY*. A logical and necessary next step for our metadata archive is to select and implement a suitable model.

Interlink datasets and connect to external knowledge: The metadata, as it is currently published at our Portal Watch framework, is only partially interlinked and there are hardly any links to external knowledge bases. The reason for this is that the origin portal frameworks (e.g. CKAN, Socrata) do not provide options to describe related/associated datasets, or options to describe the datasets using external vocabularies or to add links to classes and properties in external sources.

The next chapter will focus on how to add such links and connections: to extract labels, properties and classes from the actual data sources and use these to enrich the metadata and establish connections between datasets. There is already an extensive body of research in the Semantic Web community to derive such semantic labels which we build upon, e.g. [Venetis et al., 2011, Ritze et al., 2016, Adelfio and Samet, 2013].

CHAPTER 5

Semantic Enrichment and Search

With the advent of *Knowledge Graphs* [Bonatti et al., 2019] traditional web search recently has been revolutionized in that search results can be categorized, browsed and ranked according to well-known concepts and relations, which cover typical search scenarios in search engines. But these scenarios are different for Open Data: in our experience, dataset search needs to be targeted from a different angle than keyword-search (alone).

We argue that - just like for regular Web search - Knowledge Graphs can be helpful to significantly improve dataset search; specifically to our use case, we aim at constructing Knowledge Graphs from publicly available sources. In the following chapter we propose two complementary approaches and discuss and evaluate their feasibility: First, we construct a Knowledge Graph to find candidates for semantic labels for a given bag of numeric values (e.g., a CSV column); and second, we use a Knowledge Graph to automatically annotate spatial and temporal information in datasets. In fact, the ingredients for building such a Knowledge Graph of geographic entities as well as time periods and events exist already on the Web of Data, although they have not yet been integrated and applied – in a principled manner – to the use case of Open Data search.

The contributions in this chapter are structured as follows:

(i) We propose an approach to *find and rank candidates of semantic labels and context descriptions for a given bag of numerical values* in Section 5.1. To this end, we apply a hierarchical clustering over information taken from DBpedia to build a background Knowledge Graph of possible "semantic contexts" for bags of numerical values, over which we perform a nearest neighbour search to rank the most likely candidates.

(ii) We present a scalable approach to *construct a spatio-temporal Knowledge Graph that hierarchically structures geographical as well as temporal entities, and annotate a large corpus of tabular datasets* from Open Data portals with entities from this Knowledge Graph in Section 5.2.

(iii) In Section 5.3 we *enable structured, spatio-temporal, querying and search* over the annotated datasets, and further showcase how the spatio-temporal labels can be used to automatically generate visualizations.

Eventually, we present related works in the field in Section 5.4 and conclude the chapter with discussion and future directions in Section 5.5.

5.1 Semantic Labelling of Numerical Values

Connecting CSV data to the Web of Linked Data involves typically two steps, that is, (i) transforming tabular data to RDF and (ii) mapping, i.e. linking the columns (which adhere to different arbitrary schemata) and contents (cell values) of such tabular data sources to existing RDF knowledge bases. While a recent W3C standard [Tandy et al., 2015],[1] provides a straightforward canonical solution for (i), the mapping step (ii) though remains difficult.

Mapping involves to semantically label columns by linking column headers or cell values to either properties or classes in ontologies or instances in knowledge bases, and to determine the relationship between columns [Taheriyan et al., 2014]. For the semantic labelling, most approaches so far rely on mapping textual values [Venetis et al., 2011, Wang et al., 2012, Syed et al., 2010]; these work well e.g. for HTML/Web tables which have rich textual descriptions, as they are published mainly for human consumption. However, in typical Open Data portals many data sources exist where such textual descriptions (such as column headers or cell labels) are missing or cannot be mapped straightforwardly to known concepts or properties using linguistic approaches, particularly when tables contain many numerical columns for which we cannot establish a semantic mapping in such manner.

Indeed, as we already showed in a large-scale profiling in Section 3.6, a major part of the datasets published in Open Data portals are tables containing many numerical columns with missing or non human-readable headers (e.g., organisational identifiers, sensor codes, internal abbreviations for attributes like “population count”, or geo-coding systems for areas instead of their names, e.g. for districts, etc.) [Lopez et al., 2012]. We also verified this observation by inspecting 1200 tables collected from the European Open Data portal and the Austrian Government Open Data Portal and attempted to map the header values using the BabelNet service (`http://babelnet.org`): on average, half of the columns in CSV files served on these portals contain numerical values, only around 20% of which the header labels could be mapped with the BabelNet services to known terms and concepts (cf. more details in our evaluation in Section 5.1.4). Therefore, the problem of semantically labelling numerical values, i.e., identifying the most likely property or classes for instances described by a bag of numerical values remains open.

Some early attempts focus on specific “known” numerical datatypes, such as longitude and latitude values [Cruz et al., 2013], or – more generally – on classifying numerical columns

[1] Or, likewise with RDB2RDF direct mapping [Arenas et al., 2012], the basis of [Tandy et al., 2015].

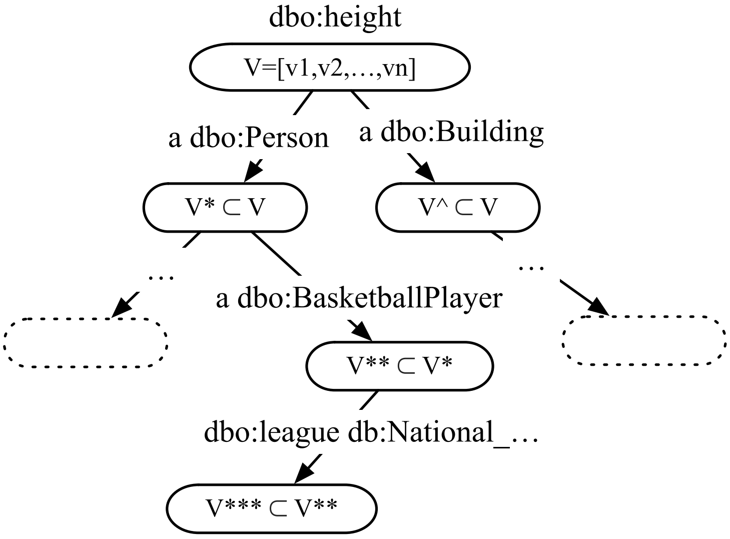


Figure 5.1: Hierarchical background knowledge

using (manually) pre-labelled numeric value sets [Ramnandan et al., 2015]. To the best of our knowledge, so far no unsupervised approaches have been devised for semantic labelling of numerical value sets. Additionally, the latter approach by Ramnandan et. al. only assigns a single predefined semantic label, corresponding to a “property” per column. In the context of RDF, we deem such semantic labelling insufficient in (at least) two aspects: (a) We do not only need to map columns to properties, but to what we will call “contexts”, that is property-domain pairs. (b) Since, given the variety and heterogeneity of Open Data, it is likely we cannot rely on a manually curated, pre-defined set of semantic labels. Therefore, there is a need to build a hierarchical “background Knowledge Graph” of semantic labels in an unsupervised manner, cf. Figure 5.1. As an example for (a), we do not only want to label a bag of numerical values as *height*, but instead we want to identify that the values represent the *heights of basketball players who played in the NBA*, or that the values represent the *heights of buildings*.

Even if we cannot identify such precise labels, we still want to assign the most likely contexts the values belong to, e.g. *height of a person*. To this end, and in order to achieve (b), we automatically generate a hierarchical background knowledge base of contexts from DBpedia. Different than previous approaches that assign a single label to a bag of values, we assign different labels/contexts, with different confidence values. This way, our approach could potentially be combined with textual labelling techniques for further label refinement, which is left for future work. Herein, we present this concrete contributions in the following order:

1. We propose a *hierarchical clustering over an RDF Knowledge Graph* to build a

background Knowledge Graph containing information about typical numerical representatives of contexts, i.e., grouped by properties and their shared domain (subject) pairs, e.g. city temperatures, peoples ages, longitude and latitudes of cities in Section 5.1.2.

2. We perform a *k-nearest neighbours search* and aggregate the results of *semantically labelling numerical values at different levels* in our Knowledge Graph in Section 5.1.3.

3. We *evaluate our approach by cross-validating over a sample* of DBpedia data generated from the most widely used numeric properties and their associated domain concepts in Section 5.1.4.

4. We *test our approach "in the wild"* on tabular data extracted from Open Data portals and report valuable insights and upcoming challenges which we have to tackle in order to successfully label data from the Open Data domain in Section 5.1.4.

5.1.1 Approach

Next, we outline the steps of our approach of finding the most likely semantic label and to determine the context in which a bag of numerical values are derived. In the following we formally define our notation and state the problem.

We denote a bag of numerical values annotated by a given label and context description $< l, c >$ as $\mathcal{V}^{<\mathrm{l,c}>} = \{v_1, v_2, \cdots, v_n\}$, with $v_i \in \mathbb{R}$. Similar to [Ramnandan et al., 2015], we define a semantic label l as an attribute of a set of values, which can potentially appear in different contexts. In this work, the semantic label l is a property from an ontology. However, this could be generalised. The concept of context description corresponds to a set of attribute-value pairs which explain/describe the commonalities of the values in $\mathcal{V}^{<\mathrm{l,c}>}$. As such, one can assume that the set of input values $\mathcal{V}^{<\mathrm{l,c}>}$ are the result of applying a query over a knowledge base ($\mathcal{V}^{<\mathrm{l,c}>} = \mathcal{Q}(KB) = \{v_1, v_2, \cdots, v_k\}$) with the semantic label and the set of attribute value pairs as filter attributes of the query. For instance, the following SPARQL query returns the set of values labelled with *height* and sharing the attribute-value pair *a basketball player*:

```
SELECT ?v WHERE {[a dbo:BasketballPlayer] dbp:height ?v.}
```

Numerical values for a semantic label can appear in different contexts. For instance, values can represent the *height* of a building, mountain or a person. Even further, we might find values representing the height of basketball players that played in the NBA. We model this observation in form of a tree for each label l. The root node in such a tree corresponds to the set of all values which fulfill the property l. The remaining nodes of the tree represent further semantic information for this values, i.e., a shared context in the form of attribute-value pair. Edges in the tree are subset-relations between these values, directed from the superset to the subset. For instance, considering the semantic label *height*, the root node could have child nodes corresponding to the context *a mountain* and *a person*.

name	capacity	city	country
Ernst Happel ...	50865	Vienna	Austria
Franz Horr Stadium	13400	Vienna	Austria
Red Bull Arena	32000	Salzburg	Austria
...	...	...	...

Table 5.1: Example table

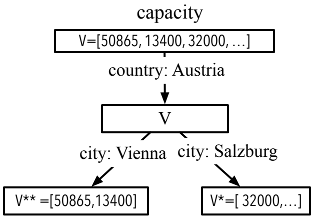


Figure 5.2: Resulting tree

The background knowledge can be constructed in an either top-down or bottom-up approach The former starts with the root node of the graph and then detects subsets while the latter starts with leaf nodes which are then combined into parent/super nodes. The top-down approach is suitable for building the context graph from RDF Knowledge Graphs and requires to start with a set of entities which are described by several attribute-value pairs. Next, we can group such entities by attributes which have numerical values, and then detect subgroups of entities with shared attribute-value pairs. We will show in the next section how we can build the background Knowledge Graph from an RDF Knowledge Graph.

The bottom-up approach is more suitable for building the background knowledge from a set of CSV files. We first find a set of annotated numerical value triples $\{(v_1, l_1, c_1)$,$(v_2, l_1, c_2), \cdots, (v_n, l_m, c_n)\}$, each consisting of a set of numerical values v_i, a label l_j and a context c_i. An input triple (v, l, c) can be extracted from a numerical column which was either manually or automatically annotated with semantic labels (e.g. based on the column header). The possible context information can be modelled from column headers, the author or title of the table, or shared attributes within the table.

For instance, take the example table in Table 5.1 and the numerical column `capacity`, as context we could extract that the numerical values describe an attribute of entities which are of type *football stadium*. Further, all values share the attribute-value pair *country: Austria*. Additionally, we could build a subset of values with the common context *city: Vienna* and another subgroup with the context *city: Salzburg* (cf. Figure 5.2). The resulting background knowledge can be exploited by machine learning algorithms or statistical methods to predict the most likely label and context for a given bag of input values. We will outline how we apply a nearest neighbour search approach to derive the most likely label and context pair for a set of values in Section 5.1.3.

5.1.2 Background Knowledge Graph Construction from DBpedia

Herein, we outline our automatic top-down approach to build a background Knowledge Graph from RDF data. To do so, we execute the following steps:

1. We extract all RDF properties which have numerical values as their objects and group the subjects by their numerical properties. These properties are used as

labels. We derive the list of RDF properties which have numerical values as their range; the following SPARQL query could be used, cf. [Fleischhacker et al., 2014], however, we note that this query does not return results on the live DBpedia SPARQL endpoint due to timeouts:

```
SELECT ?p, COUNT(DISTINCT ?o) AS ?cnt
WHERE {?s ?p ?o. FILTER (isNumeric(?o))} GROUP BY ?p
```

Another approach would be to directly query the vocabularies if we know that the RDF KB contains OWL vocabulary listing all datatype properties. We resorted to just filtering triples of the DBpedia dump with numeric objects, sorting them by property and counting via a script.

2. Next (in another pass/sorting), we collect/group by subjects in the different property groups the values of the numerical properties l. For "typing" of these subjects we collect property-object pairs – what we call context – for which the object is an RDF resource (an IRI); this includes `rdf:type`-*Class* triples, but also others, e.g. `dbo:locatedIn-dbr:Japan`.

3. Next, we also extract and materialise the OWL class hierarchy for the *Class*es. This can be done directly by extracting the `rdfs:subClassOf` hierarchy from the DBpedia ontology for these *Class*es; we will use this *type hierarchy* to further enrich our background graph collecting contexts.

After grouping the entities by the selected context labels we construct our background Knowledge Graph as follows: An abstraction of our graph is depicted on the left hand side of Figure 5.3: the graph consists of multiple trees, each tree corresponding to a property. The root node of such a tree is labelled by the property and contains the bag (i.e., multiset) of all numerical values of this property.

4. The first "layer" of our Knowledge Graph is the so-called type hierarchy which represents the `rdfs:subClassOf` relation for all available types of the triples for property l from the *type hierarchy.* Since subjects can be of more than one type, the sibling nodes in this layer can share values from the same triples. In order to not keep too fine grained, rare classes, we filter by discarding types with less than δ instances (e.g., property-class combinations with less than 50 instances).

5. Next, we construct the second layer, termed *p-o* hierarchy for the identified non-`rdf:type` property-object pairs to further refine out context structure, beyond classes, using a divisive hierarchical clustering approach. We start with one node/-group and split/compute sub-contexts recursively as we move down the hierarchy, to further refine the type hierarchy. In order to decide how to split a node, we impose the following requirements for possible candidates:

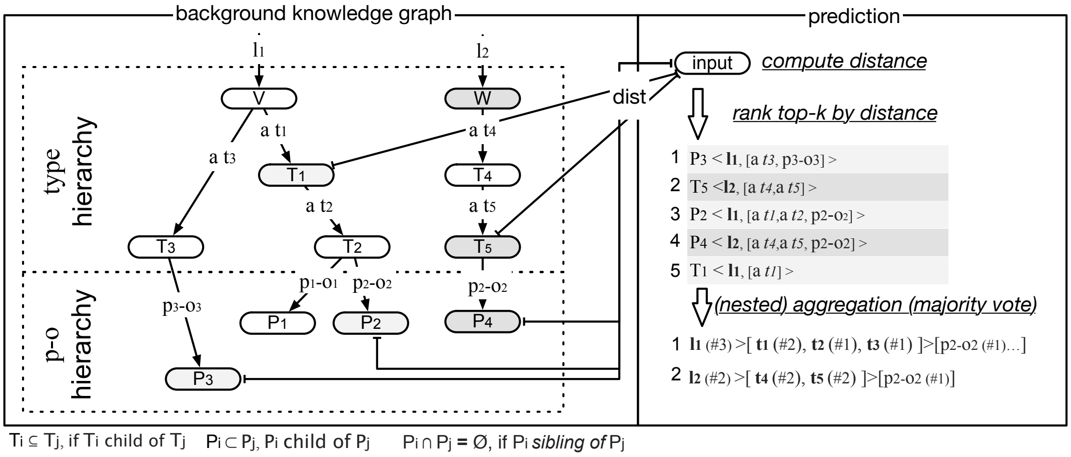

background knowledge graph
prediction
l1
l2
type hierarchy
p-o hierarchy
V
W
a t1
a t3
a t4
T1
T4
a t2
a t5
T3
T2
T5
p1-o1
p2-o2
p2-o2
p3-o3
P1
P2
P4
P3
dist
input
compute distance
rank top-k by distance
1 P3 < l1, [a t3, p3-o3] >
2 T5 <l2, [a t4,a t5] >
3 P2 < l1, [a t1,a t2, p2-o2] >
4 P4 < l2, [a t4,a t5, p2-o2] >
5 T1 < l1, [a t1] >
(nested) aggregation (majority vote)
1 l1 (#3) >[t1 (#2), t2 (#1), t3 (#1)]>[p2-o2 (#1)…]
2 l2 (#2) >[t4 (#2), t5 (#2)]>[p2-o2 (#1)]
Ti ⊆ Tj, if Ti child of Tj
Pi ⊂ Pj, Pi child of Pj
Pi ∩ Pj = ∅, if Pi sibling of Pj

Figure 5.3: Background knowledge & prediction

a) **constrain property-object:** we use the same constraint as [Fleischhacker et al., 2014] that subjects in a candidate node share the same property-object pair.

b) **constrain size:** again, in order to avoid too fine-grained subdivision, the size of a candidate node has to be larger than 1% of the parent node size (or, resp. larger than δ) and smaller than 99% of the parent node size.

Once the set of possible sub-contexts is computed, we sort the candidates by their distance to the parent node in descending order. Details on the distance measures used to compare bags of numerical values are given in Section 5.1.3. To guarantee a high diversity as well as disjointedness of the sub-contexts within the hierarchy, we select the candidate with the biggest distance first, and then subsequently the non-overlapping sub-contexts from the list with decreasing parent distance. Additionally, the disjointedness requirement also helps to limit the number of sub groups. We recursively perform the above steps for the new selected groups. consequently, shared property-object pairs of a node on the *p-o* hierarchy are encoded in the path to the resp. *p-o* node.

Node type terminology: Regarding terminology, we refer to the *exact type* of a context graph node as the lowest type node in the path to a *p-o* node. For instance, considering node P_3 in our example in Figure 5.3, the exact type would be T_2. As a *super type*, we consider all type nodes on the path between the exact type node and the *p-o* node (e.g. T_1 would be a super type of node P_3). Eventually, the *root type* of a node, is the highest type node on the path to the *p-o* node (e.g. T_1 is the root type of P_3).

5.1.3 Prediction approach

We use nearest neighbours classification over our background Knowledge Graph to predict the most likely "semantic context" for a given bag of numerical values. Given an input bag, we compute the distance between the values to all context nodes in our background Knowledge Graph and return the resp. contexts in ascending order of distance. Ideally, the node with the closest distance is the most likely semantic context/description for the input values. However, obviously numerical values for different types and properties might share the same value range and distribution and so we cannot even expect that the correct semantic description is always the top ranked result. As such, we also provide aggregation functions for predicates, type and *p-o* nodes over the top-k results. The idea is similar to the K-nearest neighbour classification for which the classification of an object is based on a majority vote over the top-k neighbour contexts.

Distance measures

An important part for any prediction algorithm, be it based on machine learning or statistical methods, is the distance measure to determine how closely related two items (e.g. feature vectors) are. We consider two distance measures, namely (i) the euclidean distance between two feature vectors and (ii) the distribution similarity between two bags of numerical values.

Euclidean distance between descriptive features: The first distance function is the euclidean distance between two numerical n-dimensional feature vectors. For our use case we consider the following features for the vectors:

- *min and max value:* The range of minimum and maximum values is an important feature which allows us to easily discard "out of scope" labels or contexts. For instance, the heights of humans might have a maximum range of 213 centimeters which distinguishes it from buildings which have much higher max height.
- *5% and 95% quantile*: Due to the fact that minimum and maximum values as features are prone to outliers and errors in the set of values we also consider quantiles and inter-quantile ranges, e.g., using 5%- and 95%-quantile instead of min and max as features in a feature vector [Fleischhacker et al., 2014].
- *Additional descriptive statistics (mean, stddev):* Additionally, descriptive features such as the mean and the standard deviation of a set of values give better results for values which are within the same range but follow different distributions.

Distribution similarity: Another distance measure is the similarity of two distributions of numeric values. This approach was already successfully used in a similar setup by Ramnadan et al. [Ramnandan et al., 2015]. The authors also showed in their evaluation that the Kolmogorov-Smirnov (KS) test performs best for this particular setup compared to tests such as Welch's t-test or Mann-Whitney's U-test.

Kolmogorov-Smirnov (KS) distance: The KS test is a non-parametric test which quantifies the distance between two empirical distribution functions with the advantage of making no assumptions about the distribution of the data. As a distance measure between two samples, the KS test computes the KS-statistic D for two given cumulative distribution functions F_1 and F_2 in the following way:

$$D = \underset{x}{sup}|F_1(x) - F_2(x)| \tag{5.1}$$

where *sup* is the supremum of the distances. If two samples are equally distributed, i.e., the two bags hold the same numeric values, then the statistic D converges to 0.

Aggregation function

As in the K-nearest neighbour classification, we also aggregate the top-k nearest neighbours by their properties, types and property object pairs. This allows us to classify the input values at several levels:

Before we apply the specific voting function, we aggregate the neighbours for the following different levels:

- `property level`: aggregation of the top-k neighbours by their properties
- `exact type level`: aggregation of the top-k neighbours by their exact type
- `root type level`: aggregation of the top-k neighbours by their root type
- `all types level`: aggregation of the top-k neighbours by each of their types (including the exact and all super types)
- *p–o* `level`: aggregation of the top-k neighbours by each of their *p-o* nodes

We consider the following two aggregation functions:

- *Majority vote:* This is the standard method for the K-nearest neighbour classification for which the input values are classified based on a majority vote over the k nearest neighbours. Therefore, given an aggregation level, we rank the aggregated results (e.g. properties) based on the appearance in the top-k neighbours. Consider the right part of Figure 5.3 in which we illustrate such a ranking process. For instance, the property aggregation would rank p_1 higher than p_2 since p_1 appears three times in comparison to p_2 which only appears 2 times.
- *Aggregated distance:* Our second aggregation function, we rank the aggregated results not by the number of their appearances, but compute the average distance. For instance, we would compute the distance for p_1 in Figure 5.3 by averaging the distance of node P_3, P_2 and T_1.

In addition to the aggregation of properties, types and property-object pairs, we can also perform a nested level aggregation. For instance, we could aggregate first on the property level and then inside each property on the type level. An example for the nested aggregation based on the majority vote is depict in Figure 5.3; the most likely type for p_1 would be t_1 with 2 votes, followed by t_2 and t_3.

5.1.4 Evaluation & Experiments

We have implemented a prototype system in Python to evaluate our approach with different functions. As a dataset to construct our background knowledge we use the DBpedia 3.9 dump.[2] The aim of our evaluation is twofold: We first automatically evaluate the accuracy of our prediction functions with different setups of the background knowledge in a controlled environment by splitting the DBpedia data into a test and training dataset. Secondly, we manually test our approach over Open Data CSV files to gain first insights for future directions, whether there is a chance to label tabular columns outside of DBpedia.

Background Knowledge Construction

We selected 50 of the the most frequently used numerical DBpedia properties to build our background knowledge for both evaluation scenarios:we excluded properties which clearly indicate internal DBpedia IDs only, such as `dbo:wikiPageRevisionID` as well as properties which are not directly in the root path of the `http://dbpedia.org/ontology/` prefix. Figure 5.4 plots in the left figure the 5% to 95% inter-quantile ranges of our selected properties (in logarithmic scale) and in the right figure the total number of numeric values for each property. The range plot visualises the overlap of numerical values for our different properties and the quantiles are used to smoothen the ranges and eliminate possible outliers. About 60% of the properties have values within the range 0-1000 and about 90% within 0-2000.[3] The shortest range has the property `dbp:displacement` (inter-quantile range of 0.0058) and the maximum range of 2.56 billion has the property `dbp:areaTotal`. Regarding the total number of values, the longest bar, with 421k values, corresponds to the `dbp:years` property and the shortest to `dbp:width` (9.6k values).

We built three versions of our background Knowledge Graph to better understand the impact of the three different distance functions. One function is based on the Kolmogorov-Smirnov distribution test, and two based on the euclidean distance over feature vectors. The first type of vector uses the minimum and maximum of the values as features while the other uses the 5% and 95% quantile as features. We add to both vectors the mean and standard deviation as additional dimensions. Table 5.2 gives an overview of our three knowledge bases together with the number of nodes, the construction time of the

[2] `http://downloads.dbpedia.org/3.9/en/mappingbased_properties_en.nt.bz2`, last accessed 2019-06-20

[3] Note, that around 30% of the properties have values in the range of 1000-2000 and mainly describe years.

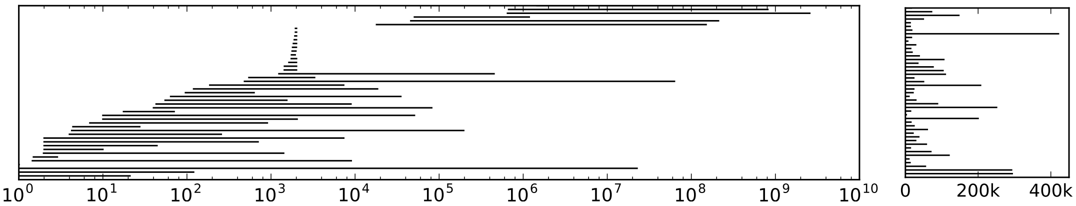


Figure 5.4: 5%-95% inter-quantile ranges and number of values of training properties.

ID	DISTANCE MEASURE	NODES	BUILD TIME	AVG. PRED. TIME
KS	Kolmogorov-Smirnov test	11431	30m	2.5s
FV1	(min, max, mean, std)	11432	24m	2.3s
FV2	(5-q, 95-q, mean, std)	11432	38m	4.6s

Table 5.2: Setup of our three background Knowledge Graphs

background Knowledge Graph and the average prediction time for a given set of values (based on our evaluation runs).

In addition we added our average prediction times for the different setups. However, please note that we did not optimize our system wrt. runtimes. In future work we plan the improvement of these prediction times in order to provide our algorithm as a live service. All evaluation are conducted on a machine with 30GB of RAM.

Model Evaluation

Our first experiment is designed to obtain the performance characteristics of our prediction for different distance functions.

Test & training data selection: To get an unbiased assessment we randomly assigned 20% of the subjects for each property as test data and the remaining subjects are used to build the knowledge bases. The test data is further processed to find suitable test groups. To build those test-groups, we proceed in a similar manner as for the construction of the background knowledge base. That is, we analogously built type hierarchy and *p-o* hierarchy for per property, however, this time without imposing any constraints and creating all possible test contexts and sub-contexts. Eventually, we randomly select the leaf nodes of this “test context graph" and the respective numerical value bags as test data. This process ensures that we select context nodes which are not necessarily contained 1-to-1 in the background Knowledge Graph.

Evaluating distance functions: Our first evaluation aims to i) test the impact of the distance function for the prediction and ii) to select the best setup for further tests.

	FV1			FV2			KS		
	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10
`exact`	2.5	8.2	8.2	2.5	8.2	8.2	**12.3**	**41.8**	**47.9**
`prop`	45.4	60.3	60.3	45.4	60.3	60.3	**57.1**	**74.1**	**79.8**
`type`	11.3	24.9	24.9	11.3	24.9	24.9	**16.1**	**43.9**	**56.0**
`stype`	24.9	41.1	41.1	24.9	41.1	41.1	**35.8**	**58.6**	**67.5**

Table 5.3: Accuracy in % for different distance functions

We set up an initial experiment by randomly selecting a maximum of 50 leave nodes from each property tree in our test dataset; resulting in 1787 test nodes.

To initially measure the accuracy of the top-k neighbours, we introduce the following evaluation measures:

- `exact`: the top-k neighbours contain the *correct node* in the graph, that is, the test node and predicted node share the same property, type and *p-o* pairs.
- `prop`: the top-k neighbours contain the *correct property/label*
- `type`: the top-k neighbours contain the *correct type*
- `stype`: the top-k neighbours contain the *correct super type* of the test node

The results in Table 5.3 show the accuracy for different metrics for the top-k neighbours, with the best results marked bold. We can clearly see that the Kolmogorov-Smirnov based distance function (`KS`) outperforms for all metrics the feature vectors based functions in terms of prediction accuracy. The initial results show that our approach already predicts the correct property of the input values in the top-10 neighbours for 79% of all test and the right type in 56% of the cases. Based on the clear results, we decided to use the prediction approach based on Kolmogorov-Smirnov distance function in the remaining evaluation.

Large-scale model validation: The next experiment focuses on the evaluation of the different aggregation functions and levels. We randomly sampled 33657 test nodes by selecting a maximum of 20% of the leave nodes for each property in our test data set. The test data is $\sim$ 18 times larger than in the previous experiment and 3 times the size of our training nodes. In addition, only 9% of the test context nodes are contained 1-to-1. This allows us to study our approach for input data for which we have only partial evidences available. We evaluate the accuracy for the different levels by measuring if the top-k aggregated results contain the correct property, type, parent types or any *p-o* context of the test instance.

top-*k*		`prop`		`type`		`all-types`		`root-type`		`p-o level`	
neigh.	agg.	maj.	avg.	maj.	avg.	maj.	avg.	maj.	avg.	maj.	avg.
25	1	59.3	34.5	64.7	57.8	64.7	57.8	66.4	69.2	20.4	24.9
	5	87.9	82.9	91.4	85.3	91.4	85.3	94.7	94.7	75.8	66.2
	10	98.5	98.5	94.7	94.7	94.7	94.7	94.7	94.7	83.8	74.0
50	1	57.4	23.7	66.4	37.6	66.4	37.6	66.7	70.7	20.4	24.9
	5	98.4	83.7	93.2	65.4	93.2	65.4	96.3	96.3	75.8	66.2
	10	99.3	99.3	96.3	96.1	96.3	96.1	96.3	96.3	83.8	74.0

Table 5.4: Accuracy in % for different aggregation levels and functions (agg. = aggregated results, maj. = majority vote, avg. = average distance)

Table 5.4 summarises the accuracy (in %) over 33k test instances for two aggregation functions over the top-k nearest neighbours. Overall, the results show a high prediction accuracy of over 92% across all different levels for the top-10 aggregated results using the top-50 closest neighbours. For the root-type prediction, we observe the highest accuracy within the top-5 aggregated results. Regarding test nodes for which we have only partial information available, our approach can still predict the correct property, (parent) type and even some of the shared *p-o* pairs. Our results also show that doubling the number of neighbours significantly improves the prediction accuracy by up to 15%. Considering the two aggregation functions, we see that ranking the results based on majority votes performs slightly better than using the average distance, with the biggest impact for the *p-o* level aggregation. Interestingly, inspecting the top-1 aggregated results using the 50 nearest neighbours, we see that the `root-type` accuracy is lower than the `all types` accuracy. This is a false negative classification which can happen if there exists more than *k* results with equal votes or average distances. In such case, we rank the results in alphabetical order and return only the top-k, leading to a cutoff of possible correct results.

Looking at the top-10 of the aggregated results, we correctly predicated 99.5% of the properties, 96.3% of the exact and parent types and 92% of the *p-o* pairs. These results are encouraging to use our approach for labelling numerical columns in tabular data, especially since we can also partially label values for which we do not have full evidences in our background Knowledge Graph.

Semantic labelling of numerical columns in Open Data tables

Eventually, we study how our approach performs for numerical columns in Open Data tables. We have to emphasise upfront, that this experiment is of rather exploratory than quantitative nature, since - due to the heterogeneity of data typically published in Open Data portals vs. DBpedia, we could not expect a lot of exact matches.

To conduct our experiment, we downloaded and parsed in total 1343 CSV files from two

Portal	Tables	$\overline{cols}$	$\overline{num.cols}$	w/o Header	Num. H.	Mapped
AT	968	13	8	154	6,482	1,323
EU	357	20	4	223	1,233	349

Table 5.5: Header mapping of CSVs in Open Data portals

Open Data portals, namely the Austrian Open Government Data portal (AT[4]) and the European Open Data portal (EU[5]). We used the standard Python CSV parser to analyse the tables for missing header rows and performed a simple datatype detection to identify numerical columns. In order to get insights into the descriptiveness of these headers we tried to map header labels to BabelNet [Navigli and Ponzetto, 2012] in a non-restrictive manner: we performed a simple preprocessing on the headers (splitting on underscores and camel case) and retrieved all possible mappings from the BabelNet API.

Table 5.5 shows some basic statistics of the CSV tables in the two portals. An interesting observation is that the AT portal has an average number of 20 columns per table with an average of 8 numerical columns, while the EU portal has larger tables with an average of 4 out of 20 columns being numerical. Regarding the descriptiveness of possible column headers, we observed that 28% of the tables have missing header rows. Eventually, we extracted headers from 7714 out of around 10K numerical columns and used the BabelNet service to retrieve possible mappings. We received only 1472 columns mappings to BabelNet concepts or instances, confirming our assumption that many headers in Open Data CSV files cannot easily be semantically mapped.

Exploratory experiments: We used the numerical columns from our CSV corpus as input for our system and manually study select columns to gain first insights. Initially, we ranked the columns by their average distance over the 50 nearest neighbours and inspected the top-100 columns for each portal. We share the interesting results for tables and columns online.[6]

Our first observation observation is related to the time coverage of numerical values and the difference between Open Data and DBpedia. For instance, the Austrian Open Data portal hosts tables as specific as numbers of cars per brands per district in Vienna, or current (every 15min) weather data from different weather stations in Austria. We do not expect matches for such specific numbers or even for temperature values if they are given in the form of timelines. In contrast, DBpedia typically has numeric values only for “current” or “latest” for many properties, taking population numbers of settlements as an example. Still, we are curious to see what the method would return and partially could explore interesting findings.

[4] http://data.gv.at/

[5] http://open-data.europa.eu/

[6] http://data.wu.ac.at/iswc2016_numlabels/

Another observation is that our knowledge base does not cover some of the domains and attributes of the numerical Open Data columns. For instance, many columns describe "counts" or "statistics". Examples for such count columns are the number of registered car model per district, the count of tourists grouped by their nationality, month of year and country/region they visit or the count of valid or invalid votes for an election. Examples for statistics are election results or the percentage of registered people for different age groups and districts in a city. For instance, take the 14th ranked Column#14[7] which describes election results divided by different regions, with a non-descriptive header `UNG.` (we assume this means "invalid votes"). The second-ranked property is *populationTotal* which is arguable a related labelling, since election results are basically sub-populations of different regions. Looking at the results of the type aggregation for this column, we find five times the type *Settlement* within the first ten neighbours, which further indicates that the values rise from (sub-)populations. Similarly, Column#1[8] holds counts of car models grouped by regions which our algorithm again labelled as population. This shows clearly that to label Open Data columns we need a very broad coverage of numerical domains in our background knowledge.

We also aggregate the results across columns to identify the "domain" of a table using the top-10 results of our `all-types level` aggregation and manually inspected some results. Again, we ranked the tables based on their average distances across all their numerical columns. For instance, consider the second ranked Table#2[9] which consists of multiple columns which describe population counts for different districts. Aggregating and ranking the types across these columns results in the types *Town* and *PopulatedPlace* which proved to be right.

Discussion of Findings

While the findings did not yet provide, clear and convincing matches, we could collect valuable insights from this results on challenges to be tackled in future work:

- *Dealing with timeline data:* To correctly handle timeline data, we first need to be able to detect the time dependency and than regroup or transform the table.

- *Domain specific background knowledge:* Open Data contains many tabular data which is similar in itself, but not necessarily matching DBpedia categories and values reported there, e.g. reports for spendings/budget election results, tourism or population demographics. Our results highlight the limited coverage of DBpedia, which was also observed in the work from Ritze et al. [Ritze et al., 2016]. Therefore, we have to gradually enrich the background Knowledge Graph from categories learned from Open Data tables themselves.

[7] http://data.wu.ac.at/iswc2016_numlabels/submission/col14.html
[8] http://data.wu.ac.at/iswc2016_numlabels/submission/col1.html
[9] http://data.wu.ac.at/iswc2016_numlabels/submission/tab2.html

- *Aggregating column scores:* While single columns provided partially bad recognition, in some cases combined recognition of columns revealed interesting combinations.
- *Combine with existing complementary approaches:* Lastly, while we deliberately left it out of scope in these experiments, linguistic cues could and probably should be used in combination with our methods as an additional cue to gradually improve labelling/matching capabilities.

5.2 Spatio-Temporal Labelling of Datasets

We derive our next steps directly from the findings in Section 5.1: Now we focus on certain, specific, dimensions, and integrate linguistic cues for labelling the data; also, we solve the problem of de-composing and understanding date-time information. Intuitively, most datasets found in Open Data – as it is mostly regional/national census-based – are organised by spatio-temporal scopes, that is, single datasets provide data for a certain region, and are valid for a certain time period; our goal is to develop a semantic labelling algorithm to cover exactly these two prevalent dimensions.

Recently, Kacprzak et al. [Kacprzak et al., 2017] confirmed the relevance and need of spatio-temporal annotations and search across Open Data portals: They analyzed the query logs of four data portals (including data.gov.uk) wrt. different aspects and characteristic and list temporal and geospatial queries as the top-two query types.

In the following, we present a scalable approach to (i) construct a spatio-temporal Knowledge Graph that hierarchically structures geographical entities, as well as temporal entities, and (ii) annotate a large corpus of tabular Open Data, currently holding datasets from eleven European (governmental) data portals.

In more detail, we present the following concrete contributions:

- A detailed *construction of a hierarchical Knowledge Graph of geo-entities and temporal entities* and links between them.
- A scalable *labelling algorithm for linking open datasets* (both on a dataset-level and on a record-level) to this Knowledge Graph.
- Indexing and annotation of datasets and metadata from 11 Open Data portals from 10 European countries and an *evaluation of the annotated datasets* to illustrate the feasibility and effectiveness of the approach.
- Code, data and a description on how to re-run our experiments, which we hope to be a viable basis for further research extending our results, are available for re-use (under GNU General Public License v3.0).[10]

[10] https://github.com/sebneu/geolabelling/

In the next section (Section 5.2.1) we introduce (linked) datasets, repositories and endpoints to retrieve relevant temporal and spatial information. Section 5.2.2 provides a schematic description of the construction and integration of these sources into our *base* Knowledge Graph – a constructed Knowledge Graph which serves as a basis for annotation and linking of the datasets; its actual realization in terms of implementation details is fully explained in Appendix A.3. In Section 5.2.3 we present the algorithms to add spatio-temporal annotations to datasets from Open Data portals, and evaluate and discuss the performance (in terms of precision and recall based on a manually generated sample) and limitations of our approach.

5.2.1 Existing Repositories of Temporal- and Geo-entities

When people think of spatial and temporal context of data, they usually think of *concepts* rather than numbers, that is "countries" or "cities" instead of coordinates or a bounding polygon, or an "event" or "time period" instead of e.g. start times end times. In terms of dataset search that could mean someone searching for datasets containing information about demographics for the last government's term (or in other words between the last two general elections).

In order to enable such search by spatio-temporal concepts, our goal is to build a concise, but effective knowledge base, that collects the relevant concepts from openly available data into a coherent Knowledge Graph, for both (i) enabling spatio-temporal search within Open Data portals and (ii) interlinking Open Data portals with other datasets by the principles of Linked Data.

The following section gives an overview of datasets and sources to construct the base Knowledge Graph of temporal- and geo-entities, namely the geo-data sources GeoNames, OpenStreetMap and NUTS, the knowledge bases Wikidata and DBpedia, and the periods/events dataset PeriodO.

GeoNames.org The GeoNames database contains over 10 million geographical names of entities such as countries, cities, regions, and villages. It assigns unique identifiers to geo-entities and provides a detailed hierarchical description including countries, federal states, regions, cities, etc. For instance, the GeoNames-entity for the city of Munich[11] has the parent relationship "Munich, Urban District", which is located in the region "Upper Bavaria" of the federal state "Bavaria" in the country "Germany", i.e. the GeoNames database allows us to extract the following hierarchical relation for the city of Munich:

$$\begin{aligned}&\textit{Germany} > \textit{Bavaria} > \textit{Upper Bavaria}\\&> \textit{Munich, Urban District} > \textit{Munich}\end{aligned}$$

The relations are based on the GeoNames ontology[12] which defines administrative divisions (first-order `gn:A`, second-order `gn:A.ADM2`, until `gn:A.ADM5`)[13] for countries, states,

[11] `http://www.geonames.org/6559171/`

[12] `http://www.geonames.org/ontology/ontology_v3.1.rdf`

[13] Here, `gn:` corresponds to the namespace URL `http://www.geonames.org/ontology#`

cities, and city districts/sub-regions. In this work we make use of an RDF dump of the GeoNames database,[14] which consists of alternative names and hierarchical relations of all the entities.

OpenStreetMap (OSM) OSM[15] was founded in 2004 as a collaborative project to create free editable geospatial data. The map data is mainly produced by volunteers using GPS devices (on foot, bicycle, car, ..) and later by importing commercial and government sources, e.g., aerial photographies. Initially, the project focused on mapping the United Kingdom but soon was extended to a worldwide effort. OSM uses four basic "elements" to describe geo-information:[16]

- *Nodes* in OSM are specific points defined by a latitude and longitude.
- *Ways* are lists of *nodes* that define a line. OSM ways can also define areas, i.e. a "closed" way where the last node on the way equals to the first node.
- *Relations* define relationships between different OSM elements: They either split long ways into smaller segments (for easier processing), or build complex objects, e.g., a *route* is defined as a relation of multiple ways (such as highway, cycle route, bus route) along the same route.
- *Tags* are used to describe the meaning of any elements, e.g., there could be a tag `highway=residential`[17] (tags are represented as key-value pairs) which is used on a *way* element to indicate a road within settlement.

There are already existing works which exploit the potential of OSM to enrich and link other sources. For instance, in [Posada-Sánchez et al., 2016] we have extracted indicators, such as the number of hotels or libraries in a city, from OSM to collect statistical information about cities.

Likewise, the software library *Libpostal*[18] uses addresses and places extracted from OSM: it provides street address parsing and normalization by using machine learning algorithms on top of the OSM data. The library converts free-form addresses into clean normalised forms and can therefore be used as a pre-processing step to geo-tagging of streets and addresses. We integrate Libpostal in our framework in order to detect and filter streets and city names in text and address lines.

[14] `http://www.geonames.org/ontology/documentation.html`, last accessed 2018-01-05

[15] `https://www.openstreetmap.org/`

[16] A detailed description can be found at the OSM documentation pages: `http://wiki.openstreetmap.org/wiki/Main_Page`

[17] cf. `https://wiki.openstreetmap.org/wiki/Tag:highway=residential`

[18] `https://medium.com/@albarrentine/statistical-nlp-on-openstreetmap-b9d573e6cc86`, last accessed 2017-09-12

Sources to obtain Postal codes and NUTS codes Postal codes are regional codes consisting of a series of letters (not necessarily digits) with the purpose of sorting mail. Since postal codes are country-specific identifiers, and their granularity and availability strongly varies for different countries, there is no single, complete, data source to retrieve these codes. The most reliable way to get the complete dataset is typically via governmental agencies (made easy, in case they publish the codes as open data).[19] Another source worth mentioning for matching postal codes is GeoNames: it provides a collection of postal codes for several countries and the respective name of the places/districts.[20]

Partially, postal codes for certain countries are available in the knowledge bases of Wikidata and DBpedia (see below) for the respective entries of the geo-entities (using "postal code" properties). However, we stress that these entries are not complete, i.e., not all postal codes are available in the knowledge bases as not all respective geo-entities are present, and also, the codes' representation is not standardised.

NUTS (French: nomenclature des unitès territoriales statistiques). Apart from national postal codes another geocode standard has been developed and is being regulated by the European Union (EU). It references the statistical subdivisions of all EU member states in three hierarchical levels, NUTS 1, 2, and 3. All codes start with the two-letter ISO 3166-1 [iso, 2013] country code and each level adds an additional number to the code. The current NUTS classification lists 98 regions at NUTS 1, 276 regions at NUTS 2 and 1342 regions at NUTS 3 level and is available from the EC's Webpage.[21] Also worth mentioning in this context – as an additional source for statistical and topographical maps on NUTS regions – are the basemaps developed at the European level by Eurostat, available as REST services.[22]

Wikidata and DBpedia These domain-independent, multi-lingual, knowledge bases provide structured content and factual data. While DBpedia [Lehmann et al., 2015b] is automatically generated by extracting information from Wikipedia, Wikidata [Vrandecic and Krötzsch, 2014], in contrary, is a collaboratively edited knowledge base which is intended to provide information to Wikipedia. These knowledge bases already partially include links to GeoNames, NUTS identifier, and postal code entries, as well as temporal knowledge for events and periods, e.g., elections, news events, and historical epochs, which we also harvest to complete our Knowledge Graph.

PeriodO The PeriodO project [Golden and Shaw, 2016] offers a gazetteer of historical, art-historical, and archaeological periods. The user interface allows to query and filter the

[19] For instance, the complete list of Austrian postal codes is available as CSV via the Austrian Open Data portal: `https://www.data.gv.at/katalog/dataset/f76ed887-00d6-450f-a158-9f8b1cbbeebf`, last accessed 2018-04-03

[20] `http://download.geonames.org/export/zip/`, last accessed 2018-01-05

[21] `http://ec.europa.eu/eurostat/web/nuts/overview`, last accessed 2018-01-05

[22] `http://ec.europa.eu/eurostat/statistical-atlas/gis/arcgis/rest/services/Basemaps`, last accessed 2018-08-30

periods by different facets. Further, the authors published the full dataset as JSON-LD download[23] and re-use the W3C `skos`, `time` and `dcterms:spatial` vocabularies to describe the temporal and spatial extend of the periods. For instance, the (shortened) PeriodO entry in Figure 5.5 describes the period of the First World War.

```
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix skos:<http://www.w3.org/2004/02/skos/core#>
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix time: <http://www.w3.org/2006/time#> .

<http://n2t.net/ark:/99152/p0kh9ds3566>
  dcterms:spatial dbr:United_Kingdom ;
  skos:altLabel "First World War"@eng-latn ;
  time:intervalFinishedBy [
    skos:prefLabel "1918" ;
    time:hasDateTimeDescription [
      time:year "1918"^^xsd:gYear
    ]
  ] ;
  time:intervalStartedBy [
    skos:prefLabel "1914";
    time:hasDateTimeDescription [
      time:year "1914"^^xsd:gYear
    ]
  ] .
```

Figure 5.5: PeriodO entry for the period of World War I.

5.2.2 Base Knowledge Graph Construction

The previous section listed several geo-data repositories as well as datasets containing time periods and event data – some already available as Linked Data via an endpoint – which we use in the following to build up a spatio-temporal Knowledge Graph: Section 5.2.2 describes the extraction and integration of geospatial, and Section 5.2.2 of temporal knowledge. The remaining section uses an additional color coding of turquoise for introducing temporal and blue for geospatial properties.

Herein, we describe the composition of the graph by presenting *conceptual* SPARQL `CONSTRUCT` queries. This means that (most of) the presented queries cannot be executed because either there is no respective endpoint available or the query is not feasible and times out. While this section shall serve as a schematic specification of the constructed graph, we detail the actual realization of the queries in A.3.

Still, we deem the use of these conceptual SPARQL `CONSTRUCT` useful as a mechanism to declaratively express Knowledge Graph compilation from Linked Data sources, following

[23] `http://perio.do/`, last accessed 2018-03-27.

Rospocher et al.'s definition, who describe Knowledge Graphs as "a knowledge-base of facts about entities typically obtained from structured repositories"[Rospocher et al., 2016].[24]

The namespaces that are used in the SPARQL queries throughout the section (and the whole thesis) are listed in Appendix A.1.

Spatial Knowledge

Our Knowledge Graph of geo-entities is based on the GeoNames hierarchy, where we extract

- geo-entities and their labels,
- links to parent entities and particularly the containing country,
- coordinates in terms of points and (if available) geometries in terms of polygons,
- postal codes (again, if available), and
- sameAs-links to other sources such as DBpedia, OSM, or Wikidata (again, if available).

The respective SPARQL `CONSTRUCT` query[25] in Figure 5.6 displays how the hierarchical data can be extracted from the GeoNames datasets – loaded into a SPARQL endpoint – for a selected country `?c`: The GeoNames Ontology[26] allows to retrieve the relevant data for our Knowledge Graph per country, by replacing `?c` in this query with a concrete country URI, such as `http://sws.geonames.org/2782113/` (for Austria). The GeoNames RDF data partially already contains external links to DBpedia (using `rdfs:seeAlso`) which we model as equivalent identifiers using `owl:sameAs`. The hierarchy is constructed using the `gn:parentFeature` property. As GeoNames offers various different properties containing names, we extract all official English and (for the moment) German names, as we will use those later on for fueling our search index.

The query in Figure 5.7 then displays how we integrate the information in Wikidata into our spatial Knowledge Graph. In particular, Wikidata serves as a source to add labels and links for postal codes (`gn:postalCode`) and NUTS identifiers (`wdt:P605`) for the respective geographical entities. Further, we again add external links (to OSM and Wikidata itself) that we harvest from Wikidata as `owl:sameAs` relations to our graph.

[24] As a side remark, such queries could for instance be used to declaratively annotate the provenance trail of Knowledge Graphs compiled from other Linked Data sources, e.g. expressed through labelling the activity to extract the relevant knowledge with PROV's [Lebo et al., 2013] `prov:wasGeneratedBy` property with a respective SPARQL CONSTRUCT query.

[25] The plain queries are online at `https://github.com/sebneu/geolabelling/tree/master/jws_evaluation/queries`

[26] cf. `http://geonames.org/ontology/`

```
CONSTRUCT {                                                                     knowledge graph model
  ?geoentity rdfs:label ?label ; gn:parentFeature ?parent ; gn:parentCountry ?c ;
     gn:postalCode ?code ; geo:lat ?lat ; geo:long ?long ; owl:sameAs ?external .
} WHERE {                                                                                      labels
  ?geoentity gn:name ?label .
  OPTIONAL { ?geoentity gn:officialName ?label
    FILTER ( LANGMATCHES(LANG(?label), "EN") || LANG(?label) = "" ) }
  OPTIONAL { ?geoentity gn:alternateName ?label
    FILTER ( LANGMATCHES(LANG(?label), "EN") || LANGMATCHES(LANG(?label), "DE") || LANG(?label) = "" ) }
                                                                                           geospatial
  ?geoentity gn:parentCountry ?c ; geo:lat ?lat ; geo:long ?long .
  OPTIONAL { ?geoentity gn:parentFeature ?parent }
  # external links if available
  OPTIONAL { ?geoentity rdfs:seeAlso ?external }
  # postal code literals
  OPTIONAL { ?wd gn:postalCode ?code }
}
```

Figure 5.6: Conceptual SPARQL CONSTRUCT query to extract hierarchical data for our Knowledge Graph from GeoNames for a particular country ?c.

```
CONSTRUCT {                      knowledge graph model
  ?geoentity owl:sameAs ?wd ;
     gn:postalCode ?code;
     owl:sameAs ?osm ;
     owl:sameAs ?nuts .
  ?nuts wdt:P605 ?n .
} WHERE {                                   geospatial
  ?wd wdt:P1566 ?geoentity .
  # postal code literals
  OPTIONAL { ?wd wdt:P281 ?code }
  # NUTS identifier
  OPTIONAL { ?wd wdt:P605 ?n.
    BIND (CONCAT("<http://dd.eionet.europa.eu/vocabulary
          concept/common/nuts/", ?n,">") AS ?nuts) }
  # OSM relations
  OPTIONAL { ?wd wdt:P402 ?osm }
}
```

Figure 5.7: SPARQL query to extract Wikidata links and codes – times out on https://query.wikidata.org

```
PREFIX : <http://data.wu.ac.at/ns/osm#>

CONSTRUCT {                       knowledge graph model
  ?geoentity rdfs:label ?label;
    geo:lat ?lat; geo:long ?long ;
    gn:parentFeature ?parent;
    gn:parentCountry ?pc ;
    geosparql:hasGeometry ?geometry .
} WHERE {                                        labels
  [ :display_name ?label ;
                                             geospatial
    :osm_region ?parent ;
    :osm_id ?id ; :osm_type ?type ;
    :address [ :country ?country ];
    :lat ?lat ; :lon ?long ;
    :geojson [ :coordinates ?geometry] #this is simplifying!
  ] .
  ?pc gn:countryCode ?country .
  BIND(IRI(CONCAT(STR(osm:),?type,"/",?id)) AS ?geoentity)
}
```

Figure 5.8: Conceptual SPARQL query to extract data from OSM for a particular OSM entity with the OSM numeric identifier ?id.

The query in Figure 5.8 conceptually shows how and which data we extract for certain OSM entities into our Knowledge Graph. We note here that OSM does not provide an RDF or SPARQL interface, but the idea is that we - roughly - perceive and process data returned by OSM's Nominatim API in JSON as JSON-LD; details and pre-processing steps in A.3.2 below.

Temporal Knowledge

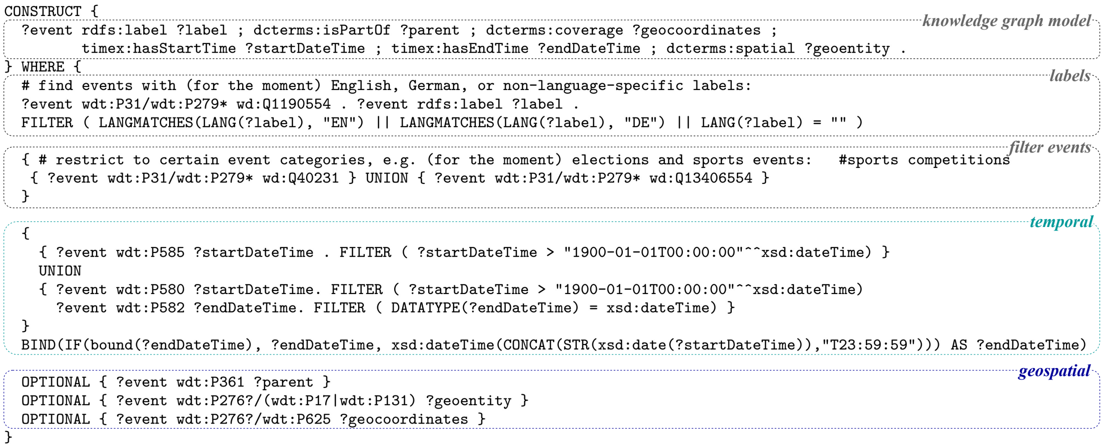
As for temporal knowledge, we aim to compile into our Knowledge Graph a base set of temporal-entities (that is, named periods and events from Wikidata and PeriodO) where we want to extract

- named events and their labels,
- links to parent periods that they are part of, again to create a hierarchy,

- temporal extent in terms of a single beginning and end date, and
- links to a spatial coverage of the respective event or period (if available).

We observe here that temporal knowledge is typically less consolidated than geospatial knowledge, i.e. “important” named entities in terms of periods or events are not governed by internationally agreed and nationally governed structures such as border-agreements in terms of spatial entities. Even worse, cross-cultural differences, such as different calendars or even time zones, add additional confusion. We still believe that the two integrated sources, which cover events of common interest in a multilingual setting on the one hand (Wikidata), and historical periods and epochs from the literature on the other (PeriodO), provide a good starting point.

In the future, it might be useful to also index news events, or recurring periods or points in time, such as public holidays, that occur regularly. However, we did not find any structured datasets available as linked data for that, so, we have to defer these to future work, or respectively, the creation of respective structured datasets as a challenge for the community. One obvious existing starting point here would be the work by Rospocher et al. [Rospocher et al., 2016] and the news events datasets they created in the EU Project NewsReader.[27] For the moment, we did not consider this work due to its fine granularity, which in our opinion is not needed in a majority of Open Data search use cases.


knowledge graph model
labels
filter events
temporal
geospatial

```
CONSTRUCT {
  ?event rdfs:label ?label ; dcterms:isPartOf ?parent ; dcterms:coverage ?geocoordinates ;
         timex:hasStartTime ?startDateTime ; timex:hasEndTime ?endDateTime ; dcterms:spatial ?geoentity .
} WHERE {
  # find events with (for the moment) English, German, or non-language-specific labels:
  ?event wdt:P31/wdt:P279* wd:Q1190554 . ?event rdfs:label ?label .
  FILTER ( LANGMATCHES(LANG(?label), "EN") || LANGMATCHES(LANG(?label), "DE") || LANG(?label) = "" )

  { # restrict to certain event categories, e.g. (for the moment) elections and sports events:   #sports competitions
   { ?event wdt:P31/wdt:P279* wd:Q40231 } UNION { ?event wdt:P31/wdt:P279* wd:Q13406554 }
  }

  {
    { ?event wdt:P585 ?startDateTime . FILTER ( ?startDateTime > "1900-01-01T00:00:00"^^xsd:dateTime) }
    UNION
    { ?event wdt:P580 ?startDateTime. FILTER ( ?startDateTime > "1900-01-01T00:00:00"^^xsd:dateTime)
      ?event wdt:P582 ?endDateTime. FILTER ( DATATYPE(?endDateTime) = xsd:dateTime) }
  }
  BIND(IF(bound(?endDateTime), ?endDateTime, xsd:dateTime(CONCAT(STR(xsd:date(?startDateTime)),"T23:59:59"))) AS ?endDateTime)

  OPTIONAL { ?event wdt:P361 ?parent }
  OPTIONAL { ?event wdt:P276?/(wdt:P17|wdt:P131) ?geoentity }
  OPTIONAL { ?event wdt:P276?/wdt:P625 ?geocoordinates }
}
```

Figure 5.9: Conceptual SPARQL query to extract event data (from elections and sports competitions) from Wikidata – times out on `https://query.wikidata.org`.

Again, we model the Knowledge Graph extraction and construction in terms of conceptual SPARQL queries: We use the query in Figure 5.9 to extract events information from Wikidata. Note, that this query times out on the public Wikidata endpoint. Therefore, in order to extract the relevant events and time periods as described in Figure 5.9, we converted a local Wikidata dump to HDT [Fernández et al., 2013], extracted only

[27] http://www.newsreader-project.eu/results/data/

the relevant triples for the query, materialised the path expressions, and executed the targeted CONSTRUCT query over these extracts on a local endpoint; the full details are provided in A.3.3. We do not just extract existing triples from the source, but try to aggregate/flatten the representation to a handful of well-known predicates from Dublin Core (prefix dcterms:) and the OWL time ontology (prefix time:).

Likewise, we use the query in Figure 5.11 to extract periods from the PeriodO dataset, again using the same flattened representation. To execute this query, in this case we could simply download the available PeriodO dump into a local RDF store.

Note that in these queries – in a slight abuse of the OWL Time ontology – we "invented" the properties timex:hasStartTime and timex:hasEndTime that do not really exist in the original OWL time ontology. This is a compromise for the desired compactness of representation in our Knowledge Graph, i.e. these are mainly introduced as shortcuts to avoid the materialization of unnecessary blank nodes in the (for our purposes too) verbose notation of OWL Time. A proper representation using OWL Time could be easily reconstructed by means of the CONSTRUCT query in Figure 5.10.

```
CONSTRUCT {
   ?X time:hasBeginning [
      time:inXSDDateTime ?startDateTime
   ] ;
      time:hasEnd [
        time:inXSDDateTime ?endDateTime
   ] .
} WHERE {
   ?X timex:hasStartTime ?startDateTime ;
      timex:hasEndTime ?endDateTime
}
```

Figure 5.10: CONSTRUCT query to reconstruct the OWL Time model of our flattened representation timex:hasStartTime and timex:hasEndTime.

For this purpose we define our own vocabulary extension of the OWL Time ontology, for the moment, under the namespace http://data.wu.ac.at/ns/timex#.

5.2.3 Dataset Labelling

In this section we first describe the algorithms to add geospatial annotations (Section 5.2.3) and to extract temporal labels and periodicity patterns (Section 5.2.3) and subsequently evaluate and discuss the performance – in terms of precision and recall based on a manually evaluated sample – and limitations of our approach in Section 5.2.3.

In order to add spatial and temporal annotations to Open Data resources we use the CSV files and metadata from the resources' data portals as signals. The metadata descriptions and download links are provided by our Open Data Portal Watch framework (cf. Section 3.3) which monitors and archives over 260 data portals, and provides

```
CONSTRUCT {
  ?event rdfs:label ?label ; dcterms:isPartOf ?parent ; dcterms:spatial ?geoentity ;      knowledge graph model
         timex:hasStartTime ?startDateTime ; timex:hasEndTime ?endDateTime .
} WHERE {
  {                                                                                        labels
   { ?event skos:prefLabel ?label } UNION { ?event skos:altLabel ?label } UNION { ?event rdfs:label ?label }
  }
  ?event time:intervalFinishedBy ?end ; time:intervalStartedBy ?start.                     temporal
  OPTIONAL{ ?end time:hasDateTimeDescription ?endTime .
    OPTIONAL{ ?endTime time:year ?endYear }
    OPTIONAL{ ?endTime periodo:latestYear ?endYear }
  }
  OPTIONAL{ ?start time:hasDateTimeDescription ?startTime .
    OPTIONAL{ ?startTime time:year ?startYear }
    OPTIONAL{ ?startTime periodo:earliestYear ?startYear }
  }
  OPTIONAL{ ?start (!periodo:aux)+ ?startYear. FILTER (isLiteral(?startYear)) }
  OPTIONAL{ ?end (!periodo:aux)+ ?endYear. FILTER (isLiteral(?startYear)) }
  FILTER( ?startYear >= "1900"^^xsd:gYear || xsd:integer(?startYear) >= 1900 ||
          ?endYear >= "1900"^^xsd:gYear || xsd:integer(?endYear) >= 1900 )

  BIND(xsd:dateTime(CONCAT(STR(?startYear),"-01-01T00:00:00")) AS ?startDateTime )
  BIND(xsd:dateTime(CONCAT(STR(?endYear),"-12-31T23:59:59")) AS ?endDateTime )
  OPTIONAL { ?event periodo:spatialCoverage ?geoentity }                                   geospatial
  OPTIONAL { ?event dcterms:spatial ?geoentity }
  OPTIONAL { ?event dcterms:isPartOf ?parent. }
}
```

Figure 5.11: SPARQL query to extract event data (from historic periods) from PeriodO.

APIs to retrieve their metadata descriptions in an homogenised way using the W3C DCAT vocabulary [Maali and Erickson, 2014]. Regarding the meta-information, we look into several available metadata-fields: we consider the title, description, the tags and keywords, and the publisher. For instance, the upper part of Figure 5.12 displays an example metadata description. It holds cues in the title and the publisher field (cf. "Veröffentlichende Stelle" - publishing agency) and holds a link to the actual dataset, a CSV file (cf. lower part in Figure 5.12), which we download and parse.

Geospatial Labelling

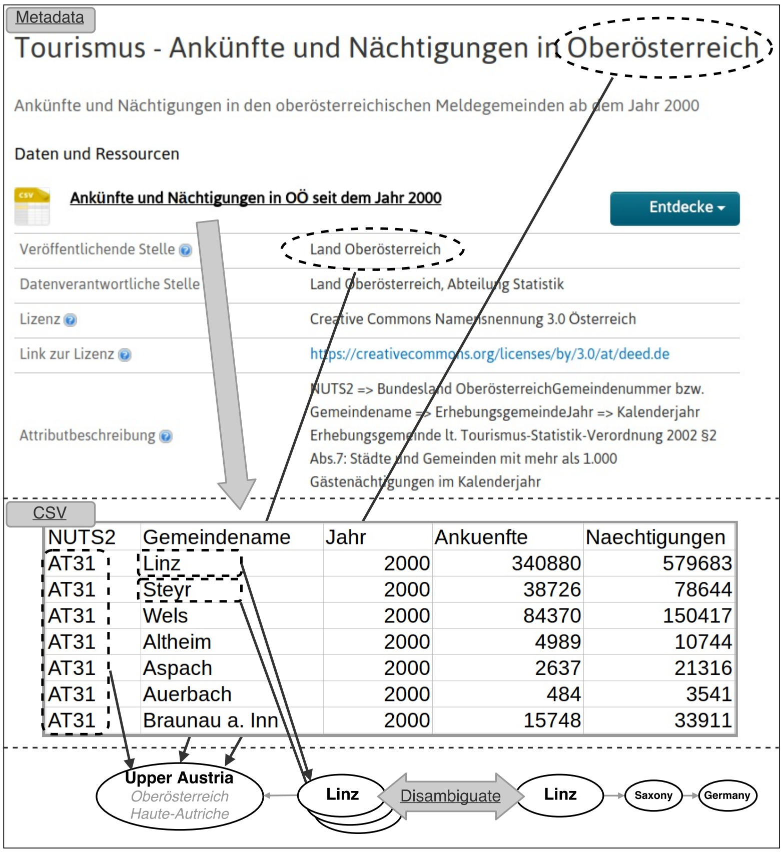
The geospatial labelling algorithm uses the different types of labels in our Knowledge Graph to annotate the metadata and CSV files from the input data portals.

CSVs: Initially, the columns of a CSV get classified based on regular expressions for NUTS identifier and postal codes. While the NUTS pattern is rather restrictive,[28] the postal codes pattern has to be very general, potentially allowing many false positives. Basically, the pattern is designed to allow all codes in the Knowledge Graph, and to filter out other strings, words, and decimals.[29]

Potential NUTS column (based on the regular expression) get mapped to the existing NUTS identifier. If this is possible for a certain threshold (set to 90% of the values) we consider a column as NUTS identifier and add the respective semantic labels. In case of potential postal codes the algorithm again tries to map to existing postal codes, however,

[28] [A-Z]{2}[A-Z0-9]{0,3}

[29] (([A-Z\d]){2,4}|([A-Z]{1,2}.)?\d{2,5}(\s[A-Z]{2,5})?(.[\d]{1,4})?)



Tourismus - Ankünfte und Nächtigungen in Oberösterreich

Ankünfte und Nächtigungen in den oberösterreichischen Meldegemeinden ab dem Jahr 2000

Daten und Ressourcen

Ankünfte und Nächtigungen in OÖ seit dem Jahr 2000

Entdecke ▾

Veröffentlichende Stelle	Land Oberösterreich
Datenverantwortliche Stelle	Land Oberösterreich, Abteilung Statistik
Lizenz	Creative Commons Namensnennung 3.0 Österreich
Link zur Lizenz	https://creativecommons.org/licenses/by/3.0/at/deed.de
Attributbeschreibung	NUTS2 => Bundesland OberösterreichGemeindenummer bzw. Gemeindename => ErhebungsgemeindeJahr => Kalenderjahr Erhebungsgemeinde lt. Tourismus-Statistik-Verordnung 2002 §2 Abs.7: Städte und Gemeinden mit mehr als 1.000 Gästenächtigungen im Kalenderjahr

NUTS2	Gemeindename	Jahr	Ankuenfte	Naechtigungen
AT31	Linz	2000	340880	579683
AT31	Steyr	2000	38726	78644
AT31	Wels	2000	84370	150417
AT31	Altheim	2000	4989	10744
AT31	Aspach	2000	2637	21316
AT31	Auerbach	2000	484	3541
AT31	Braunau a. Inn	2000	15748	33911

Figure 5.12: Geo-information in metadata and CSVs. Example dataset from the Austrian data portal: `https://www.data.gv.at/katalog/dataset/4d9787ef-e033-4c4f-8e50-65beb0730536`

we restrict the set of codes to the originating country of the dataset. This again results in a set of semantic labels which get accepted with a 90% threshold.

The labelling of string columns, i.e. set of words or texts, uses all the labels from GeoNames and OSM and is based on the following disambiguation algorithm:

Value disambiguation: The algorithm in Figure 5.13 shows how we disambiguate a set of string values based on the surroundings. As surroundings we consider all the values of a single column, however, in case of multiple labels in a row we use these as additional signals. E.g., consider a CSV row with the values “Austria”, “Linz”, and “Hauptplatz 1”, i.e., a row specifying and address, which we clearly want to consider for disambiguation.

First, the function `get_context(values)` counts all potential parent GeoNames entities for all of the values. To disambiguate a single value we use these counts and select the GeoNames candidate with the most votes from the context values’ parent regions; cf. `disamb_value(value)`. The function `get_geonames(value)` returns all potential GeoNames entites for an input string. Additionally, we use the origin country of the dataset (if available) as a restriction, i.e., we only allow GeoNames labels from the matching country.

For instance, in Figure 5.12 the Austrian “Linz” candidate gets selected in favor of the German “Linz” because the disambiguation resulted in an higher score based on the matching predecessors “Upper Austria” and “Austria” for the other values in the column (Steyr, Wels, Altheim, ...).

```
# disambiguate values based on ancestors
def disamb_values(values, country):
  disambiguated = []
  cont_par = get_context(values)
  for v in values:
    v_id = disamb_value(v, country, cont_par)
    disambiguated.append(v_id)
  return disambiguated

# disambiguate a single value based on the parents of the surrounding values
def disamb_value(value, country, cont_par):
  candidates = get_geonames(value)
  c_score = {}
  for c in candidates:
    if country != c.country:
      continue
    else:
      parents = get_all_parents(c)
      for p in parents:
        c_score[c] += cont_par[p]
  top = sorted(c_score)[0]
  return top

# counts parent values
def get_context(values):
  cont_par = {}
  for v in value:
    for c in get_geonames(value):
      parents = get_all_parents(c)
      for p in parents:
        cont_par[p] += 1
  return cont_par
```

Figure 5.13: Python code fragment for disambiguating a set of input values.

If no GeoNames mapping was found the algorithm tries to instantiate the string values with OSM labels from the Knowledge Graph. Again, the same disambiguation algorithm is applied, however, with the following two preprocessing steps for each input value:

1. In order to better parse addresses, we use the Libpostal library (cf. Section 5.2.1) to extract streets and place names from strings.
2. We consider the context of a CSV row, e.g., if addresses in CSVs are separated into dedicated columns for street, number, city, state, etc. To do so we filter the allowed OSM labels by candidates within any extracted regions from the metadata description or from the corresponding CSV row (if geo-labels available).

Metadata descriptions: The CSVs' meta-information at the data portals often give hints about the respective regions covering the actual data. Therefore, we use this additional source and try to extract geo-entities from the titles, descriptions and publishers of the datasets:

1. As a first step, we tokenise the input fields, and remove any stopwords. Also, we split any words that are separated by dashes, underscores, semicolon, etc.
2. The input is then grouped by word sequences of up to three words, i.e. all single words, groups of two words, ..., and the previously introduced algorithm for mapping a set of values to the GeoNames labels is applied (including the disambiguation step).

Figure 5.12 gives an example dataset description found on the Austrian data portal `data.gv.at`. The labelling algorithm extracts the geo-entity "Upper Austria" (an Austrian state) from the title and the publisher "Oberösterreich". The extracted geo-entities are added as additional semantic information to the indexed resource.

Temporal Labelling

Similarly to the geospatial cues, temporal information in Open Data comes in various forms and granularity, e.g., as datetime/timespan information in the metadata indicating the validity of a dataset, or year/month/time information in CSV columns providing timestamps for data points or measurements.

CSVs: To extract potential datetime values from the datasets we parse the columns of the CSVs using the Python dateutil library.[30] This library is able to parse a variety of commonly used date-time patterns (e.g., `"January 1, 2047"`, `"2012-01-19"`, etc.), however, we only allow values where the parsed year is in the range of 1900 and 2050.[31]

For both sources of temporal information, i.e. metadata and CSV columns, we store the minimum and maximum (or *start* and *end* time) value so that we can allow range queries over the annotated data.

[30] `https://dateutil.readthedocs.io/en/stable/`

[31] The main reason for this restriction is that any input year easily yields to wrong mappings of e.g. postal codes, counts, etc.

Datetime periodicity patterns: The algorithm in Figure 5.14 displays how we estimate any pattern of periodicity of the values in a column for a set of input datetime values. Initially, we check if all the values are the same (denoted as `static` column), e.g., a column where all cells hold "2018". Then we sort the values; however, note that this step could lead to unexpected annotations, because the underlying pattern might not appear in the unsorted column.

We compute all differences (`deltas`) between the input dates and check if all these deltas have approximately – with 10% margin – the same length. We distinguish `daily`, `weekly`, `monthly`, `quarterly`, and `yearly` pattern; in case of any other recurring pattern we return `other`.

```
def datetime_pattern(dates):
  # all the dates have the same value
  if len(set(dates)) == 1:
    return 'static'

  # sort the dates and compute the deltas
  dates = sorted(dates)
  deltas = [(d-dates[i-1]) for i, d in enumerate(dates)][1:]

  for p, l in [('daily', delta(days=1)),
               ('weekly', delta(days=7)),
               ('monthly', delta(days=30)),
               ('quarterly', delta(days=91)),
               ('yearly', delta(days=365))]:
    # add 10% tolerance range
    if all(l-(l*0.1) < d < l+(l*0.1) for d in deltas):
      return p

  # none of the pre-defined pattern
  if len(set(deltas)) == 1:
    return 'other'

  # values do not follow a regular pattern
  return 'varying'
```

Figure 5.14: Python code fragment for estimating the datetime patterns of a set of values.

Metadata descriptions: We extract the datasets' temporal contexts from the metadata descriptions available at the data portals in two forms: (i) We extract the `published` and `last modified` information in case the portal provides dedicated metadata fields for these. (ii) We use the resource title, the resource description, the dataset title, the dataset description, and the keywords as further sources for temporal annotations. However, we prioritise the sources in the above order, meaning that we use the temporal information in the resource metadata rather than the information in the dataset title or description.[32]

The datetime extraction from titles and descriptions is based on the Heideltime framework [Strötgen and Gertz, 2013] since this information typically comes as natural text. Heideltime supports extraction and normalization of temporal expressions from natural text for ten different languages. In case the data portal's origin language is not supported we use English as a fallback option.

Indexed Datasets & Evaluation

Our framework currently contains CSV tables from 11 European data portals from 10 different countries, cf. Table 5.6. We manually selected European governmental data portals (potentially also using NUTS identifier in their datasets) which are already monitored by the Open Data Portal Watch. Note, that the notion of *datasets* on these data portals (wrt. Table 5.6) usually groups a set of resources; for instance, typically a dataset groups resources which provide the same content in different file formats. A detailed description and analysis of Open Data portals' resources can be found in Chapter 3. The statistics in Table 5.6, i.e. the number of datasets and indexed CSVs is based on the third week of March 2018. The differing numbers of *CSVs* and *indexed* documents in the table can be explained by offline resources, parsing errors, etc. Also, we currently do not index documents larger than 10MB due to local resource limitations; the basic setup (using Elasticsearch for the indexed CSVs, cf. Section 5.3.2) is fully scalable.

portal	datasets	CSVs	indexed
total			15728
govdata.de	19464	10006	5646
data.gv.at	20799	18283	2791
offenedaten.de	28372	4961	2530
datos.gob.es	17132	8809	1275
data.gov.ie	6215	1194	884
data.overheid.nl	12283	1603	828
data.gov.uk	44513	7814	594
data.gov.gr	6648	414	496
data.gov.sk	1402	877	384
www.data.gouv.fr	28401	6038	258
opingogn.is	54	49	41

Table 5.6: Indexed data portals

[32] For instance, consider a dataset titled "census data from 2000 to 2010" that holds several CSVs titled "census data 2000", "census data 2001", etc.: This metadata allows to infer that the temporal cues in the CSVs' titles are more accurate/precise than the dataset's title, which gives a more general time span for all CSVs.

Table 5.7 lists the total number of annotated datasets. With respect to the spatial labelling algorithm, we were able to annotate columns of 3518 CSVs and metadata descriptions of 11231 CSVs (of a total of 15k indexed CSVs). For 3299 of the annotated CSVs our algorithm found GeoNames mappings, and for 292 OSM mappings. Regarding the temporal labelling, we detected date/time information in 2822 CSV columns and in 9112 metadata descriptions.

Spatial		*Temporal*	
Columns	**Metadata**	**Columns**	**Metadata**
3518	11231	2822	9112

Table 5.7: Total numbers of spatial and temporal annotations of metadata descriptions and columns.

Here we focus on evaluating the annotated geo-entities, and neglect the temporal annotations with the following two main reasons: First, the datetime detection over the CSV columns is based on the standard Python library `dateutil`. The library parses standard datetime formats (patterns such as `yyyy-mm-dd`, or `yyyy`) and the potential errors here are that we incorrectly classify a numerical column, e.g., classifying postal codes as years. As a very basic pre-processing, where we do not see a need for evaluation, we reduce the allowed values to the range 1900-2050 (with the drawback of potential false negatives), however, using the distribution of the numeric input values (cf. Section 5.1) would allow a more informed decision. Second, the labelling of metadata information is based on the temporal tagger Heideltime [Strötgen and Gertz, 2013] which provides promising evaluations over several corpora.

Manual inspection of a sample set: To show the performance and limitations of our labelling approach we have randomly selected 10 datasets per portal (using Elasticsearch's built-in random function[33]) and from these again randomly select 10 rows, which resulted in a total of 101 inspected CSVs,[34] i.e. 1010 rows (with up to several dozen columns per CSV). Sampling datasets from different portals allows us to assess and compare the performance for different countries and data publishing strategies. The median percentage of annotated records (i.e. rows) per dataset (across all indexed datasets) is 92%; our sample is representative, in this respect, with a median of 88% annotated rows. The median number of total rows of all indexed datasets is lower (86 rows) than within the evaluated sample (287 rows). However, the overall number of rows varies widely with a mean of 1742 rows across all datasets, which indicates a large variety and non-even distribution of dataset sizes (between 1 and 221k rows).

[33] `https://www.elastic.co/guide/en/elasticsearch/guide/current/random-scoring.html`, last accessed 2018-04-01

[34] We only selected CSVs with assigned geo-entities – to provide a meaningful precision measure – which resulted in < 10 files for the smaller data portals, e.g., opingogn.is, and therefore in 101 files in total.

As for the main findings, in the following let us provide a short summary; all selected datasets and their assigned labels can be found at `https://github.com/sebneu/geolabelling/tree/eu-data/jws_evaluation`.

Initially, we have to state that this evaluation is manually done by the authors and therefore restricted to our knowledge of the data portals' origin countries and their respective language, regions, sub-regions, postal codes, etc. For instance, we were able to see that our algorithm correctly labelled the Greek postal codes in some of the test samples from the Greek data portal `data.gov.gr`,[35] but that we could not assign the corresponding regions and streets.[36] However, as we are not able to read and understand the Greek language (and the same for the other non-English/German/Spanish portals) we cannot fully guarantee any potential mismatches or missing annotations that we did not spot during our manual inspections.

total	c	m	g	o
101	87	53	12	5

Table 5.8: Correctly assigned labels (c), missing annotations (m), incorrect links to GeoNames (g) or OSM (o) in the dataset.

We categorise the datasets' labels by assessing the following dimensions: are there any correctly assigned labels in the dataset (c), are there any missing annotations (m), and did the algorithm assign incorrect links to GeoNames (g) or OSM (o); a result overview is given in Table 5.8.

Out of 101 inspected datasets we identified in 87 CSVs correct annotations. In particular, for the Spain and the Greek data portal only in 50% of the test samples there were correct links, while for the 9 other indexed data portals we have a near to 100% rate. Regarding any missing annotations, we identified 53 datasets where our algorithm (and also the completeness of our spatial Knowledge Graph) needs improvements. For instance, in some datasets from the Netherlands' data portal[37] and also the Slovakian portal[38] we identified street names and addresses that could potentially mapped to OSM entries.

Regarding incorrect links there were only 12 files with wrong GeoNames and 5 files with wrong OSM annotations. An exemplary error that we observed here was that some files[39] contain a column with the value "Norwegen" ("Norway"): Since the file is provided at a

[35] E.g., `https://github.com/sebneu/geolabelling/blob/eu-data/jws_evaluation/data_gov_gr/0.csv`, the datasets use "T.K." in the headers to indicate these codes.

[36] The Greek data portal uses the Greek letters in their metadata and CSVs which would require a specialised label mapping wrt. lower-case mappings, stemming, etc.

[37] E.g.,`https://github.com/sebneu/geolabelling/tree/eu-data/jws_evaluation/data_overheid_nl/4.csv`

[38] E.g., `https://github.com/sebneu/geolabelling/tree/eu-data/jws_evaluation/data_gov_sk/3.csv`

[39] `https://github.com/sebneu/geolabelling/blob/eu-data/jws_evaluation/offenedaten_de/0.csv`

German data portal, we incorrectly labelled the column using a small German region Norwegen instead of the country, because our algorithm prefers labels from the origin country of the dataset. Another example that we want to consider in future versions of our labelling algorithm is this wrong assignment of postal codes:[40] We incorrectly annotated a numeric column with the provinces of Spain (which use two-digit numbers as postal codes).

Table 5.9 displays the *precision*, *recall*, and *F1 score* for all sample records, i.e. for all annotated cells of the 101 sample CSVs. We want to emphasise that these results do not say anything about the quality of the data portals themselves. As mentioned in the above paragraph, again, we can see in Table 5.9 that the Greek (data.gov.gr) and the Spain data portal (datos.gob.es) have a notable drop in precision[41] while for the other portals the *total* precision is still at 86%. The total recall is at 73%, which again shows that our approach needs further improvements in terms of missed annotations and completeness of the spatial Knowledge Graph.

portal	**precision**	**recall**	**F_1 score**
total	**.86**	**.73**	.79
govdata.de	.89	.67	.77
data.gv.at	1	.81	.90
offenedaten.de	.93	1	.96
datos.gob.es	**.51**	.91	.66
data.gov.ie	.98	.86	.92
data.overheid.nl	1	.29	.44
data.gov.uk	.98	.58	.73
data.gov.gr	**.51**	.64	.57
data.gov.sk	.82	.79	.81
www.data.gouv.fr	.98	.68	.81
opingogn.is	1	.72	.84

Table 5.9: Evaluation of the sample CSVs on record level.

Discussion and Analysis of Errors: To better understand the above results, and analyse our approach for future improvements, we list the common error patterns that we identified during the inspection of the sample set.

- *Incomplete External Knowledge:* While GeoNames and OSM are extremely rich and complete for certain countries and regions, there are countries, e.g. the Icelandic

[40] `https://github.com/sebneu/geolabelling/blob/eu-data/jws_evaluation/datos_gob_es/7.csv`

[41] There are streets in OSM that are labelled by an identifier (e.g. "2810 254 527") and, coincidentally, match columns in Greek datasets. Regarding the Spain datasets we incorrectly matched several columns containing the numbers 1-50: We mapped these to the fifty provinces of Spain, which use the numbers 1-50 as ID/zip codes. In future work we plan to include simple rules and heuristics to avoid such simple errors.

opingogn.is and the Greek data.gov.gr, where the recall is due to *missing street names and places in the knowledge bases.*

- *Incomplete Mapping of OSM:* As detailed in Section A.3.2, we first collect all administrative regions of a country, their subdivision, and so on, and then use their polygons to extract all OSM street names, and places. This, however, extracts only streets within these boundaries, while regions-to-streets is a n:m relation, i.e. *some streets cross several regions and therefore do not get mapped using this approach.* Also, we observed that some OSM entries are missing when using the OSM services, while they were findable via the UI. In future work, we want to improve this mapping, to integrate a more complete set of OSM entries.

- *Heuristics for Portal-Specifics:* Some of the wrong annotations are simple errors, e.g., allowing numbers as street names. To further increase the precision for specific portals we would have to implement heuristics based on characteristics of the datasets.

5.3 Query, Search and Visualization Interfaces

In this section we present our approach to enable structured, spatio-temporal search over Open Data catalogs through the spatio-temporal knowledge graph constructed in Section 5.2. The running prototype is online available at `https://data.wu.ac.at/odgraphsearch/`.

- A prototypical *search interface*, consisting of a web user interface allowing faceted and full-text search, a RESTful JSON API that allows programmatic access to the search UI, as well as API-access to retrieve the indexed dataset and respective RDF representations.

- A *SPARQL endpoint* that exposes the annotated links and allows structured search queries.

- A framework to automatically *generated visualisations* of open datasets based on queries for geo-entities: The showcase application displays automatically generated visualisations based on queries for geo-annotated datasets from the data portals.

The vocabularies and schema of our RDF data export are explained in Section 5.3.1 and the back-end, the search user interface and the SPARQL endpoint (including example queries) are presented in Section 5.3.2, and in Section 5.3.3 a web application to browse automatically generated dataset visualisations.

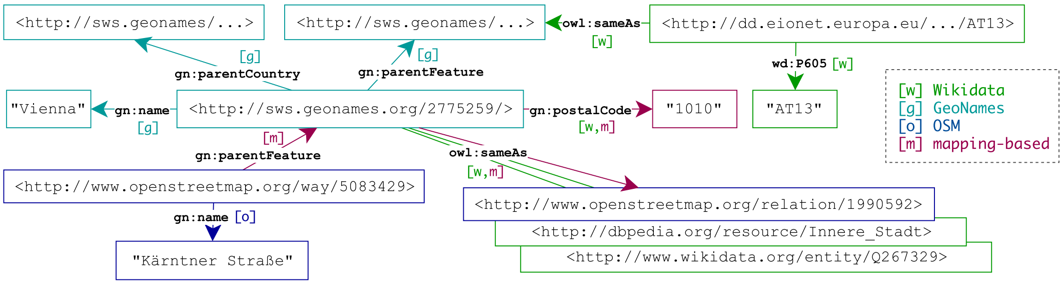

<http://sws.geonames/...>
<http://sws.geonames/...>
owl:sameAs
[w]
<http://dd.eionet.europa.eu/.../AT13>
[g]
gn:parentCountry
[g]
gn:parentFeature
wd:P605 [w]
"Vienna"
gn:name
[g]
<http://sws.geonames.org/2775259/>
gn:postalCode
[w,m]
"1010"
"AT13"
[w] Wikidata
[g] GeoNames
[o] OSM
[m] mapping-based
[m]
gn:parentFeature
owl:sameAs
[w,m]
<http://www.openstreetmap.org/way/5083429>
gn:name [o]
"Kärntner Straße"
<http://www.openstreetmap.org/relation/1990592>
<http://dbpedia.org/resource/Innere_Stadt>
<http://www.wikidata.org/entity/Q267329>

Figure 5.15: Example RDF export of the geo-entities knowledge graph.

5.3.1 Export RDF

We make our knowledge graph and RDFized linked data points from the CSVs available via a SPARQL endpoint. Figure 5.15 displays an example extract of the RDF export of the knowledge graph. The sources of the aggregated links between the different entities and literals in our graph are indicated in the figure; we re-use the GeoNames ontology (`gn:`) for the hierarchical enrichments generated by our algorithms (see links `[m]`), and `owl:sameAs` for mappings to OSM relations and NUTS regions, cf. Figure 5.15.

Annotated data points: We export the linked data points from the CSVs in two ways: First, for any linked geo-entity `<g>` in our knowledge graph, we add triples for datapoints uniquely linked in CSV resources (that is, values appearing in particular columns) by the following triple schema: if the entity `<g>` appears in a column in the given CSV dataset, i.e., the value $VALUE$ in that column has been labeled with `<g>`, we add a triple of the form

`<g>` `<`*u*#*col*`>` `"`$VALUE$`"` `.`

That is, we mint URIs for each column *col* appearing in a CSV accessible through a URL *u* by the schema *u*#*col*, i.e., using fragment identifiers. The column's name *col* is either the column header (if a header is available and the result is a valid URI) or a generic header using the columns' index `column1`, `column2`, etc. These triples are coarse grained, i.e. they do not refer to a specific row in the data. We chose this representation to enable easy-to-write, concise SPARQL queries like for instance:

```
SELECT ?name ?value
WHERE {
  ?geo <https://www.wien.gv.at/finanzen/ogd/
             hunde-wien.csv#Postal_CODE> ?value ;
       rdfs:label ?name .
}
```

The above query selects all values and their geo-annotations for the selected column named "Postal CODE" in a specific dataset about dog breeds in Vienna per district.

Second, a finer grained representation, which we also expose, provides exact table cells where a certain geospatial entity is linked to, using an extension of the CSVW vocabulary [Pollock et al., 2015]: note that the CSVW vocabulary itself does not provide means to conveniently annotated table cells in column *col* and row n which is what we need here, so we define our own vocabulary extension for this purpose (for the moment, under the namespace `http://data.wu.ac.at/ns/csvwx#`).

We use the CSVW class `csvw:Cell` for an annotated cell and add the row number and value (using `csvw:rownum` and `rdf:value`) to it. We extend CSVW by the property `csvwx:cell` to refer from a `csvw:Column` (using again the fragmented identifier *u*#*col*) to a specific cell, and the property `csvwx:rowURL` to refer to the CSV's row (using the schema *u*#row=n). Here, the property `csvwx:refersToEntity` (cf. the blue line) connects a specific cell to the labelled geo-entity <g>. Analogously, in case of available (labelled) temporal information for a cell, we use the property `csvwx:hasTime`; cf. the turquoise line in the following example:

```
@prefix csvwx: <http://data.wu.ac.at/ns/csvwx#> .
@prefix csvw: <http://www.w3.org/ns/csvw#> .
<u#col> csvwx:cell [
  a csvw:Cell ; csvw:rownum n ;
  csvwx:rowURL <u#row=n> ;
  rdf:value "VALUE" ;
  csvwx:refersToEntity <g> ;
  csvwx:hasTime "DATE"8sd:dateTime
] .
```

Moreover, we denote the geospatial scope of the column itself by declaring the type of entities *within* which geographic scope appearing in the column. The idea here is that we annotate – on column level – the least common ancestor of the spatial entities recognized in cells within this column. E.g.,

```
<u#col> csvwx:refersToEntitiesWithin <g1> .
```

with the semantics that the entities linked to *col* in the CSV u all refer to entities within the entity g_1 (such that g_1 is the least common ancestor in our knowledge graph.

This could be seen as a shortcut/materialization for a `CONSTRUCT` query as in Figure 5.16. Obviously, this query is very inefficient and we rather compute these least common ancestors per column during labelling/indexing of each column.

CSV on the Web: In order to complete the descriptions of our annotations in our RDF export, we describe all CSV resources gathered from the annotated Open Data Portals and their columns using the CSV on the Web (CSVW) [Pollock et al., 2015] vocabulary, re-using the following parts of the CSVW schema. Firstly, we use the following scheme to connect our aforementioned annotations to the datasets:

```
<d> a dcat:Dataset [ dcat:distribution
         [ dcat:accessURL u ] ].
```

```
CONSTRUCT
{ ?UrlCol csvwx:refersToEntitiesWithin ?G_1 }
WHERE {
 ?Col csvwx:cell [ csvwx:refersToEntity ?G ].
 ?G gn:parentFeature* ?G_1 .
 # all elements of this column have to share
 # parent feature ?G_1
 FILTER NOT EXISTS {
  ?Col csvwx:cell [ csvwx:refersToEntity ?G_ ].
  FILTER NOT EXISTS {
   ?G_ gn:parentFeature* ?G_1.
  }
 }
 # this parent feature is the least one that
 # fulfills this property:
 FILTER NOT EXISTS {
  ?G_2 gn:parentFeature ?G_1.
  ?Col csvwx:cell [ csvwx:refersToEntity ?G ].
  ?G gn:parentFeature* ?G_2 .
  # all elements of this column have to share
  # parent feature ?G_2
  FILTER NOT EXISTS {
   ?Col csvwx:cell [ csvwx:refersToEntity ?G__ ].
   FILTER NOT EXISTS {
    ?G__ gn:parentFeature* ?G_2.
   }
  }
 }
}
```

Figure 5.16: SPARQL `CONSTRUCT` query to materialize the geographic scope of a column.

```
[] csvw:url $u$;
  csvw:tableSchema [
   csvw:column (<$u$#$col_1$> <$u$#$col_2$>... <$u$#$col_n$>)] .

<$u$#$col_1$> a csvw:name "$col_1$" ; csvw:datatype $d_{col_1}$ .
<$u$#$col_2$> a csvw:name "$col_2$" ; csvw:datatype $d_{col_2}$ .
```

Then, we enrich this skeleton with further CSVW annotations that we can extract automatically from the respective CSV files. This process of generating CSVW metadata can be found in Section 4.3.

5.3.2 Search Prototype Setup

Our integrated prototype for a spatio-temporal search and query system for Open Data currently consists of three main parts: First, the geo-entities DB and search engine in the *back end*, second the *user interface* and *APIs*, and third, access to the above described RDF exports via an *SPARQL endpoint*.

Back End

All labels from all the integrated datasets and their corresponding geo-entities are stored in a look-up store, where we use the NoSQL key-value database MongoDB. It allows an easy integration of heterogeneous data sources and very performant look-ups of keys (e.g., labels, GeoNames IDs, postal codes, etc. in our case).

Further, we use Elasticsearch to store and index the processed CSVs and their metadata descriptions. In our setup, an Elasticsearch document corresponds to an indexed CSV and consists of all cell values of the table (arranged by columns), the potential geo-labels for a labelled column, metadata of the CSV (e.g., the data portal, title, publisher, etc.), the temporal annotations, and any additional labels extracted from the metadata.

The different components all have an impact on the performance and efficiency of the system. The indexing performance depends on the MongoDB database for label look-ups, the time-tagger Heideltime, and, Elasticsearch for storing the datasets. To show the efficiency and scalability of our approach, we timed the indexing of a sample of 2160 datasets, with an average file size of ∼50kB. The total processing time for all dataset was 16.8 hours – deactivated parallelization, including download, parsing, and processing time – whereof 8 hours were consumed by the labelling algorithms. Notably, the median total time for indexing a dataset is only 1.2 seconds, with a median time of 0.7 seconds for the labelling algorithms.[42]

User interface

The user interface, available at `https://data.wu.ac.at/odgraphsearch/`, allows search queries for geo-entities but also full-text matches. Note, that the current UI implements geo-entity search using auto-completion of the input (but only suggesting entries with existing datasets) and supports full-text querying by using the "Enter"-key in the input form. The screenshot in Figure 5.17 displays an example query for the Austrian city "Linz". The green highlighted cells in the rows below show the annotated labels, for instance, the annotated NUTS2 code "`AT31`" in the second result in Figure 5.17.

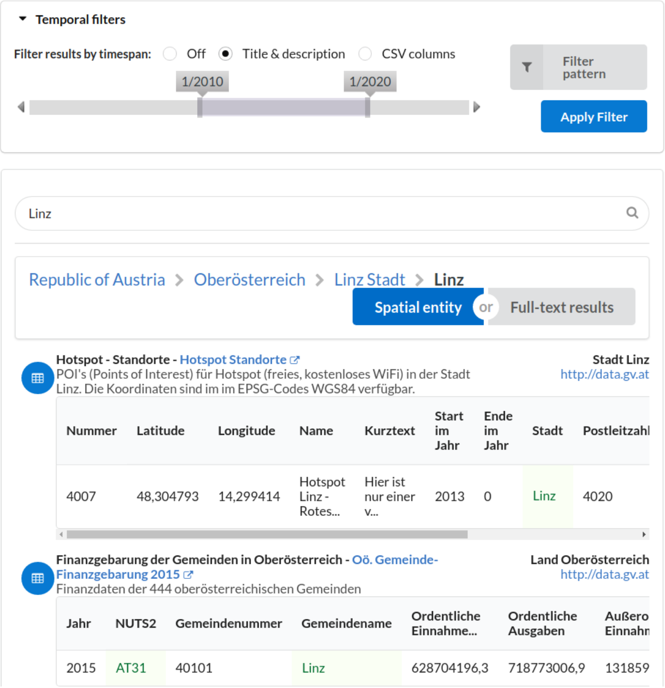
Also, we add facets to filter datasets relevant to a particular period either by auto-completion in a separate field to filter the time period by a period/event label, or by choosing start and end dates via sliders (cf. Figure 5.17). The users can decide to apply this filter to temporal information in title and description of the dataset, or the CSV columns.

By separating the search at these two levels we do not mix dates within the data and the metadata level. For instance, the metadata could have date/time that refers to the

[42] We deliberately discuss the median since the shape and size of the datasets can vary widely, which heavily influences the total and mean values.

present such as created, modified, etc. while in the datasets there can be a mixture of dates referring to temporal information or events (e.g., a column of birth dates).

The chosen geo-entities and durations which are fixed via these lookups in our search index through the UI are passed on as parameters as a concrete geo-ID and/or start&end-date to our API, which we describe next.



Temporal filters

Filter results by timespan: ○ Off ● Title & description ○ CSV columns

1/2010 1/2020

Linz

Republic of Austria > Oberösterreich > Linz Stadt > Linz

Hotspot - Standorte - Hotspot Standorte
POI's (Points of Interest) für Hotspot (freies, kostenloses WiFi) in der Stadt Linz. Die Koordinaten sind im im EPSG-Codes WGS84 verfügbar.

Stadt Linz
http://data.gv.at

Nummer	Latitude	Longitude	Name	Kurztext	Start im Jahr	Ende im Jahr	Stadt	Postleitzahl
4007	48,304793	14,299414	Hotspot Linz - Rotes...	Hier ist nur einer v...	2013	0	Linz	4020

Finanzgebarung der Gemeinden in Oberösterreich - Oö. Gemeinde-Finanzgebarung 2015
Finanzdaten der 444 oberösterreichischen Gemeinden

Land Oberösterreich
http://data.gv.at

Jahr	NUTS2	Gemeindenummer	Gemeindename	Ordentliche Einnahme...	Ordentliche Ausgaben	Außero Einnahn
2015	AT31	40101	Linz	628704196,3	718773006,9	131859

Figure 5.17: Screenshot of of an example search at the UI.

Additionally, the web interface provides APIs (available at `https://data.wu.ac.at/odgraphsearch/api`) to retrieve the search results, all indexed datasets, and the RDF export per dataset. To allow programmatic access to the search UI we offer the following HTTP GET API:

```
/api/v1/get/datasets?l={GeoIDs}
                      &limit={limit}&offset={offset}
                      &start={startDate}&end={endDate}
                      &mstart={startDate}&mend={endDate}
                      &periodicity={dateTimePattern}
                      &q={keyword}
```

The API takes multiple instances of geo IDs, that is, `GeoNames` or OSM IDs (formatted as `osm:{ID}`) using parameter `l`, a `limit` and an `offset` parameter, which restricts the amount of items (datasets) returned, and an optional white space separated list of keywords (`q`) as full-text query parameter to enable conventional keyword search in the metadata and header information of the datesets. To restrict the results to a specific temporal range we implemented the parameters `mstart`, `mend` (for filtering a time range from the metadata-information), and `start`, `end` (for the min and max values of date annotations from CSV columns). The parameter `periodicity` allows to filter for datetime periodicity patterns such as "yearly", "monthly", or "static" (in case there is only a single datetime value in this column), cf. Section 5.2.3 for a detailed description of the periodicity patterns.

The output consists of a JSON list of documents that contain the requested GeoNames/OSM IDs or any tables matching the input keywords.

SPARQL endpoint

We offer a SPARQL endpoint at `https://data.wu.ac.at/odgraphsearch/sparql` where we provide the data as described in Section 5.3.1. Currently, as of the first week of April 2018, the endpoint contains 88 million triples: 15 million hierarchical relations using the `gn:parentFeature` relation, 11768 CSVs (together with their CSV on the Web descriptions), where we added a total of 5 million geo-annotations using the `csvwx:refersToEntity` property, and 1.3 million datetime-annotations using the `csvwx:hasTime` annotation.

Example queries The first example in Figure 5.18 lists all datasets from Vienna, using the `csvwx:refersToEntity` metadata annotation, and only lists CSVs where there exists a column with dates within the range of the last Austrian legislative period, using the Wikidata entities of the past two elections.

The next example query in Figure 5.19 combines text search for time periods with a structured query for relevant data; it looks for information of datasets about a time period before the 2nd World War, called the "Anschluss movement" (i.e., the preparation of the annexation of Austria into Nazi Germany) and queries for all available CSV rows where a date within this period's range (1918-1938, according to PeriodO), and a geo-entity within the period's spatial coverage location (i.e. Austria) occurs.

GeoSPARQL GeoSPARQL [Perry and Herring, 2012] extends SPARQL to a geographic query language for RDF data. It defines a small ontology to represent geometries (i.e., points, polygons, etc.) and connections between spatial regions (e.g., contains, part-of, intersects), as well as a set of SPARQL functions to test such relationships. The example query in Figure 5.20 (namespaces as in Figure A.1) uses the available polygon of the Viennese district "Leopoldstadt" to filter all annotated data points within the borders of this district.

While we do not yet offer a full GeoSPARQL endpoint for our prototype yet (which we leave to a forthcoming next release), our RDFized datasets and knowledge graph is GeoSPARQL "ready", i.e. having all the geo-coordinates and polygons in the endpoint using the GeoSPARQL vocabulary; an external GeoSPARQL endpoint could already

```
SELECT ?d ?url WHERE {
  # dates of the past two elections in Austria
  wd:Q1386143 timex:hasStartTime ?t1 .
  wd:Q19311231 timex:hasStartTime ?t2 .

  ?d dcat:distribution [
    dcat:accessURL ?url ;
    # the min and max date values
    timex:hasStartTime ?start ;
    timex:hasEndTime ?end
  ] .
  # filter datasets about Vienna
  ?d csvwx:refersToEntity
      <http://sws.geonames.org/2761369/> .

  FILTER((?start >= ?t1) && (?end <= ?t2))
}
```

Figure 5.18: Example SPARQL query using the spatial property `csvwx:refersToEntity` and the temporal properties `timex:hasStartTime` and `timex:hasEndTime`.

access our data using the SERVICE keyword and evaluate the GeoSPARQL specific functions locally, or simply import our data.

5.3.3 Dataset visualisations

To enable non-technical users to explore the results of our search API, in the showcase context of the nine Austrian federal states, we built an interactive web application [Heil and Neumaier, 2018]. This project is the outcome of a collaboration with the Vienna based Startup 23degrees[43] which is specialised in dataset visualisations. The developed application allows to choose a geographic entity (e.g., a federal state of Austria), uses our search API with the respective geo-entity, and gathers the corresponding datasets. Our application parses for categorical and numeric columns and then scans for geo-references and time components that can be visualised on a map or a barchart. While theoretically the amount of visualisations that can be generated is huge – for one visualiation type the number of possible combinations is the amount of string columns multiplied with the amount of numeric columns – we heuristically reduce this number and generated 6117 visuals for 393 datasets from the Austrian Open Data portal, that can be explored.

Out of 1321 results for the nine Austrian federal states, 928 results could not be visualised either because of download/parsing errors, or because of the structure of the data. The example visualisation in Fig. 5.21 illustrates how users can rate visualisations in our application. We save the rating information and use it to evaluate and improve our visual

[43] `https://23degrees.io`

```
SELECT ?d ?url ?rownum WHERE {
   # get the "Anschluss movement"
   ?p rdfs:label ?L.
   FILTER (CONTAINS(?L, "Anschluss movement") ) .
   ?p timex:hasStartTime ?start ;
      timex:hasEndTime ?end ;
      dcterms:spatial ?sp .
   # find the GeoNames entities
   ?spatial owl:sameAs ?sp .
   ?d dcat:distribution [ dcat:accessURL ?url ] .
   [] csvw:url ?url ;
      csvw:tableSchema ?s .
   # find a cell where date falls in the range of the found period
   ?s csvw:column ?col1 .
   ?col1 csvwx:cell [
      csvw:rownum ?rownum ;
      csvwx:hasTime ?cTime
   ]
   FILTER((?cTime >= ?start) && (?cTime <= ?end))
   # find another cell in the same row where the geo-entity has the
   # spatial coverage area of the found period as the parent country
   ?s csvw:column ?col2 .
   ?col2 csvwx:cell [
      csvw:rownum ?rownum ;
      csvwx:refersToEntity [ gn:parentCountry ?spatial ]
   ]
}
```

Figure 5.19: Example SPARQL query combining text search for a time period with a structured query for datasets within the period's temporal and spatial coverage.

generation process. Currently, in this prototype the users cannot choose specific datasets, and the visualisation is limited to a random selection. We chose this random approach to get some initial ratings, which will hopefully shed some light on dataset features that are useful for meaningful visualisations.

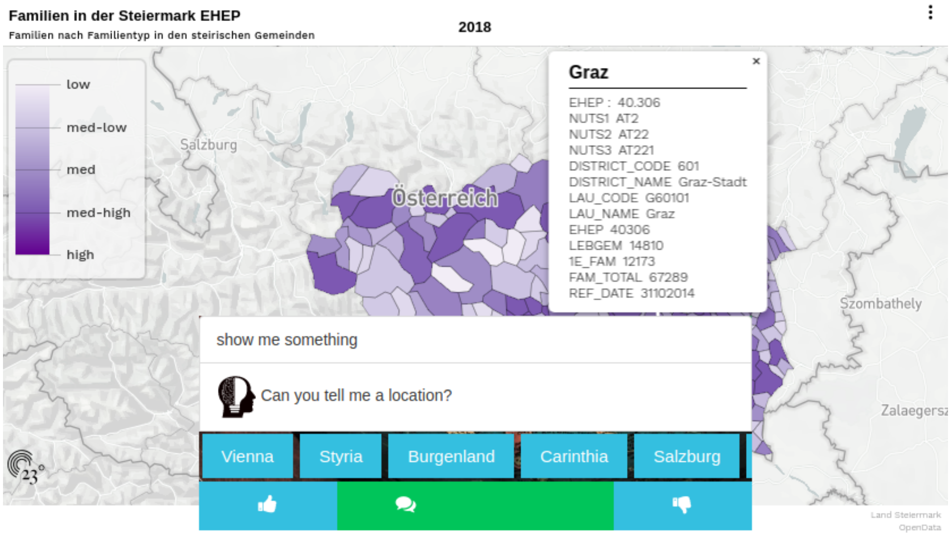
In future versions of this visualisation framework, we plan to integrate information gathered from the user ratings: In case of inadequate representations we will adapt the visualisation (i.e. change the input columns). Also, we plan to scale our systems to datasets/data portals worldwide, so that users can query for any geo-entity/location. Complementary, users might benefit from other dimensions such as temporal and topic filters.

```
SELECT ?d ?url ?rownum WHERE {
  # get the geometry of the Viennese district "Leopoldstadt"
  <http://sws.geonames.org/2772614/>
               geosparql:hasGeometry ?polygon .

  ?d dcat:distribution [ dcat:accessURL ?url ] .
  [ csvw:url ?url ; csvw:tableSchema ?s ].
  # select the geometries of any annotated cells
  ?s csvw:column ?col .
  ?col csvwx:cell [ csvw:rownum ?rownum ;
    csvwx:refersToEntity [geosparql:hasGeometry ?geoentity]]

  # filter all annotated data points
  # within the polygon of Leopoldstadt
  FILTER(geof:sfWithin(?g, ?polygon))
}
```

Figure 5.20: Example GeoSPARQL query over using the available geometries – not yet available via the endpoint.



Graz

EHEP : 40.306
NUTS1 AT2
NUTS2 AT22
NUTS3 AT221
DISTRICT_CODE 601
DISTRICT_NAME Graz-Stadt
LAU_CODE G60101
LAU_NAME Graz
EHEP 40306
LEBGEM 14810
1E_FAM 12173
FAM_TOTAL 67289
REF_DATE 31102014

show me something

Can you tell me a location?

Figure 5.21: Example visualisation by reboting.com: Number of families in Styria, Austria. The color indicates the density; details can be displayed by selecting a subregion.

5.4 Related Work

In this section, we introduce related works in the field of semantic table interpretation and labelling of numerical values (Section 5.4.1) and spatio temporal labelling of datasets (Section 5.4.2).

5.4.1 Semantic Table Interpretation and Labelling of Numerical Values

There exists an extensive body of research in the Semantic Web community to derive semantic labels for attributes in structured data sources (such as columns in tables) which are used to (i) map the schema of the data source to ontologies or existing semantic models or (ii) categorise the content of a data source (e.g. a table talking about politicians, i.e., in our case mapping the rows of a table into classes). The majority of these approaches [Zhang and Chakrabarti, 2013, Ermilov et al., 2013, Rastan, 2013, Taheriyan et al., 2014, Ramnandan et al., 2015, Halevy et al., 2016, Ritze et al., 2015, Oulabi and Bizer, 2019] assume well-formed relational Web tables, rely on textual information, such as table headers and string cell values in the data sources, and apply common, text-based entity linkage techniques for the mapping (see [Zhang, 2017] for a good survey). Moreover, typical approaches for semantic labelling such as [Venetis et al., 2011, Wang et al., 2012, Adelfio and Samet, 2013] recover the semantics of Web tables by considering as additional information the, again textual, "surrounding" (section headers, paragraphs) of the table and leverage a database of class labels and relationships automatically extracted from the Web.

In summary, the main focus of all these works is on textual relations inside the tables and in their surroundings. Techniques for recovering numerical relationships are often left for future work. As for used techniques, while these are out of the scope of our paper, many advanced textual entity recognition and linkage techniques are implemented in the Babelnet system [Navigli and Ponzetto, 2012], as we highlighted in the previous section these techniques are not necessarily applicable to a large portion of (numerical) Open Data. In contrast, our approach assumes that we only have a bag of numerical values available, in the worst case lacking any other rich textual information.

Most closely related to our efforts is the work by Ramnandan et. al. [Ramnandan et al., 2015], where the authors proposed to semantically label tuples of attribute-value pairs (textual and numerical). The semantic labelling of numerical values is achieved by analysing the distribution of the values and compare it to known and labelled distributions given as input by using statistical hypothesis testing. In contrast to their approach, we build a knowledge hierarchy and annotate sources not only with a single label but with a possible type and shared property-object pairs. Also complementary to our efforts is the work of Cruz et.al. [Cruz et al., 2013] which focus on detecting geolocation information in tables and apply heuristics specifically for numerical longitude and latitude values.

Outside the area of semantic labelling as such, but as an inspiration for our approach, the authors of [Fleischhacker et al., 2014, Wienand and Paulheim, 2014] developed approaches to detect natural errors/outliers in RDF knowledge bases and automatically clustered candidate sets from the RDF knowledge base they want to analyse by grouping numerical values of a selected property by their types. We use a similar approach to build our background knowledge: we also group the subjects (and their corresponding values) by their types. However, we use a more fine-grained notion of "type", not only considering named classes but also “subtypes” defined in terms of shared property-object pairs.

5.4.2 Semantic Description and Labelling of Spatio-Temporal Information

Besides the already introduced DCAT metadata standard (cf. Section 2.3), the INSPIRE directive[44] and the GeoDCAT-AP specification[45] provide restrictive requirements for modelling spatial metadata, i.e., they model spatial coverage either as a bounding box, or using a geographic identifier; notably, the specification also mentions GeoNames as potential identifiers. The main barrier with these approaches is a lacking adoption: We could not see a broad use of these standards across the portals (neither in terms of vocabulary nor in complete spatial descriptions) and therefore could not further use them. In principle, our approach distinguishes from these activities by not only having the spatio-temporal descriptions but also interlinking the datasets to external sources, i.e. to GeoNames, Wikidata, and OSM. Also, these standards only allow descriptions on dataset level, whereas we annotate the data on record level as well.

The 2013 study by Janowicz et al. [Janowicz et al., 2013] gives an overview of Semantic Web approaches and technologies in the geospatial domain. Among the Linked Data repositories and ontologies listed in the article we also find the GeoNames ontology (cf. Section 5.2.1), the W3C Geospatial Ontologies [Lieberman et al., 2007], and the GeoSPARQL Schemas [Perry and Herring, 2012]. However, when looking into the paper’s listed repositories, most of them (6/7) were not available, i.e. offline, which seems to indicate that many efforts around Geo-Linked data have unfortunately not been pursued in a sustainable manner.

The 2012 project LinkedGeoData [Stadler et al., 2012] resulted in a Linked Data resource, generated by converting a subset of OpenStreetMap data to RDF and deriving a lightweight ontology from it. In [Hahmann and Burghardt, 2010] the authors describe their attempts to further connect GeoNames and LinkedGeoData, using string similarity measures and geometry matching. LinkedGeoData is also listed in [Janowicz et al., 2013] as a geospatial Linked Data repository. Their work is complementary to ours: they also perform an interlinking with DBpedia, GeoNames, and a mapping from OpenStreetMap, but do not integrate general Open Data resources. That is, their mappings are driven on generic entity linkage between these existing data sources, whereas we create a bespoke new Knowledge Graph out of the existing spatial and temporal linked data sources for our use case. The recent effort “Sophox”[46] can be seen as a conceptual continuation of the LinkedGeoData project: actually intended as a cleanup tool, it allows SPARQL queries over OSM elements and tags. In the future we could also consider directly using the SPARQL interface of Sophox.

[44] https://inspire.ec.europa.eu/

[45] https://joinup.ec.europa.eu/release/geodcat-ap/v101

[46] https://wiki.openstreetmap.org/wiki/Sophox, last accessed 2018-09-03

The GeoKnow project [Lehmann et al., 2015a] is another attempt to provide and manage geospatial data as Linked Data. GeoKnow provides a huge toolset to process these datasets, including the storage, authoring, interlinking, and geospatially-enabled query optimization techniques.

The project PlanetData (2010 to 2014), funded by the European Commission, released an RDF mapping of the NUTS classifications[47] [Harth and Gil, 2014] using the GeoVocab vocabulary.[48] This dataset models the hierarchical relations of the regions, provides labels and polygons. Unfortunately, the project does not include external links to GeoNames, or Wikidata, except for the country level, i.e. there are only 28 links to the corresponding GeoNames entries of the EU member states.

Complementary to our approach to access street addresses via OSM, Open Addresses[49] is a global collection of address data sources, which could be considered for future work as an additional dataset to feed into our Knowledge Graph. The manually collected and homogenized dataset consists of a total of 478M addresses; street names, house numbers, and post codes combined with geographic coordinates, harvested from governmental datasets of the respective countries.

A conceptually related approach, although not focusing on geo-data, is the work by Taheriyan et al. [Taheriyan et al., 2013], who learn the semantic description of a new source given a set of known semantic descriptions as the training set and the domain ontology as the background knowledge.

In [Paulheim, 2017] Paulheim provides a comprehensive survey of refinement methods, i.e., methods that try to infer and add missing data to a graph, however, these approaches work on graphs in a domain independent setting and do not focus on temporal and spatial knowledge. Still, in some sense, we view our methodology of systematic Knowledge Graph aggregation from Linked Data sources via declarative, use-case specific, minimal mappings as potentially complementary to the domain-independent methods mentioned therein, i.e., we think in future works, such methods should be explored in combination.

Most related wrt. the construction of the temporal Knowledge Graph is the work by Gottschalk and Demidova [Gottschalk and Demidova, 2018]: they present a temporal Knowledge Graph that integrates and harmonizes event-centric and temporal information regarding historical and contemporary events. In contrast to [Gottschalk and Demidova, 2018] we additionally integrate data from PeriodO [Golden and Shaw, 2016] and focus on periods in a geospatial context. This work is built upon [Tran and Alrifai, 2014] where the authors extract event information from the Wikipedia Current Events Portal (WCEP). In future work we want to connect the resource from [Gottschalk and Demidova, 2018], since the additional data extracted from the WCEP and WikiTimes interface is in particular interesting for our framework. Similar to [Gottschalk and Demidova, 2018], [Rula et al., 2014] gather temporal information from knowledge bases, and additionally

[47] `http://nuts.geovocab.org/`, last accessed 2018-01-05

[48] `http://geovocab.org/`, last accessed 2018-01-05

[49] `https://openaddresses.io/`, last accessed 2018-04-01

from the Web of documents. The extracted facts get then mapped and merged into time intervals.

In [Rospocher et al., 2016], Rospocher et al. build a Knowledge Graph directly from news articles, and in [Spitz and Gertz, 2016] by extracting event-centric data from Wikipedia articles. These approaches work over plain text (with the potential drawback of noisy data) while we integrate existing structured sources of temporal information; therefore these are complementary/groundwork to our contributions.

Modelling and querying geospatial information has also been discussed conceptually in the literature: [Keßler and Farmer, 2015] present an ontology design pattern derived from time geography, and [Corti et al., 2016] discuss the requirements of a geospatial search platform and present a geospatial registry.

5.5 Critical Discussion and Future Directions

In this chapter we have presented and evaluated our approaches for semantic enrichment and search over datasets with potentially non-human-readable labels, domain-specific content, and predominantly numerical values.

First, in Section 5.1 we have presented our approach to find and rank candidates of semantic labels and context descriptions for a given bag of numerical values. To this end, we have applied a hierarchical clustering over information taken from DBpedia to build a background Knowledge Graph of possible “semantic contexts” for bags of numerical values, over which we perform a nearest neighbour search to rank the most likely candidates. While to the best of our knowledge, this is the first work addressing semantic labelling of numerical values by applying k-nearest neighbours search over a background Knowledge Graph, which is constructed in an unsupervised manner using hierarchical clustering, our work inspired and served as a baseline for a number of research contributions [Nguyen et al., 2018, Alobaid and Corcho, 2018, Kacprzak et al., 2018]. The obtained results were encouraging for labelling numerical columns in tabular data and despite the simplicity of our solution, we can confirm that a knowledge base can be harnessed to perform automatic semantic labelling of datasets with promising results. However, a feasibility evaluation using numerical columns in Open Data CSV files showed that further research is needed to extend our Knowledge Graph to cater for the specifics of the Open Data domain, such as addressing timeliness. Also, we have to investigate performance optimization techniques, since our prediction time increases linearly with the number of context nodes. For example, we will explore range indices or pre-filtering to reduce the search space in the context graph.

Complementary to the numeric labelling algorithm, in Section 5.2 we have presented our approach to construct a Knowledge Graph that hierarchically structures geographical and temporal entities, and we have described algorithms to annotate CSV tables and their respective metadata descriptions using this graph. Governmental data portals release local, regional and national data which is mainly collected as part of census collections, infrastructure assessments or any other, secondary output data. We argue that these datasets almost always contain or refer to some kind of geographic and temporal information – for instance, think of public transport data, results of past elections, demographic indicators, etc. – and therefore have identified the spatial and temporal dimensions as the crucial, characterizing dimensions of datasets on such data portals.

To demonstrate the performance and limitations of our spatio-temporal labelling we have evaluated the annotations by manual inspection of a random sample per data portal, where we identified correct geo-annotations for around 90% of the inspected datasets. To improve the usability we plan to integrate a visual representation and search interface for datasets by displaying and filtering the datasets/records on a map; a first prototypical implementation is discussed in Section 5.3. A further extension of our work could be the parsing and annotation of coordinates in datasets. To do so we need a reverse geocoding service and have to consider that the long/lat pairs potentially come in column groups, i.e., one column per coordinate, which we might need to combine.

To access and query the data, we discuss RDF representations, user interfaces, RESTful APIs and SPARQL endpoints in Section 5.3, which allow structured queries over our spatio-temporal annotations.

While CSV is a popular and dominant data-publishing format on the Web, we also want to extend our labelling and indexing to other popular Open Data formats (such as XLS and JSON). Additionally, we want to test how well our approaches could be applied to unstructured or semi-structured data and other domains (apart from Open Data) such as tweets or web pages (e.g., newspaper articles), or complementarily, we could use our Knowledge Graph, along with methods for numerical, temporal and geo-labelling of such unstructured sources, to link them to (supporting) data, to enable for instance fact checking. The applications of Open Data sources searchable and annotated in such a manner seem promising and widespread.

CHAPTER 6

Conclusion

The following chapter concludes this thesis by summarising (Section 6.1) and critically reviewing (6.2) the contributions and research questions, before eventually giving an outlook of future directions and research challenges in Section 6.3.

6.1 Summary of Contributions

Monitoring Data Quality on the Web. We have started in Chapter 3 with proposing a generic formal model to represent data and metadata in web portals. We have defined a set of 18 concrete quality metrics across five dimensions, using this formal model. To assess and monitor the quality of existing Open Data portals, we have developed a framework that continuously harvests existing data portals and maps the metadata of the major portal software providers to our metadata model (using the W3C's DCAT vocabulary). The framework monitors and assesses the quality of 261 data portals and around 1.1 million datasets on a weekly basis; we have reported the metrics and general findings such as an alarming retrievability of the resources and a high prevalence of non-machine-readable file formats. We have ended the chapter with a study of a corpus of CSVs from the monitored portals in order to gain insights into common properties and characteristics of the Open Data tables.

Lifting Data Portals to the Web of Data. In Chapter 4, we have described our approach for homogenising and re-exposing the metadata collected from the monitored portals. For publishing metadata descriptions we use the DCAT and the Schema.org dataset vocabulary, which we enrich with the quality measurements using the W3C's Data Quality Vocabulary, and the descriptions of tabular datasets using the CSV on the Web vocabulary. Eventually, we have discussed and implemented different data access methods: We provide the dataset descriptions as Linked Data as dumps, via APIs with versioned/historic access using the Memento framework, and via a SPARQL endpoint.

Semantic Enrichment and Search Finally, in Chapter 5 we have presented and evaluated our approaches for semantic enrichment and labelling of datasets, having the use case of dataset search in mind. First, we have constructed a Knowledge Graph by taking numeric values from DBpedia and organising them by common classes and properties. We use this constructed graph to find and rank candidates of semantic labels and context descriptions for numeric columns: we perform a hierarchical clustering over the nodes of the graph to rank the most likely labels for the numerical values in columns.

Complementary to the numeric labelling algorithm, we have presented in Section 5.2 a second approach to construct a Knowledge Graph: We have curated and combined other, existing Knowledge Graphs in a purpose-driven manner. In our scenario, a large number of points of interest, street, districts, and other geo-spatial entities are available and described already in existing Knowledge Graphs such as Geonames, or DBpedia. However, (i) alignment between the existing KGs is necessary, and (ii) the integration of other structured sources or KGs (such as OpenStreetMaps, or other national geographical databases) is needed. We have detailed such a construction of a hierarchical KG of geo-entities and temporal entities and links between them, and have presented a scalable labelling algorithm for linking open datasets (both on a dataset-level and on a record-level) to this Knowledge Graph; an evaluation of the annotated datasets to illustrate the feasibility and effectiveness of the approach is provided. Further, we have implemented a search interface where the indexed and annotated datasets can be queried.

6.2 Critical Assessment of the Research Questions

After summarising our contributions, we want to critically review the contributions with respect to our motivating research questions from Section 1.1.

Q1 *(i) How can we use Semantic Web technologies for the semantic enrichment and integration of Open Data, and what are the limitations and upcoming challenges? (ii) How is the current state of published Open Data in terms of quality?*

We answer (i) in Chapter 2 by surveying the foundations and basic challenges: Current Open Data is mainly published at so called Open Data portals, in different, heterogeneous formats and schemata; RDF or Linked Data are less readily available. We review W3C vocabularies for rich descriptions of datasets, their provenance and licensing. Eventually, we discuss the increasingly important role of Knowledge Graphs for publishing and linking data.

In Chapter 3 we focus on answering (ii), the quality assessment of current Open Data. Initially, we formally define the concepts of *web data portals*, *metadata descriptions*, and propose a set of concrete quality dimensions and metrics based on this model. To actually assess these metrics, we have developed a framework for continuous and extensible assessment, monitoring and archiving of metadata descriptions – the Open Data Portal Watch – however, a **continuous profiling**

and archiving of the actual data is still missing. We discuss this in detail in Section 6.3.1.

Q2 *How to describe and publish datasets, and (potentially enriched/improved) metadata, as Linked Data in an homogenised way?*

We detail our approach for question two in Chapter 4: The Open Data portals' metadata descriptions come in different schemata, and potentially low quality. To solve this issue, we describe the process of re-exposing the collected data, we enrich the integrated metadata by the quality measurements, provenance, and descriptions of tabular data. As proof of concept and service for the community, we currently host the processed dataset descriptions as dumps, query endpoints, and APIs. Ideally, however, the users should **find the Linked Data endpoints and rich descriptions already at the data portals**, i.e. data/portal providers should be incentified to output Linked Data standards upfront.

Q3 *How to extract spatial, and temporal information from the data sources and, orthogonally, how to assign semantic entities (e.g. spatial or temporal contexts such as locations, events, etc.) to (combinations of) values in rows of such tables?*

Our approach to annotate CSV tables and metadata at scale, with links to spatial and temporal entities, and a search and query interface, can be found in Chapter 5. We showcase this work by annotating a large corpus of tabular datasets from Open Data portals with entities from a constructed hierarchical Knowledge Graph of geo- and temporal-entities and links between them. It is still open, however, if our approach is **generalisable for other formats** and **generalisable for other entities**, apart from spatio-temporal references alone, such as categories, (governmental) organisations, etc.

Q4 *How to assign semantic labels to columns lacking textual information?*

To answer research question four, we propose an approach to find and rank candidates of semantic labels and context descriptions for a given bag of numerical values in Section 5.1. To this end, we apply a hierarchical clustering over information taken from DBpedia to build a background Knowledge Graph of possible "semantic contexts" for bags of numerical values, over which we perform a nearest neighbour search to rank the most likely candidates. While we could successfully assign fine-grained semantic labels, if there is enough evidence in the background knowledge, i.e., if the data is covered by respective DBpedia properties, in general, there is **missing domain knowledge for labelling Open Data**. We further discuss this challenge in Section 6.3.2.

6.3 Open Challenges and Future Research Directions

There are several important and – to us – interesting future directions and open research challenges arising from the work presented in this thesis.

6.3.1 Structural Heterogeneity

Whereas most data integration approaches focus on syntactically homogeneous data, in this thesis we have encountered various data formats and shapes, including *tabular data*, which again can be subdivided in different types (e.g., relational vs matrix vs multi-tables, etc.) [Eberius et al., 2015a, Mitlöhner et al., 2016], tree-shaped (JSON) and graph-like (e.g. RDF or property graphs) Knowledge Graphs. Both, classifying unknown data sources into one of these is non-trivial as well as a common uniform abstract model that combines these and represents their schema in a common manner is missing. While theoretical works transferring relational schema dependencies and normal forms theory to RDF [Calvanese et al., 2014] or XML [Arenas and Libkin, 2005] exist, their performance at scale and, most importantly, impact on classification and understanding of data, have not been sufficiently investigated.

Works on classifying (tabular) datasets by structure, such as [Lehmberg et al., 2016, Eberius et al., 2015b] on classifying Web tables (using traditional classification algorithms [Eberius et al., 2015a], or neural networks [Nishida et al., 2017, Ghasemi-Gol and Szekely, 2018]), have not been applied to (mostly numeric) structured data tables as we investigate them; we expect challenges particularly due to the insufficient natural language cues which prior approaches largely relied upon.

Methodologically, in terms of how to tackle the challenge of analysing and resolving structural heterogeneity, we see the following necessary next steps:

Table Structure: Initially, we have to analyse and categorise the structure of tabular data in a large-scale data corpus, in order to better understand the content.[1] Information about the structure allows us, in the next steps, to select the methods that we want to apply.

Table Classification: Using the structural information, we can classify different table models [Ghasemi-Gol and Szekely, 2018, Eberius et al., 2015a, Ermilov and Ngomo, 2016], such as relational vs. matrix tables, different orientations of a table, and detecting sub-header rows/multi-tables in a dataset. While there is already existing work on subject column identification [Ermilov and Ngomo, 2016], having multiple subject columns relating to the properties, and having potential inter-column-relationships between the classes makes this a complex research challenge.

Leverage Context Information: We want to further exploit the datasets' metadata information and context in order to identify additional key elements. In particular, many existing works on semantic table interpretation and annotation have ignored/neglected this additional cues [Das Sarma et al., 2012, Oulabi and Bizer,

[1] In Section 3.6 we already performed an initial profiling of a corpus of CSV files; we analysed the distributions of columns, rows, simple data types, etc. Using this as a basis, we can go into in-depth profiling, including data characteristics such as character distributions, pattern representations and functional dependencies [Lehmberg and Bizer, 2019]. A good survey on profiling relational data can be found in [Abedjan et al., 2015].

2019, Zhang, 2017]. In contrast, Lehmberg and Bizer [Lehmberg and Bizer, 2019] use the context and surroundings of Web tables to discover functional dependencies. Apart from metadata and explicit context information, there is also *implicit context information* such as the temporal context of datasets – in terms of both validity time (which temporal context do the observations in the dataset refer to [Neumaier and Polleres, 2019]) and transaction time (when has the data been updated/changed).

6.3.2 Availability of Semantic Background Knowledge

As for instance our failed attempts on semantic labelling of numeric data show, Knowledge Graphs and ontologies, require extension, both in terms of (i) domain coverage and (ii) expressivity of mappings and schema descriptions in common ontology languages. On the other hand, (iii) statistical models for natural language that made machine-translation and linkage possible in many cases are not directly applicable to structured/numeric data: that is, as for (i), existing public KGs like Wikidata [Vrandecic and Krötzsch, 2014] and DBpedia [Lehmann et al., 2015b] do not cover the domain knowledge in Open Data or enterprise data, where resp. domain-specific KGs still need to be built. Secondly, as for (ii), while ontologies, e.g., about unit conversion [Rijgersberg et al., 2013] or generic formats to exchange mappings involving complex value conversions [David et al., 2011] exist, such common knowledge about value transformation is not formalised in an operational form to be directly applicable to existing KGs. As for (iii), recent approaches wrt. leveraging statistical background knowledge, in the form of pre-trained vector-space embeddings trained from large text corpora [Mikolov et al., 2013] have been successfully applied in various language translation approaches and question answering from text. Moreover, such embeddings have been used to enrich links in existing KGs [Nickel et al., 2016, Cochez et al., 2017]. However, embeddings specifically trained for interlinking structured, largely numeric and non-textual data sources are missing.

Declarative and Statistical Background Knowledge: In principle, it is important to build up knowledge about how dependent attributes can be computed, for what Das Sarmas et al. [Das Sarma et al., 2012] call *Schema Complement*: a table "contains the same set of entities (due to identical selection conditions) as [another table] does, for a different and yet semantically related set of attributes" [Das Sarma et al., 2012]. If we have the determining variables (i.e. the keys of the computable functional dependencies) from another source, we can compute the missing depending one. While Bischof et al. [Bischof et al., 2018] started to investigate this problem on small scale, and with a manually curated list of equational axioms, we envision to extend/generalise and scale this approach.

Linking to Knowledge Graphs: The current state-of-the-art methods, coming from schema re-engineering of relational data, mainly use Web tables as an evaluation corpus [Das Sarma et al., 2012, Oulabi and Bizer, 2019, Zhang and Chakrabarti, 2013, Zhang, 2017]. However, the relational data as found on Open Data portals - are fundamentally different to such Web table corpora (e.g., from Wikipedia)

[Neumaier et al., 2016a, Ritze et al., 2016]: Our studies showed for instance that an average CSV on such a portal contains 365 rows and 14 columns, and that 50% of the CSV columns contain either numerical values or non-textual identifiers [Neumaier et al., 2016a]. Moreover, different formats such as JSON gain popularity. In order to leverage the non-textual/numerical cues in our data corpus, we want to further develop our existing approaches, such as exploiting spatio-temporal cues and distributions of numerical data. Our hypothesis here is that we can significantly improve these earlier approaches, by novel combinations, particularly linkage to existing KGs, and leveraging the sparse natural language information within the data and metadata.

Large Semi-Structured and Numeric Data Corpora: While the public availability of large, well-studied text corpora has boosted research in NLP, embeddings etc., the same cannot be said for semi-structured data. While, with trends towards Open Data, this situation could change considerably with a lot more raw data sources being made available, another feature that has boosted NLP research is yet to be investigated for these structured data sources: the redundancy in textual data and availability of *overlapping or multi-lingual* text sources, talking about similar concepts and things is the real decisive factor that has enabled machine-learning at scale on these text corpora. For semi-structured data corpora, these degrees of overlap and in turn identifying promising sub-corpora with large overlaps, yet need to be analysed in detail.

6.3.3 Automated Categorisation of Datasets

As we showed in our quality assessments and analyses, the use of taxonomies is difficult to capture and very heterogeneous across the monitored data portals. In fact, we could only map the dcat:theme attribute (which is used to categorise resources) for 28% of all datasets (cf. Section 3.4.1), and saw over 3300 different category values for this property, which are used very portal-specific and individual.

In an earlier approach, in the course of a co-supervised master thesis [Prohaska, 2017], we tried to automatically align the available category values and keywords of the datasets: first, our approach uses the disambiguation and entity linkage services of the BabelFy system [Moro et al., 2014] to assign related concepts and entities to a dataset. BabelFy/BabelNet provides references to DBpedia entries, and based on these we tried to extracted the relevant DBpedia category for a dataset. While some results were reasonable for certain categories (e.g. “Politics and Government”), in general the results were not convincing enough, and the approach was not scalable – due to the restricted use of the BabelFy service – so we decided to not further pursue this work for the present thesis.

An alternative approach, that we want to note for future work, is to train a classifier using the available categorisations: as training data we can use the portals where we have an existing taxonomy (e.g. using the dcat:theme attributes) together with the titles, tags and descriptions. The learned multi-class classifier then could be used to predict the categories of the uncategorised datasets, or to align their taxonomies. Similar approaches already exists for the LOD dataset [Spahiu et al., 2019] and web tables [Primpeli et al., 2019].[2]

6.3.4 Dataset Search, Relatedness and Ranking

Currently, our dataset search approaches are based on entity recognition and linkage (cf. Section 5.3). Complementary to this approach, we want to investigate how to use the relatedness between tables and their structural components (rows, columns, etc.) for enabling dataset search and ranking. As a starting point for such relatedness measures, there is work on vector-space embeddings, successfully applied to language modeling and relating texts and documents.

While there has been initial work on leveraging and adapting the ideas behind such embeddings (and their application to learning tasks using neural networks) to KGs [Cochez et al., 2017], there is – to the best of our knowledge – not much work on using similar applications in the context of Web/Open Data tables: Existing work is based on incorporating pre-trained textual embedding models [Gentile et al., 2017, Zhang and Balog, 2018, Ghasemi-Gol and Szekely, 2018], or – in very recent work – training such embeddings from mainly the textual information in the Web tables themselves [Deng et al., 2019].

For instance, the work proposed by Gentile et al. [Gentile et al., 2017] focuses on a specific task, which we plan to use as a starting point: it uses table embeddings – vector representations of tables from the Web, following the idea of word embeddings[3] – for partitioning data, in order to reduce the complexity of entity matching, i.e. to reduce the number of performed pair comparisons. While we plan to build upon this so-called blocking approach, which already showed promising results for Web tables, we will need to generalise it to another corpus (with less natural language cues) and to other tasks (beside blocking). To this end, we want to further research how more general vector-space representations for tables and their components can be implemented and utilised:

- As opposed to [Gentile et al., 2017] which ignores numbers, dates, etc. (replaced with a static value) we want to research how to better incorporating tabular information; in particular, we will have to look into suitable vector representations of the non-textual features (e.g., structural information, data-type specific value ranges and distributions, alphanumeric patterns, and context information).
- We want to exploit the use of these word embedding techniques beyond partitioning and blocking to restrict the search space: Having a reasonable model we can test

[2] We would like to thank Christian Bizer for suggesting this approach in his review of this thesis.

[3] In more detail, Gentile et al. [Gentile et al., 2017] use a pre-processed and simplified string representation leveraging the natural language information in Web tables to train a Word2vec [Mikolov et al., 2013] model.

these representations for finding related tables and for ranking in a dataset search scenario.

6.3.5 Assessing the Impact and Usefulness of our Approaches

With our work on quality reports, on mappings to vocabularies (including Schema.org), and on metadata enrichments (e.g., by using the CSV dialect) we publish and produce data and reports, which partially already found their way into existing ecosystems. For instance, the Austrian data.gv.at integrates the quality assessment and (meta)data improvements in their portal, and the Google Crawler indexes the published Schema.org mappings of the dataset descriptions (cf. Section 4.6).

A natural next step for future work would be to evaluate now the usefulness of our approaches and ideally show an increased value. An empirical assessment of the impact of our advanced *geo-spatial search features*, however, would require to integrate our search functionality into an existing portal/system in order to quantitatively show significant improvements.

The same applies to the *Portal Watch front-end and visualisations* presented in this thesis: in their current stage they are designed, and probably most usable, for comparable analyses of portals, which we deem useful for both, the research community and portal providers. However, a subsequent study of the usefulness of these visualisations for this particular stakeholder group – starting with the collection of specific requirements from portal providers in interviews – would be a very promising future direction.

Concluding Remarks

The current main sources of structured data on the Web face a trade-off between accuracy and coverage: On the one hand, open Knowledge Graphs, such as Wikidata, are typically sparse graphs that only cover popular entities and events,[4] however, with a very high accuracy, highly structured, and aligned to a schema. Open Data, available on (governmental) portals, on the other hand, has the potential to cover all kind of domains and information, beyond and complementary to existing Knowledge Graphs. The catch: due to the heterogeneous nature and the lack of structure we do not reach high accuracy, and therefore we continue developing tools for cleaning, extraction, and semantic enrichment.

Semantic Web research already has spent many years of committed research on the integration and enrichment of datasets published on the Web. However, it also attracted criticism and doubts, and did not really develop as expected over the years: often-heard critics include that the impact remains mainly academic, and that there is a lack of practical applications. The (scientific and non-scientific) output is perceived as

[4] While Wikidata certainly covers a range of domains and topics, it lacks coverage of specific and fine-grained information, e.g., live data such as weather or public transport information, or fine-grained demographic datasets such as immigration data, distribution of income, etc.

complicated, theoretical, not applied enough; producing too many standards which are not mature/lacking adoption [Swartz, 2013]. In fact, it can be observed that the academic community is still the main driver [Schmachtenberg et al., 2014, Polleres et al., 2018], and that the dominant parties on the Web do not necessarily use/need Semantic Web technologies for their semantic systems.

In contrast to the Semantic Web, the Open Data movement experienced some direct "uptake": numerous data portals popped up, publishing patterns and ecosystems have been established, and large data amounts became available online, supported by many governments and other public bodies. However, as our work shows, Open Data still has to fight some complementary challenges. The myths that "It is a matter of simply publishing public data", and that "the publicizing of data will automatically yield benefits" [Janssen et al., 2012] – that we, admittedly, also make use of in the introduction of this thesis – are not true.

What we have been aiming to show in this thesis is that the combination of these two complementary initiatives, Semantic Web standards and technologies as well as Open Data, is just precisely what is needed to make another jump forward in pursuit of more open, transparent and easier accessible data shared on the Web. To really make the most of Open Data, it needs structured, high quality, resources that have the structure and accuracy of Knowledge Graphs, while the Semantic Web community might need the application case to show what it is capable of, while understanding which data can be integrated, and to what degree [Bernstein et al., 2016]. To this end, we see our work as guidelines, tools, and as a showcase of the potential of enriched and integrated Open Data. As voices are raising already, demanding proof of return of investments into Open Data, we hope that a better quality monitoring and easier access to open data, as enabled by the approaches shown in this theses, might be a puzzle piece enabling longer-term success of Open Data.

APPENDIX A

Appendices

A.1 Prefixes

In Figure A.1 we list the prefixes used throughout this thesis to abbreviate URIs.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX csvw: <http://www.w3.org/ns/csvw#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX periodo: <http://n2t.net/ark:/99152/p0v#>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX gn: <http://www.geonames.org/ontology#>
PREFIX osm: <https://www.openstreetmap.org/>
PREFIX timex: <http://data.wu.ac.at/ns/timex#>
PREFIX csvwx: <http://data.wu.ac.at/ns/csvwx#>
```

Figure A.1: Namespaces used throughout this thesis.

A.2 List of Open Data Portals

The following Tables A.1 to A.4 list all data portals monitored by the Open Data Portal Watch framework.

1	LWBIN_Data_HUB	http://130.179.67.140/	CA	CKAN
2	africaopendata_org	http://africaopendata.org/		CKAN
3	annuario_comune_fi_it	http://annuario.comune.fi.it/	IT	CKAN
4	*shut down*			
5	berkeley_demo_socrata_com	https://berkeley.demo.socrata.com	US	Socrata
6	bermuda_io	http://bermuda.io/	IO	CKAN
7	beta_avoindata_fi	http://beta.avoindata.fi/	FI	CKAN
8	bistrotdepays_opendatasoft_com	*shut down*	FR	OpenDataSoft
9	bmgf_demo_socrata_com	http://bmgf.demo.socrata.com	US	Socrata
10	*shut down*			
11	bronx_lehman_cuny_edu	https://bronx.lehman.cuny.edu	US	Socrata
12	bythenumbers_sco_ca_gov	http://bythenumbers.sco.ca.gov	US	Socrata
13	catalogodatos_gub_uy	https://catalogodatos.gub.uy/	UY	CKAN
14	catalogue_datalocale_fr	http://catalogue.datalocale.fr/	FR	CKAN
15	cdph_data_ca_gov	http://cdph.data.ca.gov	US	Socrata
16	ckan_gsi_go_jp	http://ckan.gsi.go.jp/	JP	CKAN
17	ckan_odp_jig_jp	http://ckan.odp.jig.jp/	JP	CKAN
18	ckan_okfn_gr	http://ckan.okfn.gr/	GR	CKAN
19	ckanau_org	http://ckanau.org/	AU	CKAN
20	controllerdata_lacity_org	https://controllerdata.lacity.org	US	Socrata
21	dados_al_gov_br	http://dados.al.gov.br/	BR	CKAN
22	dados_gov_br	http://dados.gov.br	BR	CKAN
23	dados_recife_pe_gov_br	http://dados.recife.pe.gov.br/	BR	CKAN
24	dados_rs_gov_br	https://dados.rs.gov.br/	BR	CKAN
25	dadosabertos_senado_gov_br	http://dadosabertos.senado.gov.br/	BR	CKAN
26	danepubliczne_gov_pl	https://danepubliczne.gov.pl/	PL	CKAN
27	dartportal_leeds_ac_uk	http://dartportal.leeds.ac.uk/	GB	CKAN
28	data_acgov_org	https://data.acgov.org	US	Socrata
29	data_act_gov_au	https://data.act.gov.au	AU	Socrata
30	data_albanyny_gov	https://data.albanyny.gov	US	Socrata
31	data_atf_gov	http://data.atf.gov	US	Socrata
32	data_austintexas_gov	https://data.austintexas.gov	US	Socrata
33	data_baltimorecity_gov	https://data.baltimorecity.gov	US	Socrata
34	data_bris_ac_uk_data_	https://data.bris.ac.uk/data/	GB	CKAN
35	data_buenosaires_gob_ar	http://data.buenosaires.gob.ar/	AR	unknown
36	data_burlingtonvt_gov	https://data.burlingtonvt.gov	US	Socrata
37	data_cdc_gov	https://data.cdc.gov	US	Socrata
38	data_cityofboston_gov	https://data.cityofboston.gov	US	Socrata
39	data_cityofchicago_org	http://data.cityofchicago.org	US	Socrata
40	data_cityofdeleon_org	http://data.cityofdeleon.org	US	Socrata
41	data_cityofmadison_com	https://data.cityofmadison.com	US	Socrata
42	data_cityofnewyork_us	https://data.cityofnewyork.us	US	Socrata
43	data_cityofsantacruz_com	http://data.cityofsantacruz.com/	US	CKAN
44	data_cityoftacoma_org	http://data.cityoftacoma.org	US	Socrata
45	data_cms_hhs_gov	http://data.cms.hhs.gov	US	Socrata
46	data_colorado_gov	https://data.colorado.gov	US	Socrata
47	data_ct_gov	https://data.ct.gov	US	Socrata
48	data_culvercity_org	https://data.culvercity.org	US	Socrata
49	data_datamontana_us	https://data.datamontana.us	US	Socrata
50	data_dcpcsb_org	http://data.dcpcsb.org	US	Socrata
51	data_edmonton_ca	http://data.edmonton.ca	CA	Socrata
52	data_edostate_gov_ng	http://data.edostate.gov.ng/	NG	DKAN
53	data_eindhoven_nl	http://data.eindhoven.nl	NL	OpenDataSoft
54	data_energystar_gov	https://data.energystar.gov	US	Socrata
55	data_glasgow_gov_uk	http://data.glasgow.gov.uk/	GB	CKAN
56	data_go_id	http://data.go.id/	ID	CKAN
57	data_gov	http://data.gov/	US	CKAN
58	data_gov_au	http://data.gov.au/	AU	CKAN
59	data_gov_bf	http://data.gov.bf/	BF	CKAN
60	data_gov_gr	http://data.gov.gr/	GR	CKAN

Table A.1: List of data portals including index, internal portal ID, URL, country and software framework.

61	data_gov_hk_en_	https://data.gov.hk/en/	HK	CKAN
62	data_gov_hr	http://data.gov.hr/	HR	CKAN
63	data_gov_ie	http://data.gov.ie/	IE	CKAN
64	data_gov_md	http://data.gov.md/	MD	CKAN
65	data_gov_ro	http://data.gov.ro/	RO	CKAN
66	data_gov_sk	http://data.gov.sk/	SK	CKAN
67	data_gov_uk	http://data.gov.uk/	GB	CKAN
68	data_graz_gv_at	http://data.graz.gv.at/	AT	CKAN
69	data_grcity_us	http://data.grcity.us/	US	CKAN
70	data_gv_at	http://data.gv.at	AT	CKAN
71	data_hartford_gov	http://data.hartford.gov	US	Socrata
72	data_hawaii_gov	http://data.hawaii.gov	US	Socrata
73	data_hdx_rwlabs_org	https://data.hdx.rwlabs.org/	LU	CKAN
74	data_honolulu_gov	https://data.honolulu.gov	US	Socrata
75	data_iledefrance_fr	http://data.iledefrance.fr	FR	OpenDataSoft
76	data_illinois_gov	http://data.illinois.gov	US	Socrata
77	data_illinois_gov_belleville	https://data.illinois.gov/belleville	US	Socrata
78	data_illinois_gov_champaign	https://data.illinois.gov/champaign	US	Socrata
79	data_illinois_gov_rockford	https://data.illinois.gov/rockford	US	Socrata
80	data_kcmo_org	https://data.kcmo.org	US	Socrata
81	data_kingcounty_gov	https://data.kingcounty.gov	US	Socrata
82	data_kk_dk	http://data.kk.dk/	DK	CKAN
83	data_ktn_gv_at	http://data.ktn.gv.at/	AT	CKAN
84	data_lexingtonky_gov	http://data.lexingtonky.gov/	US	CKAN
85	data_linz_gv_at	http://data.linz.gv.at/	AT	CKAN
86	data_london_gov_uk	http://data.london.gov.uk/	GB	CKAN
87	data_maryland_gov	https://data.maryland.gov	US	Socrata
88	data_medicare_gov	http://data.medicare.gov	US	Socrata
89	data_michigan_gov	http://data.michigan.gov	US	Socrata
90	data_mo_gov	https://data.mo.gov	US	Socrata
91	data_montgomerycountymd_gov	https://data.montgomerycountymd.gov	US	Socrata
92	data_murphytx_org	http://data.murphytx.org	US	Socrata
93	data_nfpa_org	https://data.nfpa.org	US	Socrata
94	data_nhm_ac_uk	http://data.nhm.ac.uk/	GB	CKAN
95	data_nj_gov	http://data.nj.gov	US	Socrata
96	data_noaa_gov_dataset	https://data.noaa.gov/dataset	US	CKAN
97	data_nola_gov	http://data.nola.gov	US	Socrata
98	data_nsw_gov_au	http://data.nsw.gov.au/	AU	CKAN
99	data_ny_gov	https://data.ny.gov	US	Socrata
100	data_oaklandnet_com	https://data.oaklandnet.com	US	Socrata
101	data_ohouston_org	http://data.ohouston.org/	US	CKAN
102	data_ok_gov	http://data.ok.gov	US	CKAN
103	data_opencolorado_org	http://data.opencolorado.org/	US	CKAN
104	data_openpolice_ru	http://data.openpolice.ru/	RU	CKAN
105	data_openva_com	http://data.openva.com/	US	CKAN
106	data_oregon_gov	http://data.oregon.gov	US	Socrata
107	data_ottawa_ca	http://data.ottawa.ca/	CA	CKAN
108	data_overheid_nl	https://data.overheid.nl/	NL	CKAN
109	data_providenceri_gov	https://data.providenceri.gov	US	Socrata
110	data_qld_gov_au	http://data.qld.gov.au/	AU	CKAN
111	data_raleighnc_gov	https://data.raleighnc.gov	US	Socrata
112	data_redmond_gov	https://data.redmond.gov	US	Socrata
113	data_rio_rj_gov_br	http://data.rio.rj.gov.br/	BR	CKAN
114	data_sa_gov_au	https://data.sa.gov.au/	AU	CKAN
115	data_salzburgerland_com	http://data.salzburgerland.com	AT	CKAN
116	data_seattle_gov	http://data.seattle.gov	US	Socrata
117	data_sfgov_org	https://data.sfgov.org	US	Socrata
118	data_somervillema_gov	http://data.somervillema.gov	US	Socrata
119	data_southbendin_gov	https://data.southbendin.gov	US	Socrata
120	data_stadt-zuerich_ch	https://data.stadt-zuerich.ch/	CH	CKAN
121	data_surrey_ca	http://data.surrey.ca/	CA	CKAN
122	data_tainan_gov_tw	http://data.tainan.gov.tw/	TW	CKAN
123	data_taxpayer_net	https://data.taxpayer.net	US	Socrata
124	data_ug	http://data.ug/	UG	CKAN
125	data_undp_org	http://data.undp.org	US	Socrata

Table A.2: List of data portals (cont'd).

126	data_upf_edu_en_main	http://data.upf.edu/en/main	ES	CKAN
127	data_vermont_gov	https://data.vermont.gov	US	Socrata
128	data_wa_gov	http://data.wa.gov	US	Socrata
129	data_weatherfordtx_gov	https://data.weatherfordtx.gov	US	Socrata
130	data_wellingtonfl_gov	http://data.wellingtonfl.gov	US	Socrata
131	data_winnipeg_ca	http://data.winnipeg.ca	CA	Socrata
132	data_wokingham_gov_uk	http://data.wokingham.gov.uk	GB	Socrata
133	data_wu_ac_at	http://data.wu.ac.at/	AT	CKAN
134	data_zagreb_hr	http://data.zagreb.hr/	HR	CKAN
135	datacatalog_cookcountyil_gov	http://datacatalog.cookcountyil.gov	US	Socrata
136	dataforjapan_org	http://dataforjapan.org	JP	CKAN
137	datagm_org_uk	http://datagm.org.uk	US	CKAN
138	datahub_io	http://datahub.io/	US	CKAN
139	datameti_go_jp_data_	http://datameti.go.jp/data/	JP	CKAN
140	datamx_io	http://datamx.io/	IO	CKAN
141	datapilot_american_edu	https://datapilot.american.edu	US	Socrata
142	dataratp_opendatasoft_com	http://dataratp.opendatasoft.com	FR	OpenDataSoft
143	daten_rlp_de	http://daten.rlp.de/	DE	CKAN
144	dati_lazio_it	https://dati.lazio.it/	IT	CKAN
145	dati_lombardia_it	https://dati.lombardia.it	IT	Socrata
146	dati_toscana_it	http://dati.toscana.it/	IT	CKAN
147	dati_trentino_it	http://dati.trentino.it/	IT	CKAN
148	dati_veneto_it	http://dati.veneto.it/	IT	CKAN
149	datos_alcobendas_org	https://datos.alcobendas.org/	ES	CKAN
150	datos_argentina_gob_ar	http://datos.argentina.gob.ar/	AR	CKAN
151	datos_codeandomexico_org	http://datos.codeandomexico.org/	US	CKAN
152	datos_gob_mx	http://datos.gob.mx/	MX	CKAN
153	datosabiertos_ec	http://catalogo.datosabiertos.gob.ec/	EC	CKAN
154	datosabiertos_malaga_eu	http://datosabiertos.malaga.eu/	ES	CKAN
155	datospublicos_org	http://datospublicos.org/	US	CKAN
156	donnees_ville_montreal_qc_ca	http://donnees.ville.montreal.qc.ca/	CA	CKAN
157	donnees_ville_sherbrooke_qc_ca	http://donnees.ville.sherbrooke.qc.ca/	CA	CKAN
158	dot_demo_socrata_com	http://dot.demo.socrata.com	US	Socrata
159	drdsi_jrc_ec_europa_eu	http://drdsi.jrc.ec.europa.eu/	EU	CKAN
160	edx_netl_doe_gov	https://edx.netl.doe.gov/	US	CKAN
161	exploredata_gov_ro	https://exploredata.gov.ro	RO	Socrata
162	finances_worldbank_org	https://finances.worldbank.org	US	Socrata
163	gavaobert_gavaciutat_cat	https://gavaobert.gavaciutat.cat	ES	Socrata
164	geothermaldata_org	http://geothermaldata.org/	US	CKAN
165	gisdata_mn_gov	http://gisdata.mn.gov	US	CKAN
166	govdata_de	http://govdata.de	DE	CKAN
167	hampton_demo_socrata_com	https://hampton.demo.socrata.com	US	Socrata
168	health_data_ny_gov	https://health.data.ny.gov	US	Socrata
169	healthdata_nj_gov	https://healthdata.nj.gov/	US	Socrata
170	healthmeasures_aspe_hhs_gov	http://healthmeasures.aspe.hhs.gov	US	Socrata
171	hubofdata_ru	http://hubofdata.ru/	RU	CKAN
172	iatiregistry_org	http://iatiregistry.org	US	CKAN
173	*shut down*			
174	irs_demo_socrata_com	http://irs.demo.socrata.com	US	Socrata
175	leedsdatamill_org	http://leedsdatamill.org/	GB	CKAN
176	linkeddatacatalog_dws_ informatik_uni-mannheim_de	http://linkeddatacatalog.dws./ informatik.uni-mannheim.de/	DE	CKAN
177	nats_demo_socrata_com_login	https://nats.demo.socrata.com/login	US	Socrata
178	nycopendata_socrata_com	http://nycopendata.socrata.com	US	Socrata
179	offenedaten_de	http://offenedaten.de/	DE	CKAN
180	open-data_europa_eu	http://open-data.europa.eu/	EU	CKAN
181	open_nrw	https://open.nrw/	DE	CKAN
182	open_whitehouse_gov	https://open.whitehouse.gov	US	Socrata
183	opencolorado_org	http://opencolorado.org/	US	CKAN
184	opendata_aberdeencity_gov_uk	https://data.aberdeencity.gov.uk/	GB	CKAN
185	opendata_admin_ch	http://opendata.admin.ch/	CH	CKAN
186	opendata_aragon_es	http://opendata.aragon.es/	ES	CKAN
187	opendata_awt_be	http://opendata.awt.be/	BE	CKAN
188	opendata_ayto-caceres_es	http://opendata.ayto-caceres.es/	ES	CKAN
189	opendata_bayern_de	https://opendata.bayern.de/	DE	CKAN
190	opendata_brussels_be	https://opendata.brussels.be/	BE	OpenDataSoft

Table A.3: List of data portals (cont'd).

191	opendata_caceres_es	http://opendata.caceres.es/	ES	CKAN
192	opendata_cnmc_es	http://opendata.cnmc.es/	ES	CKAN
193	opendata_comune_bari_it	http://opendata.comune.bari.it/	IT	CKAN
194	opendata_go_ke	http://www.opendata.go.ke/	KE	Socrata
195	opendata_go_tz	http://opendata.go.tz/	TZ	CKAN
196	opendata_government_bg	https://opendata.government.bg/	BG	CKAN
197	opendata_hu	http://opendata.hu/	HU	CKAN
198	opendata_lasvegasnevada_gov	https://opendata.lasvegasnevada.gov	US	Socrata
199	opendata_lisra_jp	http://opendata.lisra.jp	JP	CKAN
200	opendata_opennorth_se	http://opendata.opennorth.se/	SE	CKAN
201	opendata_paris_fr_opendatasoft_com	https://opendata.paris.fr/	FR	OpenDataSoft
202	opendata_rubi_cat	http://opendata.rubi.cat	US	Socrata
203	opendata_socrata_com	http://opendata.socrata.com	US	Socrata
204	opendatacanarias_es	http://opendatacanarias.es/	ES	CKAN
205	opendatadc_org	http://opendatadc.org/	US	CKAN
206	opendatagortynia_gr	http://opendatagortynia.gr/	GR	CKAN
207	opendatahub_gr	http://opendatahub.gr/	GR	CKAN
208	opendatareno_org	http://opendatareno.org/	US	CKAN
209	opengov_es	http://opengov.es/	ES	CKAN
210	openresearchdata_ch	http://openresearchdata.ch/	CH	CKAN
211	opingogn_is	http://opingogn.is/	IS	CKAN
212	oppnadata_se	http://oppnadata.se	SE	CKAN
213	parisdata_opendatasoft_com	https://opendata.paris.fr/	FR	OpenDataSoft
214	performance_chattanooga_gov	http://performance.chattanooga.gov	US	Socrata
215	performance_smcgov_org	https://performance.smcgov.org	US	Socrata
216	performance_westsussex_gov_uk	http://performance.westsussex.gov.uk	GB	Socrata
217	pod_opendatasoft_com	http://pod.opendatasoft.com	US	OpenDataSoft
218	portal_openbelgium_be	http://portal.openbelgium.be	BE	CKAN
219	public_opendatasoft_com	http://public.opendatasoft.com	IE	OpenDataSoft
220	publicdata_eu	http://publicdata.eu	US	CKAN
221	rdw_azure-westeurope-prod_socrata_com	https://opendata.rdw.nl/	US	Socrata
222	reportcard_santamonicayouth_net	https://reportcard.santamonicayouth.net	US	Socrata
223	rs_ckan_net	http://rs.ckan.net/	RS	CKAN
224	scisf_opendatasoft_com	http://scisf.opendatasoft.com	US	OpenDataSoft
225	stat_cityofgainesville_org	https://stat.cityofgainesville.org	US	Socrata
226	tourisme04_opendatasoft_com	http://tourisme04.opendatasoft.com	FR	OpenDataSoft
227	tourisme62_opendatasoft_com	http://tourisme62.opendatasoft.com	FR	OpenDataSoft
228	transparenz_hamburg_de	http://transparenz.hamburg.de/	DE	CKAN
229	udct-data_aigid_jp	http://udct-data.aigid.jp	JP	CKAN
230	westsacramento_demo_socrata_com	https://westsacramento.demo.socrata.com	US	Socrata
231	wfp_demo_socrata_com_login	https://wfp.demo.socrata.com/login	US	Socrata
232	www_amsterdamopendata_nl	http://www.amsterdamopendata.nl	NL	CKAN
233	www_civicdata_io	http://www.civicdata.io	IO	CKAN
234	data_consumerfinance_gov	https://data.consumerfinance.gov	US	Socrata
235	www_criminalytics_org	https://www.criminalytics.org	US	Socrata
236	www_dallasopendata_com	https://www.dallasopendata.com	US	Socrata
237	www_data_gc_ca	https://open.canada.ca/en	CA	CKAN
238	www_data_go_jp	http://www.data.go.jp/	JP	CKAN
239	www_data_vic_gov_au	https://www.data.vic.gov.au/	AU	CKAN
240	www_datagm_org_uk	http://www.datagm.org.uk	GB	CKAN
241	www_daten_rlp_de	http://www.daten.rlp.de/	DE	CKAN
242	www_dati_friuliveneziagiulia_it	https://www.dati.friuliveneziagiulia.it	IT	Socrata
243	www_datos_misiones_gov_ar	http://www.datos.misiones.gov.ar/	AR	CKAN
244	www_edinburghopendata_info	https://edinburghopendata.info/	GB	CKAN
245	www_hri_fi	http://www.hri.fi/	FI	CKAN
246	www_metrochicagodata_com	http://www.metrochicagodata.com	US	Socrata
247	www_nosdonnees_fr	http://www.nosdonnees.fr/	FR	CKAN
248	www_odaa_dk	http://www.odaa.dk/	DK	CKAN
249	www_offene-daten_me	http://www.offene-daten.me	DE	CKAN
250	www_opendata-hro_de	http://www.opendata-hro.de/	DE	CKAN
251	www_opendata_provincia_roma_it	http://www.opendata.provincia.roma.it/	IT	CKAN
252	www_opendataforum_info	http://www.opendataforum.info/	BE	CKAN
253	www_opendatamalta_org	http://www.opendatamalta.org/	MT	CKAN
254	www_opendatanyc_com	http://www.opendatanyc.com	US	Socrata
255	www_opendataphilly_org	https://www.opendataphilly.org/	US	CKAN
256	www_opendataportal_at	https://www.opendataportal.at/	AT	CKAN
257	www_opengov-muenchen_de	https://www.opengov-muenchen.de/	DE	CKAN
258	www_rotterdamopendata_nl	http://www.rotterdamopendata.nl/	NL	CKAN
259	www_yorkopendata_org	http://www.yorkopendata.org/	GB	CKAN

Table A.4: List of data portals (cont'd).

A.3 Realizing the Queries from Section 5.2.2

As mentioned in Section 5.2.2, we extract the relevant RDF Data for constructing our knowledge graph from different Linked Data Sources, which provide RDF[1] data either in the form of dumps or via SPARQL endpoints, where we presented the respective SPARQL queries that theoretically should suffice to extract the data relevant for us in Section 5.2.2. A common problem with these sources is however that either such a SPARQL endpoint is not available or does not support complex queries. To this end, we discuss in this appendix how such limitations could be circumvented in the specific cases. We note that we expect the presented workaround could be similarly applied to other use cases for extracting relevant data from large RDF dumps or public endpoints, so we hope the discussion herein might be useful also for others.

A.3.1 Extracting postal codes and NUTS identifier from Wikidata

Due to the fact that the query in Figure 5.7 resulted in timeouts at the Wikidata SPARQL endpoint we split the query in sub-queries.[2] The task of extracting the NUTS identifier provides mappings for 1316 (out of 1716) NUTS codes. The missing 400 codes are NUTS regions where no Wikidata and/or GeoNames entry exists because, strictly speaking, there is no such corresponding administrative region. For instance, the Austrian NUTS regions AT126 and AT127 are called "Wiener Umland/Nordteil" and "Wiener Umland/Südteil", however, these are no political districts, but statistical entities grouping a *set* of districts Wikidata/GeoNames entity to map.

To complement the set of postal codes in Wikidata we use the extra postal code dataset by GeoNames[3] which consists of a total of 1.1M entries from 84 countries. For each code it provides a *place name*, and (depending on the country) several parent region/subdivion names. Based on these names we use a simple heuristic to map the postal codes to GeoNames entities: We split place names in the dataset by separators (white spaces, "-", "/")[4] and try to find GeoNames entries, in the respective country, with matching names.

A.3.2 Extracting Spatial Data from OSM

Since there exists – to the best of our knowledge – no up-to-date and integrated linked data version of OSM, we extract OSM relations, ways and nodes and map these to our

[1] We note OSM here as an exception; the JSON-data we extract from OSM is not directly in an RDF serializtation, but the provided JSON can be easily converted to JSON-LD.

[2] `SELECT ?s ?nuts ?geonames WHERE {?s wdt:P605 ?nuts. ?s wdt:P1566 ?geonames}` to get the NUTS-to-GeoNames mappings. Similarly for the postal code property `wdt:P281`.

[3] `http://download.geonames.org/export/zip/`, last accessed 2018-03-28

[4] We add this preprocessing step because there are many translated place names separated by slash or comma.

spatial knowledge graph. To do so we perform the following steps on a local extract of OSM:[5]

1. OSM provides different administrative levels for their relations, e.g., the relation which represents the states of a country, their subdivisions, and so on.[6] We use the alignment of these administrative levels with the previously introduced NUTS identifier to add the mappings to GeoNames: We perform lookups with the GeoNames labels of the NUTS 1, 2, and 3 regions at OSM's Nominatim service.[7] This service returns a set of potential candidate OSM relations for a given label. We select the correct relation (i.e. OSM region) by choosing the OSM relation at the same administrative/NUTS level as the corresponding GeoNames region.

2. Having the mapping for the countries' regions we again use OSM Nominatim to get the polygons for all sub-regions. These polygons can be used to extract any street names, places, etc. from a OSM data extract.[8]

3. We introduce relations between the extracted OSM entities and their parent GeoNames regions (which we get from the Nominatim mappings). This hierarchical relations are indicated using the `:osm_region` property in the conceptual SPARQL query in Figure 5.8, Section 5.2.2.

The OSM polygons returned by OSM's Nominatim service in Item 2 are not available as RDF, so we try to interpret the JSON from Nominatim as JSON-LD. This could be done relatively straightforwardly by adding to the JSON you get by e.g. calling `https://nominatim.openstreetmap.org/reverse?osm_id=1990594&osm_type=R&polygon_geojson=1&format=json` for obtaining the data for OSM id `1990594` (i.e. Vienna's district "Leopoldstadt", and extending the returned JSON with a JSON-LD [Sporny et al., 2014] context:

```
"@context": {
  "@vocab": "https://data.wu.ac.at/ns/osm#"
}
```

However, the query from Figure 5.8 still would not work "as is", since OSM returns the coordinates of its entities as GeoJSON [Butler et al., 2016], which due to the way that GeoJSON represents geometries as nested JSON arrays, is incompatible with JSON-LD.[9] We therefore pre-convert GeoJSONs nested way of representing polygon's to the format

[5] We use Geofabrik, `http://download.geofabrik.de/`, to download extracts of OSM on a country level.

[6] `http://wiki.openstreetmap.org/wiki/Tag:boundary%3Dadministrative`

[7] `http://nominatim.openstreetmap.org`

[8] OSM provides a tool, Osmosis `http://wiki.openstreetmap.org/wiki/Osmosis`, to process polygons on OSM data dumps

[9] There is ongoing work to fix it, which, however points to the same problem as an outstanding issue, cf. `https://github.com/json-ld/json-ld.org/issues/397`, retrieved 2018-03-29.

compatible with GeoSPARQL [Perry and Herring, 2012], by replacing JSON attributes of the form:

```
"geojson": {
  "type":"Polygon",
  "coordinates": [[[lat_1,long_1], ... , [lat_n,long_n]]]
}
```

with:

```
"geojson": {
  "type":"Polygon",
  "coordinates": "POLYGON(lat_1 long_1, ... , lat_n long_n)"
}
```

and extend the context to:

```
"@context": {
  "@vocab": "https://data.wu.ac.at/ns/osm#",
  "coordinates": {
    "@type": "http://www.opengis.net/ont/geosparql#wktLiteral"
  }
}
```

in a simple pre-processing step. The query in Figure 5.8 works as expected on this respectively pre-processed data from Nominatim.

A.3.3 Extracting Temporal Data from Wikidata

The query to extract event and time period data from Wikidata is shown in Figure 5.9; however as mentioned above, this query times out on the public endpoint. We note that Wikidata contained (at the time of writing) 4.8b RDF triples, so retrieving a dump and trying to extract the relevant information by setting up a local SPARQL endpoint also didn't seem an attractive solution. Rather, we propose a combination of

1. extracting relevant triples to answer the query via HDT [Fernández et al., 2013] and
2. executing targeted `CONSTRUCT` queries to the full SPARQL endpoint for specific sub-queries in order to materialize path expressions.

As for Item 1, we downloaded the complete Wikidata dump,[10] converted it locally to HDT [Fernández et al., 2013] and executed the following triple pattern queries over it to collect all data to match non-property-path triple patterns in Figure 5.9. We note that alternatively, we could have used Wikidata's Triple Pattern Fragment API [Verborgh et al., 2016] at `https://query.wikidata.org/bigdata/ldf` similarly.

[10] `https://www.wikidata.org/wiki/Wikidata:Database_download`

We then executed the following extraction queries separately on the dump, to extract the necessary component data:

```
CONSTRUCT WHERE {?S wp:P17 ?O} → 6613664 triples
CONSTRUCT WHERE {?S wp:P131 ?O} → 3928939 triples
CONSTRUCT WHERE {?S wp:P276 ?O} → 697238 triples
CONSTRUCT WHERE {?S wp:P580 ?O} → 26354 triples
CONSTRUCT WHERE {?S wp:P582 ?O} → 19241 triples
CONSTRUCT WHERE {?S wp:P585 ?O} → 91509 triples
CONSTRUCT WHERE {?S wp:P625 ?O} → 4158225 triples
```

In order to retrieve the remaining triples, that are instances of (subclasses of) the Wikidata classes of elections (`wd:Q40231`) and sports competitions (`wd:Q13406554`), we executed the following queries against the Wikidata SPARQL endpoint:

```
CONSTRUCT {
  ?S a wd:Q13406554. ?S rdfs:label ?label.
} WHERE {
  ?S wdt:P31/wdt:P279* wd:Q13406554.
  ?S rdfs:label ?label.
  FILTER( LANG(?label) = "en" ||
          LANG(?label) = "de" ||
          LANG(?label) = "" )
} → 418136 triples

CONSTRUCT {
  ?S a wd:Q40231. ?S rdfs:label ?label.
} WHERE {
  ?S wdt:P31/wdt:P279* wd:Q40231.
  ?S rdfs:label ?label.
  FILTER( LANG(?label) = "en" ||
          LANG(?label) = "de" ||
          LANG(?label) = "" )
} → 46899 triples
```

We then loaded these triples into a local triple store and executed the query in Figure A.2 on it, which is equivalent to the query in Figure 5.9, Section 5.2.2.

```
CONSTRUCT {
  ?event rdfs:label ?label ;
    dcterms:isPartOf ?Parent ;
    timex:hasStartTime ?StartDateTime ;
    timex:hasEndTime ?EndDateTime ;
    dcterms:coverage ?geocoordinates ;
    dcterms:spatial ?geoentity .
} WHERE {
  ?event rdfs:label ?label .
  {   # with a point in time or start end end date
    { ?event wdt:P585 ?StartDateTime.
     FILTER(?StartDateTime >
            "1900-01-01T00:00:00"^^xsd:dateTime)
    }
    UNION
    { ?event wdt:P580 ?StartDateTime.
      FILTER(?StartDateTime >
             "1900-01-01T00:00:00"^^xsd:dateTime)
      ?event wdt:P582 ?EndDateT.
      FILTER(DATATYPE(?EndDateT) = xsd:dateTime)}
  }
  OPTIONAL { ?event wdt:P361 ?Parent. }
  # specific spatialCoverage if available
  OPTIONAL {
    ?event wdt:P276?/(wdt:P17|wdt:P131) ?geoentity
  }
  OPTIONAL {
    ?event wdt:P276?/wdt:P625 ?geocoordinates
  }
  BIND ( if(bound(?EndDateT), ?EndDateT,
xsd:dateTime(concat(str(xsd:date(?StartDateTime)),
                    "T23:59:59")))
    AS ?EndDateTime )
}
```

Figure A.2: SPARQL query on local Wikidata extract - Namespaces as in A.1

Bibliography

[iso, 2013] (2013). *ISO 3166-1, Codes for the representation of names of countries and their subdivisions.* International Organization on Standardization.

[Abedjan et al., 2015] Abedjan, Z., Golab, L., and Naumann, F. (2015). Profiling relational data: a survey. *VLDB J.*, 24(4):557–581.

[Abele et al., 2017] Abele, A., McCrae, J. P., Buitelaar, P., Jentzsch, A., and Cyganiak, R. (2017). Linking open data cloud diagram 2017.

[Adelfio and Samet, 2013] Adelfio, M. D. and Samet, H. (2013). Schema extraction for tabular data on the web. *Proceedings of the VLDB Endowment*, 6(6):421–432.

[Agosti et al., 2006] Agosti, M., Candela, L., Castelli, D., Ferro, N., Ioannidis, Y., Koutrika, G., Meghini, C., Pagano, P., Ross, S., Schek, H. J., and Schuldt, H. (2006). A Reference Model for DLMSs Interim Report. Deliverable, DELOS.

[Agosti et al., 2007] Agosti, M., Ferro, N., Fox, E. A., and Gonçalves, M. A. (2007). Modelling dl quality: A comparison between approaches: The delos reference model and the 5s model. In *Second DELOS Conference on Digital Libraries*, pages 5–7, Tirrenia, Pisa, Italy.

[Alexander et al., 2011] Alexander, K., Cyganiak, R., Hausenblas, M., and Zhao, J. (2011). Describing Linked Datasets with the VoID Vocabulary. `https://www.w3.org/TR/void/`.

[Alobaid and Corcho, 2018] Alobaid, A. and Corcho, O. (2018). Fuzzy semantic labeling of semi-structured numerical datasets. In Faron Zucker, C., Ghidini, C., Napoli, A., and Toussaint, Y., editors, *Knowledge Engineering and Knowledge Management*, pages 19–33, Cham. Springer International Publishing.

[Arenas et al., 2014] Arenas, M., Barceló, P., Libkin, L., and Murlak, F. (2014). *Foundations of Data Exchange.* Cambridge University Press.

[Arenas et al., 2012] Arenas, M., Bertails, A., Prud'hommeaux, E., and Sequeda, J. (2012). A Direct Mapping of Relational Data to RDF. W3C Recommendation.

[Arenas and Libkin, 2005] Arenas, M. and Libkin, L. (2005). An information-theoretic approach to normal forms for relational and xml data. *J. ACM*, 52(2):246–283.

[Assaf et al., 2015] Assaf, A., Troncy, R., and Senart, A. (2015). HDL - Towards a harmonized dataset model for open data portals. In *PROFILES 2015, 2nd International Workshop on Dataset Profiling & Federated Search for Linked Data, Main conference ESWC15, 31 May-4 June 2015, Portoroz, Slovenia*, Portoroz, Slovenia. CEUR-WS.org.

[Attard et al., 2015] Attard, J., Orlandi, F., Scerri, S., and Auer, S. (2015). A systematic review of open government data initiatives. *Government Information Quarterly*, 32(4):399 – 418.

[Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. G. (2007). DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, pages 722–735.

[Auer and Lehmann, 2010] Auer, S. and Lehmann, J. (2010). Creating knowledge out of interlinked data. *Semantic Web*, 1(1-2):97–104.

[Bailey et al., 2005] Bailey, J., Bry, F., Furche, T., and Schaffert, S. (2005). Web and semantic web query languages: A survey. In *Reasoning Web, First International Summer School 2005*, pages 35–133, Msida, Malta.

[Batini et al., 2009] Batini, C., Cappiello, C., Francalanci, C., and Maurino, A. (2009). Methodologies for data quality assessment and improvement. *ACM Comput. Surv.*, 41(3):16:1–16:52.

[Beckett et al., 2014] Beckett, D., Berners-Lee, T., Prud'hommeaux, E., and Carothers, G. (2014). RDF 1.1 Turtle: The Terse RDF Triple Language. W3C Recommendation. `http://www.w3.org/TR/turtle/`.

[Beek et al., 2016] Beek, W., Rietveld, L., Schlobach, S., and van Harmelen, F. (2016). LOD laundromat: Why the semantic web needs centralization (even if we don't like it). *IEEE Internet Computing*, 20(2):78–81.

[Beno et al., 2017] Beno, M., Figl, K., Umbrich, J., and Polleres, A. (2017). Open data hopes and fears: Determining the barriers of open data. In *2017 Conference for E-Democracy and Open Government, CeDEM 2017, Krems, Austria, May 17-19, 2017*, pages 69–81.

[Berners-Lee, 1998] Berners-Lee, T. (1998). Semantic web road map. `https://www.w3.org/DesignIssues/Semantic.html`.

[Berners-Lee, 2006] Berners-Lee, T. (2006). Linked Data. W3C Design Issues. `http://www.w3.org/DesignIssues/LinkedData.html`.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, pages 29–37.

[Bernstein et al., 2016] Bernstein, A., Hendler, J. A., and Noy, N. F. (2016). A new look at the semantic web. *Commun. ACM*, 59(9):35–37.

[Bertot et al., 2012] Bertot, J. C., McDermott, P., and Smith, T. (2012). Measurement of open government: Metrics and process. *2014 47th Hawaii International Conference on System Sciences*, pages 2491–2499.

[Bischof et al., 2012] Bischof, S., Decker, S., Krennwallner, T., Lopes, N., and Polleres, A. (2012). Mapping between RDF and XML with XSPARQL. *J. Data Semantics*, 1(3):147–185.

[Bischof et al., 2018] Bischof, S., Harth, A., Kämpgen, B., Polleres, A., and Schneider, P. (2018). Enriching integrated statistical open city data by combining equational knowledge and missing value imputation. *J. Web Semant.*, 48:22–47.

[Bizer and Cyganiak, 2009] Bizer, C. and Cyganiak, R. (2009). Quality-driven information filtering using the WIQA policy framework. *J. Web Sem.*, 7(1):1–10.

[Bonatti et al., 2019] Bonatti, P. A., Decker, S., Polleres, A., and Presutti, V. (2019). Knowledge Graphs: New Directions for Knowledge Representation on the Semantic Web (Dagstuhl Seminar 18371). *Dagstuhl Reports*, 8(9):29–111.

[Borriello et al., 2016] Borriello, M., Dirschl, C., Polleres, A., Ritchie, P., Salliau, F., Sasaki, F., and Stoitsis, G. (2016). From xml to rdf step by step: approaches for leveraging xml workflows with linked data. In *XML Prague 2016 – Conference Proceedings*, pages 121–138, Prague, Czech Republic.

[Bourhis et al., 2017] Bourhis, P., Reutter, J. L., Suárez, F., and Vrgoc, D. (2017). JSON: data model, query languages and schema specification. *CoRR*, abs/1701.02221.

[Braunschweig et al., 2012] Braunschweig, K., Eberius, J., Thiele, M., and Lehner, W. (2012). The State of Open Data - Limits of Current Open Data Platforms. In *Proceedings of the International World Wide Web Conference, WWW 2012, Lyon, France*. ACM.

[Bray, 2014] Bray, T. (2014). The JavaScript Object Notation (JSON) Data Interchange Format. Internet Engineering Task Force (IETF) RFC 7159.

[Brickley et al., 2019] Brickley, D., Burgess, M., and Noy, N. F. (2019). Google dataset search: Building a search engine for datasets in an open web ecosystem. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1365–1375.

[Brickley and Guha, 2014] Brickley, D. and Guha, R. (2014). RDF Schema 1.1. W3C Recommendation. `http://www.w3.org/TR/rdf-schema/`.

[Butler et al., 2016] Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., and Schaub, T. (2016). The geojson format. RFC 7946, IETF.

[Cabrio et al., 2014] Cabrio, E., Aprosio, A. P., and Villata, S. (2014). These are your rights. In *Proceedings of the 11th Extended Semantic Web Conference (ESWC).*

[Calvanese et al., 2014] Calvanese, D., Fischl, W., Pichler, R., Sallinger, E., and Simkus, M. (2014). Capturing relational schemas and functional dependencies in RDFS. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 1003–1011.

[Candela et al., 2007] Candela, L., Castelli, D., Pagano, P., Thanos, C., Ioannidis, Y. E., Koutrika, G., Ross, S., Schek, H., and Schuldt, H. (2007). Setting the foundations of digital libraries: The DELOS manifesto. *D-Lib Magazine*, 13(3/4).

[Carothers and Seaborne, 2014] Carothers, G. and Seaborne, A. (2014). RDF 1.1 N-Triples: A line-based syntax for an RDF graph. W3C Recommendation. http://www.w3.org/TR/rdf-schema/.

[Chen and Cafarella, 2013] Chen, Z. and Cafarella, M. J. (2013). Automatic web spreadsheet data extraction. In *3RD International Workshop on Semantic Search over the Web, SSW '13, Riva del Garda, Italy, August 30, 2013*, pages 1:1–1:8.

[Cochez et al., 2017] Cochez, M., Ristoski, P., Ponzetto, S. P., and Paulheim, H. (2017). Global RDF vector space embeddings. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, pages 190–207.

[Corti et al., 2016] Corti, P., Lewis, B. G., Kralidis, T., and Mwenda, J. (2016). Implementing an open source spatio-temporal search platform for spatial data infrastructures. *PeerJ PrePrints*, 4:e2238.

[Crestan and Pantel, 2011] Crestan, E. and Pantel, P. (2011). Web-scale table census and classification. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 545–554.

[Cruz et al., 2013] Cruz, I. F., Ganesh, V. R., and Mirrezaei, S. I. (2013). Semantic extraction of geographic data from web tables for big data integration. In *Proceedings of the 7th Workshop on Geographic Information Retrieval*, GIR '13, pages 19–26, New York, NY, USA. ACM.

[Cyganiak et al., 2014] Cyganiak, R., Wood, D., Lanthaler, M., , Klyne, G., Carroll, J. J., and Mcbride, B. (2014). RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation. https://www.w3.org/TR/rdf11-concepts/.

[Das Sarma et al., 2012] Das Sarma, A., Fang, L., Gupta, N., Halevy, A., Lee, H., Wu, F., Xin, R., and Yu, C. (2012). Finding related tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 817–828. ACM.

[David et al., 2011] David, J., Euzenat, J., Scharffe, F., and dos Santos, C. T. (2011). The alignment API 4.0. *Semantic Web*, 2(1):3–10.

[de Dona et al., 2012] de Dona, M., Sloïm, E., Denis, L., and Bonny, F. (2012). *Qualité Web : Les bonnes pratiques pour amÈliorer vos sites.* Temesis.

[de Sompel et al., 2013] de Sompel, H. V., Nelson, M. L., and Sanderson, R. (2013). HTTP framework for time-based access to resource states - memento. *RFC*, 7089:1–50.

[Debattista et al., 2016] Debattista, J., Dekkers, M., Guéret, C., Lee, D., Mihindukulasooriya, N., and Zaveri, A. (2016). Data on the Web Best Practices: Data Quality Vocabulary (DQV). W3C Working Group Note. `https://www.w3.org/TR/vocab-dqv/`.

[Dell'Aglio et al., 2014] Dell'Aglio, D., Polleres, A., Lopes, N., and Bischof, S. (2014). Querying the web of data with XSPARQL 1.1. In *ISWC2014 Developers Workshop*, volume 1268 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Deng et al., 2019] Deng, L., Zhang, S., and Balog, K. (2019). Table2vec: Neural word and entity embeddings for table population and retrieval. In *42nd International ACM SIGIR Conf. on Research and Development in Information Retrieval.* To appear.

[Eberius et al., 2015a] Eberius, J., Braunschweig, K., Hentsch, M., Thiele, M., Ahmadov, A., and Lehner, W. (2015a). Building the dresden web table corpus: A classification approach. In *2nd IEEE/ACM International Symposium on Big Data Computing, BDC 2015, Limassol, Cyprus, December 7-10, 2015*, pages 41–50.

[Eberius et al., 2015b] Eberius, J., Thiele, M., Braunschweig, K., and Lehner, W. (2015b). Top-k entity augmentation using consistent set covering. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management, SSDBM '15, La Jolla, CA, USA, June 29 - July 1, 2015*, pages 8:1–8:12.

[Ehrlinger and Wöß, 2016] Ehrlinger, L. and Wöß, W. (2016). Towards a definition of knowledge graphs. In *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016.*

[Ermilov et al., 2013] Ermilov, I., Auer, S., and Stadler, C. (2013). User-driven semantic mapping of tabular data. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 105–112, New York, NY, USA. ACM.

[Ermilov and Ngomo, 2016] Ermilov, I. and Ngomo, A. N. (2016). TAIPAN: automatic property mapping for tabular data. In *Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings*, pages 163–179.

[Färber et al., 2018] Färber, M., Bartscherer, F., Menne, C., and Rettinger, A. (2018). Linked data quality of dbpedia, freebase, opencyc, wikidata, and YAGO. *Semantic Web*, 9(1):77–129.

[Färber et al., 2015] Färber, M., Ell, B., Menne, C., and Rettinger, A. (2015). A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web Journal*, 1(1):1–5.

[Fernández et al., 2013] Fernández, J. D., Martínez-Prieto, M. A., Gutiérrez, C., Polleres, A., and Arias, M. (2013). Binary RDF representation for publication and exchange (HDT). *Journal of Web Semantics*, 19:22–41.

[Fernández et al., 2019] Fernández, J. D., Umbrich, J., Polleres, A., and Knuth, M. (2019). Evaluating query and storage strategies for RDF archives. *Semantic Web*, 10(2):247–291.

[Ferro and Silvello, 2013] Ferro, N. and Silvello, G. (2013). NESTOR: A formal model for digital archives. *Inf. Process. Manage.*, 49(6):1206–1240.

[Fionda et al., 2016] Fionda, V., Chekol, M. W., and Pirrò, G. (2016). Gize: A time warp in the web of data. In *Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 19, 2016.*

[Fleischhacker et al., 2014] Fleischhacker, D., Paulheim, H., Bryl, V., Völker, J., and Bizer, C. (2014). Detecting errors in numerical linked data using cross-checked outlier detection. In *The Semantic Web - ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 357–372. Springer.

[Fox et al., 2012] Fox, E. A., Gonçalves, M. A., and Shen, R. (2012). *Theoretical Foundations for Digital Libraries: The 5S (Societies, Scenarios, Spaces, Structures, Streams) Approach*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers.

[Fürber and Hepp, 2011] Fürber, C. and Hepp, M. (2011). Towards a vocabulary for data quality management in semantic web architectures. In *Proceedings of the 1st International Workshop on Linked Web Data Management*, LWDM '11, pages 1–8, New York, NY, USA. ACM.

[Gentile et al., 2017] Gentile, A. L., Ristoski, P., Eckel, S., Ritze, D., and Paulheim, H. (2017). Entity matching on web tables: a table embeddings approach for blocking. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017.*, pages 510–513.

[Ghasemi-Gol and Szekely, 2018] Ghasemi-Gol, M. and Szekely, P. A. (2018). Tabvec: Table vectors for classification of web tables. *CoRR*, abs/1802.06290.

[Gil et al., 2011] Gil, Y., Szekely, P., Villamizar, S., Harmon, T. C., Ratnakar, V., Gupta, S., Muslea, M., Silva, F., and Knoblock, C. A. (2011). Mind your metadata: Exploiting semantics for configuration, adaptation, and provenance in scientific workflows. In Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., and Blomqvist, E., editors, *The Semantic Web – ISWC 2011*, pages 65–80, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Golden and Shaw, 2016] Golden, P. and Shaw, R. B. (2016). Nanopublication beyond the sciences: the periodo period gazetteer. *PeerJ Computer Science*, 2:e44.

[Gonçalves et al., 2004] Gonçalves, M. A., Fox, E. A., Watson, L. T., and Kipp, N. A. (2004). Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. *ACM Trans. Inf. Syst.*, 22(2):270–312.

[Gonçalves et al., 2007] Gonçalves, M. A., Moreira, B. L., Fox, E. A., and Watson, L. T. (2007). "what is a good digital library?" - A quality model for digital libraries. *Inf. Process. Manage.*, 43(5):1416–1437.

[Gottschalk and Demidova, 2018] Gottschalk, S. and Demidova, E. (2018). Eventkg: A multilingual event-centric temporal knowledge graph. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, pages 272–287.

[Greenberg et al., 2001] Greenberg, J., Pattuelli, M. C., Parsia, B., and Robertson, W. D. (2001). Author-generated dublin core metadata for web resources: A baseline study in an organization. In *DC-2001, Proceedings of the International Conference on Dublin Core and Metadata Applications 2001*, pages 38–46, Tokyo, Japan. National Institute of Informatics.

[Group, 2007] Group, O. G. W. (2007). Principles of open government data. `https://public.resource.org/8_principles.html`.

[Gurstein, 2011] Gurstein, M. B. (2011). Open data: Empowering the empowered or effective data use for everyone? *First Monday*, 16(2).

[Hahmann and Burghardt, 2010] Hahmann, S. and Burghardt, D. (2010). Connecting LinkedGeoData and Geonames in the Spatial Semantic Web. In *6th International GIScience Conference.*

[Halevy et al., 2016] Halevy, A. Y., Noy, N. F., Sarawagi, S., Whang, S. E., and Yu, X. (2016). Discovering structure in the universe of attribute names. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, Montreal, Canada, pages 939–949.

[Harris and Seaborne, 2013] Harris, S. and Seaborne, A. (2013). SPARQL 1.1 Query Language. W3C Recommendation. `http://www.w3.org/TR/sparql11-query/`.

[Harth and Gil, 2014] Harth, A. and Gil, Y. (2014). Geospatial data integration with linked data and provenance tracking. In *W3C/OGC Linking Geospatial Data Workshop*, pages 1–5.

[Harth et al., 2006] Harth, A., Umbrich, J., and Decker, S. (2006). Multicrawler: A pipelined architecture for crawling and indexing semantic web data. In *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, pages 258–271.

[Hassanzadeh et al., 2015] Hassanzadeh, O., Ward, M. J., Rodriguez-Muro, M., and Srinivas, K. (2015). Understanding a large corpus of web tables through matching with knowledge bases - an empirical study. In *Proceedings of the Tenth International Workshop on Ontology Matching (OM-2012)*.

[Heath and Bizer, 2011] Heath, T. and Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers.

[Heil and Neumaier, 2018] Heil, E. and Neumaier, S. (2018). reboting.com: Towards geo-search and visualization of austrian open data. In *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers*, pages 105–110.

[Hernández et al., 2015] Hernández, D., Hogan, A., and Krötzsch, M. (2015). Reifying RDF: what works well with wikidata? In *Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located with 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 11, 2015.*, pages 32–47.

[Hernández et al., 2016] Hernández, D., Hogan, A., Riveros, C., Rojas, C., and Zerega, E. (2016). Querying wikidata: Comparing sparql, relational and graph databases. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*, pages 88–103.

[Hitzler et al., 2014] Hitzler, P., Lehmann, J., and Polleres, A. (2014). Logics for the semantic web. In Gabbay, D. M., Siekmann, J. H., and Woods, J., editors, *Computational Logic*, volume 9 of *Handbook of the History of Logic*, pages 679–710. Elesevier.

[Hughes and Kamat, 2005] Hughes, B. and Kamat, A. (2005). A metadata search engine for digital language archives. *D-Lib Magazine*, 11(2).

[Huijboom and Van den Broek, 2011] Huijboom, N. and Van den Broek, T. (2011). Open data: an international comparison of strategies. *European journal of ePractice*, 12(1):4–16.

[Iannella and Villata, 2017] Iannella, R. and Villata, S. (2017). ODRL Information Model. W3C Working Draft. https://www.w3.org/TR/odrl-model/.

[International, 2019] International, O. K. (2019). Open Definition Conformant Licenses. From http://opendefinition.org/licenses/; retr. 13/02/2019.

[Janowicz et al., 2013] Janowicz, K., Scheider, S., and Adams, B. (2013). A geo-semantics flyby. In *Reasoning web. Semantic technologies for intelligent data access*, pages 230–250. Springer.

[Janssen et al., 2012] Janssen, M., Charalabidis, Y., and Zuiderwijk, A. (2012). Benefits, adoption barriers and myths of open data and open government. *IS Management*, 29(4):258–268.

[Jarke and Vassiliou, 1997] Jarke, M. and Vassiliou, Y. (1997). Data warehouse quality: A review of the DWQ project. In *Second Conference on Information Quality (IQ 1997)*, pages 299–313. MIT.

[Kacprzak et al., 2018] Kacprzak, E., Giménez-García, J. M., Piscopo, A., Koesten, L., Ibáñez, L.-D., Tennison, J., and Simperl, E. (2018). Making sense of numerical data - semantic labelling of web tables. In Faron Zucker, C., Ghidini, C., Napoli, A., and Toussaint, Y., editors, *Knowledge Engineering and Knowledge Management*, pages 163–178, Cham. Springer International Publishing.

[Kacprzak et al., 2019] Kacprzak, E., Koesten, L., Ibáñez, L. D., Blount, T., Tennison, J., and Simperl, E. (2019). Characterising dataset search - an analysis of search logs and data requests. *J. Web Semant.*, 55:37–55.

[Kacprzak et al., 2017] Kacprzak, E., Koesten, L. M., Ibáñez, L. D., Simperl, E., and Tennison, J. (2017). A query log analysis of dataset search. In *Web Engineering - 17th International Conference, ICWE 2017, Rome, Italy, June 5-8, 2017, Proceedings*, pages 429–436.

[Keßler and Farmer, 2015] Keßler, C. and Farmer, C. J. (2015). Querying and integrating spatial–temporal information on the web of data via time geography. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35:25 – 34. Geospatial Semantics.

[Klyne and Carroll, 2004] Klyne, G. and Carroll, J. J. (2004). Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation. https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[Kremen and Necaský, 2019] Kremen, P. and Necaský, M. (2019). Improving discoverability of open government data with rich metadata descriptions using semantic government vocabulary. *Journal of Web Semantics*, 55:1–20.

[Kubler et al., 2016] Kubler, S., Robert, J., Le Traon, Y., Umbrich, J., and Neumaier, S. (2016). Open data portal quality comparison using ahp. In *Proceedings of the*

17th International Digital Government Research Conference on Digital Government Research, pages 397–407a, Shanghai, China. ACM.

[Kubler et al., 2018] Kubler, S., Robert, J., Neumaier, S., Umbrich, J., and Traon, Y. L. (2018). Comparison of metadata quality in open data portals using the analytic hierarchy process. *Government Information Quarterly*, 35(1):13–29.

[Kucera et al., 2013] Kucera, J., Chlapek, D., and Necaský, M. (2013). Open government data catalogs: Current approaches and quality perspective. In *Technology-Enabled Innovation for Democracy, Government and Governance - Second Joint International Conference on Electronic Government and the Information Systems Perspective, and Electronic Democracy, EGOVIS/EDEM 2013, Prague, Czech Republic, August 26-28, 2013, Proceedings*, pages 152–166.

[Lebo et al., 2013] Lebo, T., Sahoo, S., and McGuinness, D. (2013). PROV-O: The PROV ontology. W3C Recommendation. `http://www.w3.org/TR/2013/REC-prov-o-20130430/`.

[Lehmann et al., 2015a] Lehmann, J., Athanasiou, S., Both, A., García-Rojas, A., Giannopoulos, G., Hladky, D., Le Grange, J. J., Ngomo, A.-C. N., Sherif, M. A., Stadler, C., et al. (2015a). Managing geospatial linked data in the geoknow project.

[Lehmann et al., 2015b] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., and Bizer, C. (2015b). Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

[Lehmann et al., 2015c] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al. (2015c). DBpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

[Lehmberg and Bizer, 2019] Lehmberg, O. and Bizer, C. (2019). Synthesizing n-ary relations from web tables. In *Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics*.

[Lehmberg et al., 2016] Lehmberg, O., Ritze, D., Meusel, R., and Bizer, C. (2016). A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, pages 75–76.

[Lenat, 1995] Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

[Li et al., 2015] Li, X., Cline, D. B. H., and Loguinov, D. (2015). Temporal Update Dynamics under Blind Sampling. *Ieee Infocom 2015*, pages 1634–1642.

[Lieberman et al., 2007] Lieberman, J., Singh, R., and Goad, C. (2007). W3C Geospatial Ontologies. *Incubator group report, W3C*.

[Lopez et al., 2012] Lopez, V., Kotoulas, S., Sbodio, M. L., Stephenson, M., Gkoulalas-Divanis, A., and Aonghusa, P. M. (2012). Queriocity: A linked data platform for urban information management. In *The Semantic Web - ISWC 2012*, pages 148–163.

[Maali and Erickson, 2014] Maali, F. and Erickson, J. (2014). Data Catalog Vocabulary (DCAT). W3C Recommendation. `http://www.w3.org/TR/vocab-dcat/`.

[Marchi and Miguel, 1974] Marchi, E. and Miguel, O. (1974). On the structure of the teaching-learning interactive process. *International Journal of Game Theory*, 3(2):83–99.

[Margaritopoulos et al., 2008] Margaritopoulos, T., Margaritopoulos, M., Mavridis, I., and Manitsaris, A. (2008). 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.

[Matuszek et al., 2006] Matuszek, C., Cabral, J., Witbrock, M. J., and DeOliveira, J. (2006). An introduction to the syntax and content of cyc. In *Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, Papers from the 2006 AAAI Spring Symposium, Technical Report SS-06-05, Stanford, California, USA, March 27-29, 2006*, pages 44–49.

[Meusel et al., 2014] Meusel, R., Petrovski, P., and Bizer, C. (2014). The webdatacommons microdata, rdfa and microformat dataset series. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pages 277–292.

[Meusel et al., 2016] Meusel, R., Ritze, D., and Paulheim, H. (2016). Towards more accurate statistical profiling of deployed schema.org microdata. *J. Data and Information Quality*, 8(1):3:1–3:31.

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

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

[Miller, 1995a] Miller, G. A. (1995a). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

[Miller, 1995b] Miller, G. A. (1995b). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

[Mitchell et al., 2018] Mitchell, T. M., Cohen, W. W., Jr., E. R. H., Talukdar, P. P., Yang, B., Betteridge, J., Carlson, A., Mishra, B. D., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E. A., Ritter, A., Samadi, M., Settles, B., Wang, R. C., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., and Welling, J. (2018). Never-ending learning. *Commun. ACM*, 61(5):103–115.

[Mitlöhner et al., 2016] Mitlöhner, J., Neumaier, S., Umbrich, J., and Polleres, A. (2016). Characteristics of open data CSV files. In *2nd International Conference on Open and Big Data, OBD 2016, Vienna, Austria, August 22-24, 2016*, pages 72–79.

[Moen et al., 1998] Moen, W. E., Stewart, E. L., and McClure, C. R. (1998). Assessing metadata quality: Findings and methodological considerations from an evaluation of the U.S. government information locator service (GILS). In *Proceedings of the IEEE Forum on Reasearch and Technology Advances in Digital Libraries, IEEE ADL '98, Santa Barbara, California, USA, April 22-24, 1998*, pages 246–255.

[Moreira et al., 2007] Moreira, B. L., Gonçalves, M. A., Laender, A. H. F., and Fox, E. A. (2007). 5squal: a quality assessment tool for digital libraries. In *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2007, Vancouver, BC, Canada, June 18-23, 2007, Proceedings*, page 513.

[Moro et al., 2014] Moro, A., Raganato, A., and Navigli, R. (2014). Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244.

[Najjar et al., 2003] Najjar, J., Ternier, S., and Duval, E. (2003). The actual use of metadata in ariadne: an empirical analysis. In *Proceedings of the 3rd ARIADNE Conference*, pages 1–6.

[Najork and Heydon, 2002] Najork, M. and Heydon, A. (2002). High-performance web crawling. In *Handbook of Massive Data Sets*, volume 4 of *Massive Computing*, pages 25–45. Springer US.

[Navigli and Ponzetto, 2012] Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250.

[Neumaier and Polleres, 2019] Neumaier, S. and Polleres, A. (2019). Enabling spatio-temporal search in open data. *Journal of Web Semantics*, 55:21 – 36.

[Neumaier et al., 2017a] Neumaier, S., Polleres, A., Steyskal, S., and Umbrich, J. (2017a). Data integration for open data on the web. In *Reasoning Web. Semantic Interoperability on the Web - 13th International Summer School 2017, London, UK, July 7-11, 2017, Tutorial Lectures*, pages 1–28.

[Neumaier et al., 2018] Neumaier, S., Savenkov, V., and Polleres, A. (2018). Geo-semantic labelling of open data. In *Proceedings of the 14th International Conference on Semantic Systems, SEMANTICS 2018, Vienna, Austria, September 10-13, 2018*, pages 9–20.

[Neumaier and Umbrich, 2016] Neumaier, S. and Umbrich, J. (2016). Measures for assessing the data freshness in open data portals. In *2nd International Conference on Open and Big Data, OBD 2016, Vienna, Austria, August 22-24, 2016*, pages 17–24.

[Neumaier et al., 2016a] Neumaier, S., Umbrich, J., Parreira, J. X., and Polleres, A. (2016a). Multi-level semantic labelling of numerical values. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, pages 428–445.

[Neumaier et al., 2016b] Neumaier, S., Umbrich, J., and Polleres, A. (2016b). Automated quality assessment of metadata across open data portals. *J. Data and Information Quality*, 8(1):2:1–2:29.

[Neumaier et al., 2017b] Neumaier, S., Umbrich, J., and Polleres, A. (2017b). Lifting data portals to the web of data. In *Workshop on Linked Data on the Web co-located with 26th International World Wide Web Conference (WWW 2017).*

[Ngomo et al., 2014] Ngomo, A. N., Auer, S., Lehmann, J., and Zaveri, A. (2014). Introduction to linked data and its lifecycle on the web. In *Reasoning Web. Reasoning on the Web in the Big Data Era - 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*, pages 1–99.

[Nguyen et al., 2018] Nguyen, P., Nguyen, K., Ichise, R., and Takeda, H. (2018). Embnum: Semantic labeling for numerical values with deep metric learning. In Ichise, R., Lecue, F., Kawamura, T., Zhao, D., Muggleton, S., and Kozaki, K., editors, *Semantic Technology*, pages 119–135, Cham. Springer International Publishing.

[Nickel et al., 2016] Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

[Nishida et al., 2017] Nishida, K., Sadamitsu, K., Higashinaka, R., and Matsuo, Y. (2017). Understanding the semantic structures of tables with a hybrid deep neural network architecture. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 168–174.

[Ochoa and Duval, 2009] Ochoa, X. and Duval, E. (2009). Automatic evaluation of metadata quality in digital repositories. *Int. J. on Digital Libraries*, 10(2-3):67–91.

[Orszag, 2009] Orszag, P. (2009). Open Government Directive. `https://www.whitehouse.gov/open/documents/open-government-directive`. Memorandum for the Heads of Executive Departments and Agencies.

[Oulabi and Bizer, 2019] Oulabi, Y. and Bizer, C. (2019). Extending cross-domain knowledge bases with long tail entities using web table data. In *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*, pages 385–396.

[Paulheim, 2017] Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508.

[Perry and Herring, 2012] Perry, M. and Herring, J. (2012). OGC GeoSPARQL - A geographic query language for RDF data. *OGC Implementation Standard. Sept.*

[Pezoa et al., 2016] Pezoa, F., Reutter, J. L., Suárez, F., Ugarte, M., and Vrgoc, D. (2016). Foundations of JSON schema. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 263–273.

[Pipino et al., 2002] Pipino, L., Lee, Y. W., and Wang, R. Y. (2002). Data quality assessment. *Commun. ACM*, 45(4):211–218.

[Polleres et al., 2018] Polleres, A., Kamdar, M. R., Fernández, J. D., Tudorache, T., and Musen, M. A. (2018). A more decentralized vision for linked data. In *Proceedings of the 2nd Workshop on Decentralizing the Semantic Web co-located with the 17th International Semantic Web Conference, DeSemWeb@ISWC 2018, Monterey, California, USA, October 8, 2018.*

[Pollock et al., 2015] Pollock, R., Tennison, J., Kellogg, G., and Herman, I. (2015). Metadata Vocabulary for Tabular Data. W3C Recommendation. `https://www.w3.org/TR/2015/REC-tabular-metadata-20151217/`.

[Posada-Sánchez et al., 2016] Posada-Sánchez, M., Bischof, S., and Polleres, A. (2016). Extracting geo-semantics about cities from openstreetmap. In *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016.*

[Primpeli et al., 2019] Primpeli, A., Peeters, R., and Bizer, C. (2019). The WDC training dataset and gold standard for large-scale product matching. In *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019.*, pages 381–386.

[Prohaska, 2017] Prohaska, G. (2017). Categorization and comparison of datasets across open data portals. Master's thesis, Vienna University of Economics and Business, Vienna, Austria. `https://aic.ai.wu.ac.at/~polleres/supervised_theses/Georg_Prohaska_2017MSc.pdf`.

[Pujara et al., 2013] Pujara, J., Miao, H., Getoor, L., and Cohen, W. (2013). Knowledge graph identification. In Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J. X., Aroyo, L., Noy, N., Welty, C., and Janowicz, K., editors, *The Semantic Web – ISWC 2013*, pages 542–557, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Ramnandan et al., 2015] Ramnandan, S. K., Mittal, A., Knoblock, C. A., and Szekely, P. A. (2015). Assigning semantic labels to data sources. In *ESWC 2015*, pages 403–417.

[Rastan, 2013] Rastan, R. (2013). Towards generic framework for tabular data extraction and management in documents. In *Proceedings of the Sixth Workshop on Ph.D. Students in Information and Knowledge Management*, PIKM '13, pages 3–10, New York, NY, USA. ACM.

[Reiche et al., 2014] Reiche, K. J., Höfig, E., and Schieferdecker, I. (2014). Assessment and Visualization of Metadata Quality for Open Government Data. In *Proceedings of the International Conference for E-Democracy and Open Government, CeDEM14, 2014, Krems, Austria, May 21-23, 2014*.

[Rijgersberg et al., 2013] Rijgersberg, H., van Assem, M., and Top, J. L. (2013). Ontology of units of measure and related concepts. *Semantic Web*, 4(1):3–13.

[Ritze et al., 2015] Ritze, D., Lehmberg, O., and Bizer, C. (2015). Matching html tables to dbpedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, WIMS '15, pages 10:1–10:6, New York, NY, USA. ACM.

[Ritze et al., 2016] Ritze, D., Lehmberg, O., Oulabi, Y., and Bizer, C. (2016). Profiling the potential of web tables for augmenting cross-domain knowledge bases. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 251–261, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

[Rojas et al., 2014] Rojas, L. A. R., Bermúdez, G. M. T., and Lovelle, J. M. C. (2014). Open data and big data: A perspective from colombia. In *International Conference on Knowledge Management in Organizations*, pages 35–41. Springer.

[Rospocher et al., 2016] Rospocher, M., van Erp, M., Vossen, P., Fokkens, A., Aldabe, I., Rigau, G., Soroa, A., Ploeger, T., and Bogaard, T. (2016). Building event-centric knowledge graphs from news. *J. Web Sem.*, 37-38:132–151.

[Rula et al., 2014] Rula, A., Palmonari, M., Ngonga Ngomo, A.-C., Gerber, D., Lehmann, J., and Bühmann, L. (2014). Hybrid acquisition of temporal scopes for rdf data. In Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., and Tordai, A., editors, *The Semantic Web: Trends and Challenges*, pages 488–503, Cham. Springer International Publishing.

[Saaty, 1996] Saaty, T. L. (1996). *Decision making with dependence and feedback: The analytic network process*, volume 4922. RWS publications Pittsburgh.

[Schmachtenberg et al., 2014] Schmachtenberg, M., Bizer, C., and Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pages 245–260.

[Shafranovich, 2005] Shafranovich, Y. (2005). Common Format and MIME Type for Comma-Separated Values (CSV) Files. RFC 4180 (Informational).

[Sieber and Johnson, 2015] Sieber, R. E. and Johnson, P. A. (2015). Civic open data at a crossroads: Dominant models and current challenges. *Government information quarterly*, 32(3):308–315.

[Spahiu et al., 2019] Spahiu, B., Maurino, A., and Meusel, R. (2019). Topic profiling benchmarks in the linked open data cloud: Issues and lessons learned. *Semantic Web*, 10(2):329–348.

[Spitz and Gertz, 2016] Spitz, A. and Gertz, M. (2016). Terms over LOAD: leveraging named entities for cross-document extraction and summarization of events. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 503–512.

[Sporny et al., 2014] Sporny, M., Kellogg, G., and Lanthaler, M. (2014). JSON-LD 1.0A JSON-based Serialization for Linked Data. `http://www.w3.org/TR/json-ld/`.

[Stadler et al., 2012] Stadler, C., Lehmann, J., Höffner, K., and Auer, S. (2012). LinkedGeoData: A core for a web of spatial open data. *Semantic Web*, 3(4):333–354.

[Steyskal and Polleres, 2014] Steyskal, S. and Polleres, A. (2014). Defining expressive access policies for linked data using the ODRL ontology 2.0. In *Proceedings of the 10th International Conference on Semantic Systems, SEMANTICS 2014*.

[Strong et al., 1997] Strong, D. M., Lee, Y. W., and Wang, R. Y. (1997). Data quality in context. *Communications of the ACM*, 40(5):103–110.

[Strötgen and Gertz, 2013] Strötgen, J. and Gertz, M. (2013). Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

[Suchanek et al., 2007] Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706.

[Sugimoto, 2014] Sugimoto, S. (2014). Digital archives and metadata as critical infrastructure to keep community memory safe for the future – lessons from japanese activities. *Archives and Manuscripts*, 42(1):61–72.

[Swartz, 2002] Swartz, A. (2002). Musicbrainz: A semantic web service. *IEEE Intelligent Systems*, 17(1):76–77.

[Swartz, 2013] Swartz, A. (2013). *Aaron Swartz's The Programmable Web: An Unfinished Work*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool Publishers.

[Syed et al., 2010] Syed, Z., Finin, T., Mulwad, V., and Joshi, A. (2010). Exploiting a Web of Semantic Data for Interpreting Tables. In *Proceedings of the Second Web Science Conference*.

[Taheriyan et al., 2014] Taheriyan, M., Knoblock, C. A., Szekely, P., and Ambite, J. L. (2014). A Scalable Approach to Learn Semantic Models of Structured Sources. In *Proceedings of the 8th IEEE International Conference on Semantic Computing (ICSC 2014)*.

[Taheriyan et al., 2013] Taheriyan, M., Knoblock, C. A., Szekely, P. A., and Ambite, J. L. (2013). A graph-based approach to learn semantic descriptions of data sources. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, pages 607–623.

[Tandy et al., 2015] Tandy, J., Herman, I., and Kellogg, G. (2015). Generating RDF from Tabular Data on the Web. W3C Recommendation. `https://www.w3.org/TR/csv2rdf/`.

[Tanon et al., 2016] Tanon, T. P., Vrandecic, D., Schaffert, S., Steiner, T., and Pintscher, L. (2016). From freebase to wikidata: The great migration. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 1419–1428.

[Tran and Alrifai, 2014] Tran, G. B. and Alrifai, M. (2014). Indexing and analyzing wikipedia's current events portal, the daily news summaries by the crowd. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 511–516, New York, NY, USA. ACM.

[Tygel et al., 2016] Tygel, A., Auer, S., Debattista, J., Orlandi, F., and Campos, M. L. M. (2016). Towards cleaning-up open data portals: A metadata reconciliation approach. In *Tenth IEEE International Conference on Semantic Computing, ICSC 2016, Laguna Hills, CA, USA, February 4-6, 2016*, pages 71–78.

[Umbrich et al., 2015] Umbrich, J., Neumaier, S., and Polleres, A. (2015). Quality assessment & evolution of open data portals. In *The International Conference on Open and Big Data*, pages 404–411, Rome, Italy. IEEE.

[Veljković et al., 2014] Veljković, N., Bogdanović-Dinić, S., and Stoimenov, L. (2014). Benchmarking open government: An open data perspective. *Government Information Quarterly*, 31(2):278 – 290.

[Venetis et al., 2011] Venetis, P., Halevy, A. Y., Madhavan, J., Pasca, M., Shen, W., Wu, F., Miao, G., and Wu, C. (2011). Recovering semantics of tables on the web. *PVLDB*, 4(9):528–538.

[Verborgh et al., 2016] Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., and Colpaert, P. (2016). Triple Pattern Fragments: a low-cost knowledge graph interface for the Web. *Journal of Web Semantics*, 37–38:184–206.

[Vrandecic and Krötzsch, 2014] Vrandecic, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

[Wang et al., 2012] Wang, J., Wang, H., Wang, Z., and Zhu, K. Q. (2012). Understanding tables on the web. In *Conceptual Modeling - 31st International Conference ER 2012, Florence, Italy*, pages 141–155.

[Wang and Strong, 1996] Wang, R. Y. and Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *J. of Management Information Systems*, 12(4):5–33.

[Weibel et al., 1998] Weibel, S., Kunze, J. A., Lagoze, C., and Wolf, M. (1998). Dublin core metadata for resource discovery. *RFC*, 2413:1–8.

[Wienand and Paulheim, 2014] Wienand, D. and Paulheim, H. (2014). Detecting incorrect numerical data in DBpedia. In *ESWC 2014, Anissaras, Crete, Greece*, pages 504–518.

[Wilkinson et al., 2016] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3.

[Wilson, 2007] Wilson, A. J. (2007). Toward releasing the metadata bottleneck - a baseline evaluation of contributor-supplied metadata. *Library Resources & Technical Services*, 51(1):16–28.

[World Wide Web Foundation, 2015] World Wide Web Foundation (2015). Open Data Barometer.

[Yildiz et al., 2005] Yildiz, B., Kaiser, K., and Miksch, S. (2005). pdf2table: A method to extract table information from PDF files. In *Proceedings of the 2nd Indian International Conference on Artificial Intelligence, Pune, India, December 20-22, 2005*, pages 1773–1785.

[Zaveri et al., 2015] Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., and Auer, S. (2015). Quality assessment for linked data: A survey. *Semantic Web Journal*, 7(1):63–93.

[Zhang and Chakrabarti, 2013] Zhang, M. and Chakrabarti, K. (2013). Infogather+: Semantic matching and annotation of numeric and time-varying attributes in web tables. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 145–156, New York, NY, USA. ACM.

[Zhang and Balog, 2018] Zhang, S. and Balog, K. (2018). Ad hoc table retrieval using semantic similarity. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1553–1562.

[Zhang, 2017] Zhang, Z. (2017). Effective and efficient semantic table interpretation using tableminer^{+}. *Semantic Web*, 8(6):921–957.

[Zhu et al., 2012] Zhu, H., Madnick, S. E., Lee, Y. W., and Wang, R. Y. (2012). Data and Information Quality Research: Its Evolution and Future. In *Computing Handbook, Third Edition: Information Systems and Information Technology*, pages 16: 1–20. CRC Press, USA.

[Zuiderwijk et al., 2012] Zuiderwijk, A., Janssen, M., Choenni, S., Meijer, R., and Alibaks, R. S. (2012). Socio-technical Impediments of Open Data. *Electronic Journal of e-Government*, 10(2):156–172.