Lifting Data Portals to the Web of Data

Or: Linked Open Data (Portals) – Now for real!

Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres
Vienna University of Economics and Business
firstname.lastname@wu.ac.at

ABSTRACT

Data portals are central hubs for freely available (governmental) datasets. These portals use different software frameworks to publish their data, and the metadata descriptions of these datasets come in different schemas accordingly to the framework. The present work aims at re-exposing and connecting the metadata descriptions of currently 854k datasets on 261 data portals to the Web of Linked Data by mapping and publishing their homogenized metadata in standard vocabularies such as DCAT and Schema.org. Additionally, we publish existing quality information about the datasets and further enrich their descriptions by automatically generated metadata for CSV resources. In order to make all this information traceable and trustworthy, we annotate the generated data using the W3C's provenance vocabulary. The dataset descriptions are harvested weekly and we offer access to the archived data by providing APIs compliant to the Memento framework. All this data – a total of about 120 million triples per weekly snapshot – is queryable at the SPARQL endpoint at data.wu.ac.at/portalwatch/sparql.

1 INTRODUCTION

Open data portals, such as Austria's data.gv.at or UK's data.gov.uk, are central points of access for freely available (governmental) datasets. Government departments publish all kind of data on these portals, for instance, about economy, demography, public spendings, etc., which improves government transparency, accountability and public participation. These data catalogs mostly use software frameworks, such as CKAN[1] or Socrata[2] to publish their data. However, the metadata descriptions for the published data on these portals is only partially available as Linked Data. This work aims at exposing and connecting the datasets' metadata of various portals to the Web of Linked Data by mapping and publishing their metadata in standard vocabularies such as DCAT and Schema.org. Providing the datasets in such a way enables "to link this structured metadata with information describing locations, scientific publications, or even [knowledge graphs], facilitating data discovery for others."[3]

CKAN and Socrata use their own metadata schemas for describing the published datasets. Further, these data portal frameworks allow the portal providers to extend their schemas with additional metadata keys. This potentially leads to diverse and heterogeneous metadata across different data catalogs. Also, CKAN's and Socrata's metadata schemas do not contain links to any external knowledge and existing ontologies, and no links to other datasets and other data catalogs. This lack of external references implies the risk of rather having data silos instead of connected and interlinked data portals.

The W3C identified the issue of heterogeneous metadata schemas across the data portals, and proposed an RDF vocabulary to solve this issue: The metadata standard DCAT [8] (Data Catalog Vocabulary) describes data catalogs and corresponding datasets. It models the datasets and their distributions (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 the DCAT standard for describing their metadata. 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.

While mapping data portals' metadata to DCAT is certainly a step towards a better interlinking of open datasets, the major search engines do not 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 Vault[4]) we further map and publish the DCAT metadata descriptions using Schema.org's Dataset vocabulary.[5]

In order to tackle the aim of better integration of datasets on several fronts, we do not only want to expose the metadata descriptions as Linked Data in an homogenized representation, but also improve the descriptions of the actual data. Since CSV is the predominant format on open data portals [10], we make use of the W3C's metadata vocabulary for describing CSVs on the Web.[6] This recommendation gives the data publisher a standardized way of describing the dialect, columns, datatypes, etc. of tabular data.

Figure 1 displays the different components and the integration of the overall system. Initially, our framework collects the dataset descriptions (as JSON documents) and maps it to a common representation – the DCAT vocabulary. Then it computes for each dataset a quality assessment, and (in case of CSV resources) additional CSV metadata. We model this additional information using W3C vocabularies (DQV and CSVW) and connect the data to the DCAT

[1] http://ckan.org/
[2] https://socrata.com/
[3] Google Research Blog entry, https://research.googleblog.com/2017/01/facilitating-discovery-of-public.html, last accessed 2017-01-27
[4] https://www.google.com/intl/bn/insidesearch/features/search/knowledge.html
[5] https://schema.org/Dataset
[6] https://www.w3.org/2013/csvw/

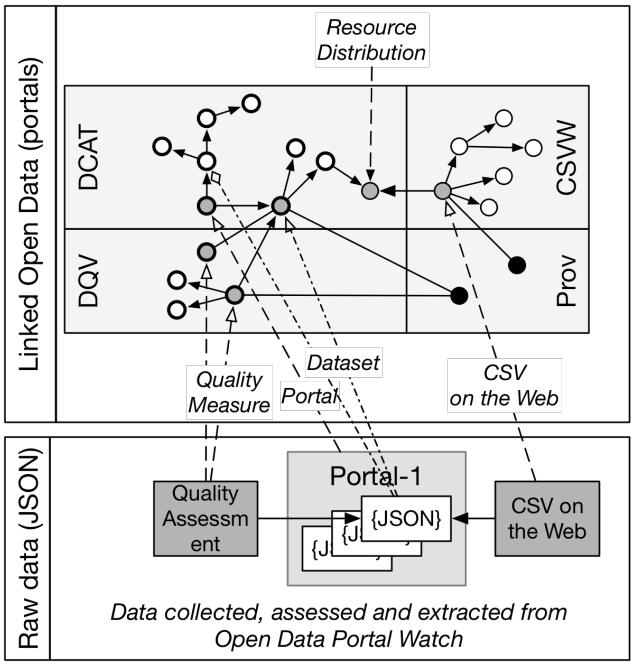

Figure 1: The collected dataset descriptions get re-exposed and enriched by quality measures and additional metadata.

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 paper we make the following concrete contributions:

- A system that re-exposes data extracted from open data portal APIs such as CKAN. The output formats include a subset of W3C's DCAT with extensions and Schema.org's Dataset-oriented vocabulary (Section 3).
- 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 DCAT, and highlight potential design issues/improvements in the target vocabularies (i.e., W3C's DCAT and Schema.org's Dataset vocabulary), see Section 3.1.
- 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).
- We present heuristics to further enrich descriptions of tabular data semi-automatically by auto-generating additional metadata using the vocabulary defined by the W3C CSV on the Web working group, which we likewise add to our enriched metadata. Additionally, as a by-product, a user interface to generate and refine such CSV metadata is available (Section 5).
- We use the PROV ontology to record and annotate the provenance of our generated/published data (which is partially generated by using heuristics). The details are described in Section 6.
- All the integrated, enriched and versioned metadata is publicly available as Linked Data at http://data.wu.ac.at/portalwatch/. Additionally, a SPARQL endpoint to query the generated RDF, described in Section 7.1.
- Finally, we enable historic access to the original and mapped dataset descriptions using the Memento framework [13], cf. Section 7.2.

We conclude and give an outlook on future work in Section 8.

2 BACKGROUND & RELATED WORK

The recent DCAT application profile for data portals in Europe (DCAT-AP)[7] extends the DCAT core vocabulary and aims towards the integration of datasets from different European data portals. In its current version (v1.1) it extends the existing DCAT schema by a set of additional properties. DCAT-AP allows to specify the version and the period of time 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`.

An earlier approach, in 2011, is the the VoID vocabulary [2] published by 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 2011 Fürber and Hepp [6] 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) and provides a total of about 50 classes and 50 properties. The ontology 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.

More recently, in 2015 Assaf et al. [3] propose HDL, an harmonized 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.

Portal Watch. The contributions of this paper are based on and build upon the Open Data Portal Watch project [10]. Portal Watch is a framework for monitoring and quality assessment of (governmental) Open Data portals, see http://data.wu.ac.at/portalwatch. It monitors data from portals using the CKAN, Socrata, and OpenDataSoft software, as well as portals providing their metadata in the DCAT RDF vocabulary.

[7] https://joinup.ec.europa.eu/asset/dcat_application_profile/description

Table 1: Monitored portals and datasets in Portal Watch

	total	CKAN	Socrata	ODSoft	DCAT
portals	261	149	99	11	2
datasets	854,013	767,364	81,268	3,340	2,041
URLs	2,057,924	1,964,971	104,298	12,398	6,092

Currently, as of the second week of 2017, the framework monitors 261 portals, which describe in total about 854k datasets with more than 2 million distributions, i.e., URLs (cf. Table 1). As we monitor and crawl the metadata of these portals in a weekly fashion, we can use the gathered insights in two ways to enrich the integrated metadata of these portals: namely, (i) we publish and serve the integrated metadata descriptions in a weekly, versioned manner, (ii) we enrich these matadata desriptions by the quality assessment metrics defined in [10].

3 GENERATING DCAT AND SCHEMA.ORG

The DCAT model suggests 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 contains 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 detail in [10]). 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.[8] For CKAN metadata, this mapping is mainly based on an existing CKAN extension to export RDF serializations of CKAN datasets based on DCAT.[9] This extension is maintained by the Open Knowledge Foundation and the source code is also in use in the Portal Watch framework (in a slightly adapted form).

As a next step, we generate 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.[10] The three main DCAT classes `Catalog`, `Dataset`, and `Distribution` are mapped to the Schema.org classes `DataCatalog`, `Dataset`, and `DataDownload` (cf. Table 2). The mapping defined in the W3C draft covers all core properties specified in the DCAT recommendation except for `dcat:identifier` and `dcat:frequency`.

[8] `dct:` is the Dublin Core Terms namespace

[9] https://github.com/ckan/ckanext-dcat

[10] https://www.w3.org/2015/spatial/wiki/ISO_19115_-_DCAT_-_Schema.org_mapping, last accessed 2016-12-26

Table 2: Mapping of main classes and missing properties

DCAT	Schema.org
`dcat:Catalog`	`schema:DataCatalog`
`dcat:Dataset`	`schema:Dataset`
`dcat:Distribution`	`schema:DataDownload`
`dcat:identifier`	?
`dcat:frequency`	?

All mapped dataset descriptions for all weekly harvested versions (hereafter referred to as *snapshots*) are accessible via the Portal Watch API (http://data.wu.ac.at/portalwatch/api/v1):

`/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 http://data.wu.ac.at/odso (and http://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.[11]

3.1 Challenges and mapping issues

3.1.1 DCAT mapping available on CKAN portals. The aforementioned CKAN-to-DCAT extension defines mappings for CKAN datasets and their resources to the corresponding DCAT classes dcat:Dataset and dcat:Distribution and offers it 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 include additional metadata fields in 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 conversion of the CKAN metadata contains these extra fields. Depending on the version and the configuration of the export-extension there are 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

[11] https://www.w3.org/TR/json-ld-syntax/#embedding-json-ld-in-html-documents

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.:

```
<http://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.:

```
<http://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.

3.1.2 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 3 lists the most frequently used fields sorted by the number of portals they appear in; most frequent `spatial` in 29 portals. Most of these cross-portal extra keys are generated by widely used CKAN extensions. The keys in Table 3 are all generated by the harvesting[12] and spatial extension.[13] 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 3. In case of an ? cell, we were not able to choose an appropriate DCAT core property.

Table 3: 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

Looking into more detail of these 3746 extra keys, we discovered that 1553 unique keys are of the form `links:{`*dataset-id*`}`, e.g., `links:air-temperature` or `links:air-pressure`. All these `links:`-keys originate from the datahub.io portal, which provides references to Linked Data as CKAN datasets. The portal uses these keys to encode links between two datasets within the portal. While this information is certainly beneficial, the encoding of links between datasets (i.e., using the metadata key as link) shows the need for expressing relations between datasets in existing data portals.

Table 4: 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 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.[14]

In future work we want to further use information in the available extra keys. For instance, given the information added by the keys of the harvesting extension we could introduce `owl:sameAs` references between the harvested and the source datasets.

3.1.3 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. A DCAT distribution is defined the following way: "*Represents a specific available form of a dataset. Each dataset might be available in different forms, these forms might represent different formats of the dataset or different endpoints. [...]*"[15] 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. 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 5). Out of the 401k multi-resource datasets, for 140k datasets all corresponding file formats are different, indicating that these are possibly *distributions* of the dataset.

[12] http://extensions.ckan.org/extension/harvest/
[13] http://docs.ckan.org/projects/ckanext-spatial/en/latest/
[14] 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 3).
[15] https://www.w3.org/TR/vocab-dcat/#class-distribution

Using string similarity we encountered similar titles for at least two resources in 261k out of the 401k multi-resource datasets.

Table 5: 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 5) 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 USING THE DATA QUALITY VOCABULARY

In this section we summarize the quality assessment performed by the Portal Watch framework and we detail how this measurements are published and connected to the corresponding datasets.

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.

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 [10].

The W3C's Data Quality Vocabulary[16] (DQV) 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 DQV to make the quality measures of the Portal Watch framework as RDF available and to link the assessment to the dataset descriptions. Figure 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.

API access to the measurements. The DQV results can be retrieved by using the following API or by querying the SPARQL endpoint (see Section 7.1):

`/portal/{portalid}/{snapshot}/dataset/{datasetid}/dqv`
: Analogous to the previous APIs (see Section 3), 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).

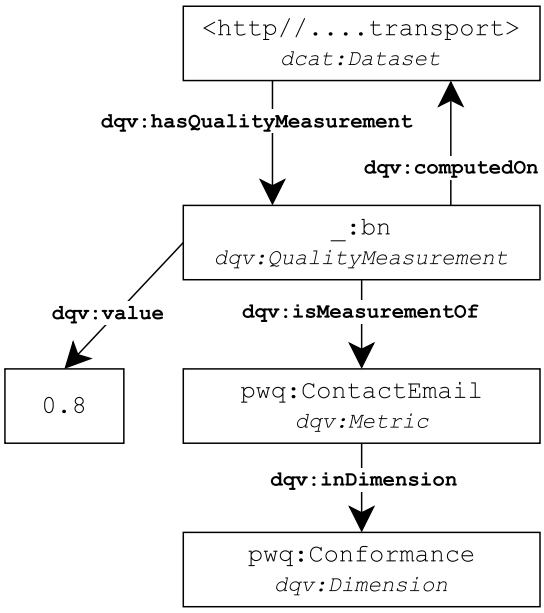

Figure 2: Example quality measurement using DQV

5 USING W3C'S CSVW METADATA

The W3C's CSV on the Web Working Group[17] (CSVW) proposed a metadata vocabulary that describes how CSV data (comma-separated-value files or similar tabular data sources) should be interpreted [11]. 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 filter out 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[18] 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 sniffed lines and

[16] https://www.w3.org/TR/vocab-dqv/

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

[18] https://pypi.python.org/pypi/python-magic/

modifying the preferred delimiters); the Python module is online available.[19]

- 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.[20]
- 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[21] `number`, `binary`, `datetime` and `anyAtomicType`.

This acquired information is then used to generate RDF which is compliant to the CSVW metadata vocabulary [11]. Figure 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).

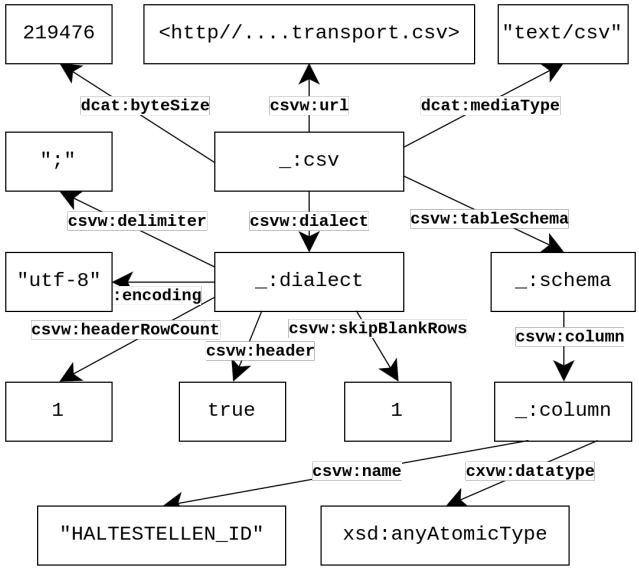

Figure 3: Example of an generated CSVW metadata

For our latest snapshot (third week of 2017) there were a total of 222k URLs with "CSV" in the dataset's metadata description. 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.

In future work we want to increase this number of analysed CSVs. There are CSV files with missing and wrong format descriptions, which could be detected by using the file extensions and the HTTP media type of the resources.

5.1 Assisted generation of CSVW metadata

The Portal Watch includes for each portals' CSV files a link to an pre-filled UI form. This form allows to further describe the name, datatype and properties of each detected column, and is pre-fill with the following detected dialect description fields: commentPrefix (is there a prefix for leading comment lines), doubleQuote (use of "" as escape character for quotation), delimiter, encoding, header, headerRowCount (number of header rows), lineTerminators, quoteChar (the character used for quotation).

The generated CSVW metadata can be downloaded as JSON-LD in order to publish it along with the corresponding CSV file. Figure 4 displays the form: "table description" provides general descriptions such as language, title, and publisher; "column description" provides properties for each column separately; "dialect description" allows the description and modification of the detected CSV dialect.

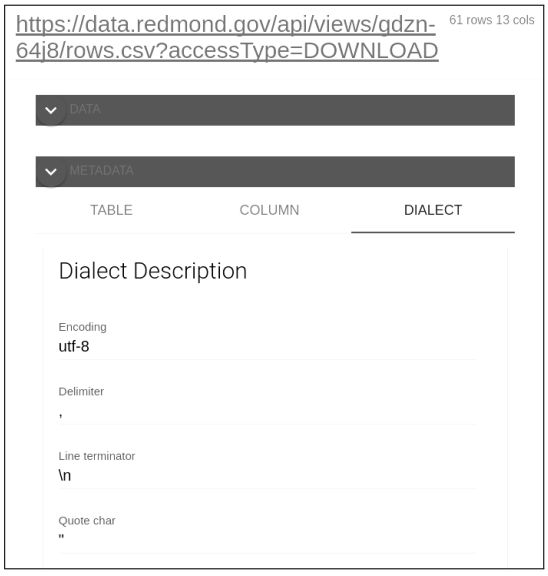

Figure 4: Pre-filled interface for generating CSV metadata

6 ADDING PROVENANCE ANNOTATIONS

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,

[19] https://github.com/sebneu/anycsv

[20] 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.

[21] http://www.w3.org/2001/XMLSchema

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 [7] 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. On a high level PROV distinguishes between *entities*, *agents*, and *activities*. *Entities* can be all kinds of things, digital or not, which are created or modified. *Activities* are the processes which create or modify entities. An *agent* is something or someone who is responsible for an activity (and indirectly also for an entity). Additionally PROV also allows to tag certain activities with time, for example a timestamp when an entity was created.

To add provenance information to our generated RDF data we define a `prov:SoftwareAgent` (a subclass of `prov:Agent`) with URI `<http://data.wu.ac.at/portalwatch>`, cf. Figure 5. 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.

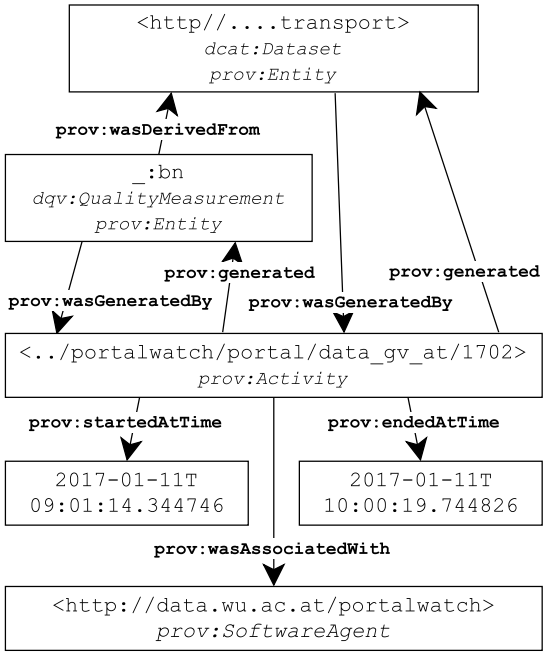

Figure 5: Provenance annotation to quality measurement

Regarding the (heuristically) generated CSVW metadata, we annotate all `_:csv` resources (cf. Section 5) as `prov:Entity` and associate them with a `prov:Activity` with URI `<http://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.

7 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 6 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.

7.1 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 (http://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.[22]

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). Additionally, the endpoint re-uses the namespaces as displayed in Listing 1.

```
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX dqv: <http://www.w3.org/ns/dqv#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX csvw: <http://www.w3.org/ns/csvw#>
PREFIX pwq: <http://data.wu.ac.at/
                    portalwatch/quality#>
```

Listing 1: Used namespaces

7.1.1 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 2 returns all portals in the Portal Watch system which use *transportation* as a keyword/tag (in total 31 portals).

7.1.2 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.

[22] https://virtuoso.openlinksw.com/

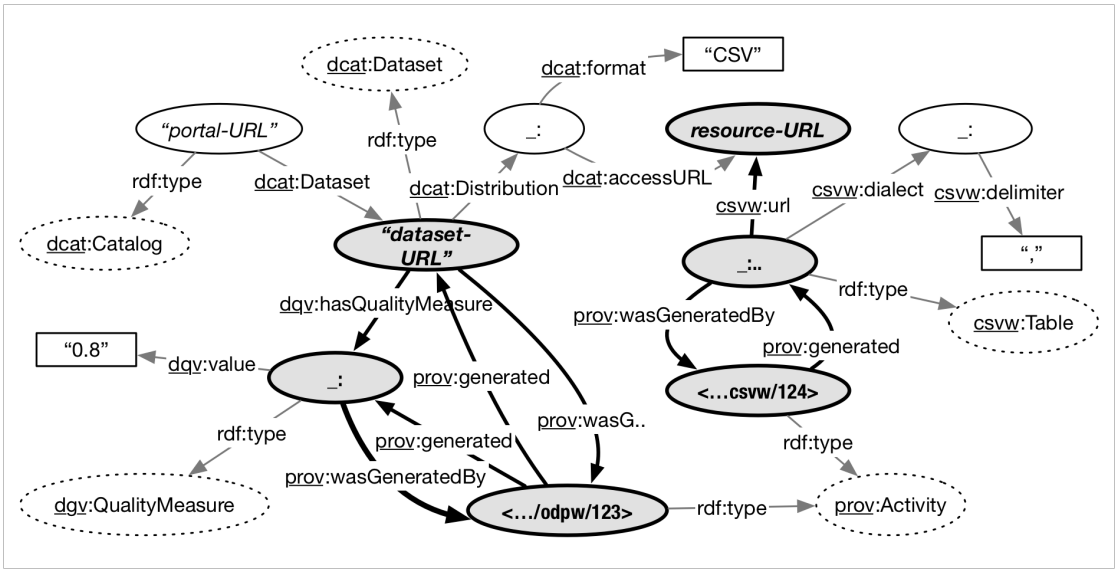

Figure 6: Properties and nodes which enable connections between generated datasets

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

Listing 2: All portals holding transportation data

For instance, the query in Listing 3 lists an aggregation of the ContactEmail quality metric (see [10] for definitions) for the organizations (i.e. publisher) on the Austrian data portal `data.gv.at`.

```
SELECT ?orga AVG(?v)
WHERE {
  <http://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 3: Average *ContactEmail* metric per organization

7.2 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 [13]. 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*[23] 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 `<http://data.wu.ac.at/portalwatch/api/v1/memento/ {portalid} / {datasetid} >` using the internal portal-ID and the dataset's ID; e.g., *uri-g*[24] 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 `<http://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*[25]; this URI returns the archived dataset description closest after January 1 2017.

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

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

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

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 [13]. In order to support datetime negotiation within our Memento implementation we implemented these headers for the available URI-G and Memento resources.

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 5). 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 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: 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 5 shows the HTTP response header to the request to *uri-g* in Listing 4. 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.

```
HTTP/1.0 200 OK
Content-Type: application/json
Memento-Datetime: Sun, 25 Dec 2016 23:00:00 GMT
Link: <http://www.data.gv.at/katalog/dataset/
    add66f20-d033-4eee-b9a0-47019828e698>; rel="
    original",
      <http://data.wu.ac.at/portalwatch/api/v1/
          memento/data_gv_at/add66f20-d033-4eee-
          b9a0-47019828e698>; rel="timegate"
Vary: accept-datetime
Content-Location: http://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 5: Response from *uri-g* to request of Listing 4

8 CONCLUSION

In this work we have extended the existing Portal Watch system such that it re-exposes the dataset descriptions of 261 data portals as RDF data using the DCAT and Schema.org vocabulary. We additionally publish quality measurements along several dimensions for each dataset descriptions, using the W3C's Data Quality vocabulary, and we further enriched 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 contain provenance annotations. To allow users access to archived versions of the dataset descriptions the Portal Watch framework offers APIs based on the Memento framework: time-based content negotiation on top of the HTTP protocol. As a next step for our framework we plan to address the following issues:

8.1 Future Work

Automatically generate richer CSVW metadata. We plan to improve the CSV analysis and generated richer 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.

Complementary, we want to further improve the assisted generation of CSVW metadata by combining our dialect and datatype detection with approaches to (semi-)automatically annotate classes to entities and properties to columns in CSVs [15].

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 [4] Fernandez et al. benchmark existing RDF archiving 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. [5] 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.

In order to add such links and connections we plan 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 can be build upon [1, 9, 14].

A recent approach by Tygel et al. [12] 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.

ACKNOWLEDGEMENTS

This work has been supported by the Austrian Research Promotion Agency (FFG) under the project ADEQUATe (grant no. 849982).

REFERENCES

[1] Marco D. Adelfio and Hanan Samet. Schema extraction for tabular data on the web. *PVLDB*, 6(6):421–432, 2013.
[2] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing Linked Datasets with the VoID Vocabulary. https://www.w3.org/TR/void/, March 2011.
[3] Ahmad Assaf, Raphaël Troncy, and Aline Senart. 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, 05 2015. CEUR-WS.org.
[4] Javier David Fernandez Garcia, Jürgen Umbrich, and Axel Polleres. Bear: Benchmarking the efficiency of rdf archiving. 2015.
[5] Valeria Fionda, Melisachew W Chekol, and Guiseppe Pirrò. A time warp in the web of data. In *15h Int. Semantic Web Conference (ISWC) Posters and Demos, Kobe, Japan*, 2016.
[6] Christian Fürber and Martin Hepp. 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, 2011. ACM.
[7] Timothy Lebo, Satya Sahoo, and Deborah McGuinness. PROV-O: The PROV ontology. http://www.w3.org/TR/2013/REC-prov-o-20130430/, April 2013. W3C Recommendation.
[8] Fadi Maali and John Erickson. Data Catalog Vocabulary (DCAT). http://www.w3.org/TR/vocab-dcat/, January 2014. W3C Recommendation.
[9] Sebastian Neumaier, Jürgen Umbrich, Josiane Xavier Parreira, and Axel Polleres. 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, 2016.
[10] Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. Automated quality assessment of metadata across open data portals. *J. Data and Information Quality*, 8(1):2:1–2:29, 2016.
[11] Rufus Pollock, Jeni Tennison, Gregg Kellogg, and Ivan Herman. Metadata Vocabulary for Tabular Data. https://www.w3.org/TR/2015/REC-tabular-metadata-20151217/, December 2015. W3C Recommendation.
[12] Alan Tygel, Sören Auer, Jeremy Debattista, Fabrizio Orlandi, and Maria Luiza Machado Campos. 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, 2016.
[13] Herbert Van de Sompel, Michael Nelson, and Robert Sanderson. HTTP framework for time-based access to resource states – Memento, 2013. RFC 7089.
[14] Petros Venetis, Alon Y. Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. Recovering semantics of tables on the web. *PVLDB*, 4(9):528–538, 2011.
[15] Ziqi Zhang. Effective and efficient semantic table interpretation using tableminer+. *Semantic Web*, (Preprint):1–37, 2016.