

# An Extensible Framework to Validate and Build Dataset Profiles

Ahmad Assaf<sup>1,2</sup>, Aline Senart<sup>2</sup> and Raphaël Troncy<sup>1</sup>

<sup>1</sup> EURECOM, Sophia Antipolis, France. <firstName.lastName@eurecom.fr>

<sup>2</sup> SAP Labs France. <firstName.lastName@sap.com>

**Abstract.** Linked Open Data (LOD) has emerged as one of the largest collection of interlinked datasets on the web. In order to benefit from this mine of data, one needs to access to descriptive information about each dataset (or metadata). Such metadata information is currently very limited to a few data portals where they are usually provided manually, thus being often incomplete and inconsistent in terms of quality. To address this issue, we propose a scalable automatic approach for extracting, validating, correcting and generating descriptive linked dataset profiles. This approach applies several techniques in order to check the validity of the metadata provided and to generate descriptive and statistical information for a particular dataset or for an entire data portal. We target CKAN powered data portals and validate our framework on the Linked Open Data (LOD) cloud. The results demonstrate that the general state of LOD cloud needs more attention as most of the datasets suffer from bad quality metadata lacking some informative metrics needed to facilitate dataset search. The noisiest metadata values were access information such as licensing information, resource descriptions as well as resource reachability problems.

**Keywords:** Linked Data, Dataset Profile, Metadata, Data Quality

## 1 Introduction

From 12 datasets cataloged in 2007, the Linked Open Data cloud has grown to nearly 1000 datasets containing more than 82 billion triples<sup>3</sup> [3]. Data is being published by both the public and private sectors and covers a diverse set of domains from life sciences to media or government data. The Linked Open Data cloud is potentially a gold mine for organizations and individuals who are trying to leverage external data sources in order to produce more informed business decisions [8]. This success lies in the cooperation between data publishers and consumers. Consumers are empowered to find, share and combine information in their applications easily. However, the heterogeneous nature of data sources reflects directly on the data quality as these sources often contain inconsistent as well as misinterpreted and incomplete metadata information. Considering the significant variation in size, the languages used and the freshness of the data,

---

<sup>3</sup> <http://datahub.io/dataset?tags=lod>

one realizes that finding useful datasets without prior knowledge is increasingly complicated. This can be clearly noticed in the LOD Cloud where few datasets such as DBPedia [4], Freebase [7] and YAGO [26] are favored over less popular datasets that may include domain specific knowledge more suitable for the tasks at hand. For example, for the task of building context-aware recommender systems in an academic digital library over LOD cloud, popular datasets like Semantic Web Dog Food, DBLP or Yovisto can be favored over lesser known but more specific datasets like VIAF<sup>4</sup> which links authority files of 20 national libraries, list of subject headings for public libraries in Spain<sup>5</sup> or the French dissertation search engine<sup>6</sup>.

The main entry point for discovering and identifying datasets is either through public data portals such as DataHub<sup>7</sup> and Europe's Public Data<sup>8</sup> or private search engines such as Quandl<sup>9</sup> and Engima<sup>10</sup>. Private portals harness manually curated data from various sources and expose them to users either freely or through paid plans. The data available is of higher quality but lesser quantity compared to what is available in public portals. Similarly, in some public data portals, administrators manually review datasets information, validate, correct and attach suitable metadata information. This information is mainly in the form of predefined tags such as *media*, *geography*, *life sciences* for organization and clustering purposes. However, the diversity of those datasets makes it harder to classify them in a fixed number of predefined tags that can be subjectively assigned without capturing the essence and breadth of the dataset [19]. Furthermore, the increasing number of datasets available makes the metadata review and curation process unsustainable even when outsourced to communities.

*Data profiling* is the process of creating descriptive information and collect statistics about that data. It is a cardinal activity when facing an unfamiliar dataset [20]. It helps in assessing the importance of the dataset, in improving users' ability to search and reuse part of the dataset and in detecting irregularities to improve its quality. Data profiling includes typically several tasks:

- **Metadata profiling:** Provides general information on the dataset (dataset description, release and update dates), legal information (license information, openness), practical information (access points, data dumps), etc.
- **Statistical profiling:** Provides statistical information about data types and patterns in the dataset, i.e. properties distribution, number of entities and RDF triples, etc.
- **Topical profiling:** Provides descriptive knowledge on the dataset content and structure. This can be in form of tags and categories used to facilitate search and reuse.

<sup>4</sup> <http://datahub.io/dataset/viaf>

<sup>5</sup> <http://datahub.io/dataset/lista-encabezamientos-materia>

<sup>6</sup> <http://datahub.io/dataset/thesesfr>

<sup>7</sup> <http://datahub.io>

<sup>8</sup> <http://publicdata.eu>

<sup>9</sup> <https://quandl.com/>

<sup>10</sup> <http://enigma.io/>

In this work, we address the challenges of automatic validation and generation of descriptive datasets profiles. This paper proposes an extensible framework consisting of a processing pipeline that combines techniques for data portals identification, datasets crawling and a set of pluggable modules combining several profiling tasks. The framework validates the provided dataset metadata against an aggregated standard set of information. Metadata fields are automatically corrected when possible, e.g. adding a missing license URL reference. Moreover, a report describing all the issues highlighting those that cannot be automatically fixed is created to be sent by email to the dataset’s maintainer. There exist various statistical and topical profiling tools for both relational and Linked Data. The architecture of the framework allows to easily add them as additional profiling tasks. However, in this paper, we focus on the task of dataset metadata profiling and present our findings by running our framework on the LOD cloud. The results demonstrate that the general state of LOD cloud needs more attention as most of the datasets suffer from bad quality metadata lacking some informative metrics needed to facilitate dataset search. The noisiest metadata are the access information such as licensing information, resource descriptions as well as resource availability problems.

The remainder of the paper is structured as follows. In Section 2, we review relevant related work. In Section 3, we describe our proposed framework’s architecture and components that validate and generate dataset profiles. In Section 4, we present the results when running this tool on the LOD cloud and we summarize the main issues found. Finally, we conclude and outline some future work in Section 5.

## 2 Related Work

There exists a considerable amount of tools that tackle specific profiling tasks. For example, [1][23] focus on generating statistical dataset information where in [19][15] authors use various techniques to attach additional topical information. However, to the best of our knowledge, this is the first effort towards extensible automatic validation and generation of descriptive dataset profiles. For this paper, we will focus on Linked Data metadata profiling tasks. However, one of the advantages of this framework is the ability to easily configure additional profiling tasks e.g. statistical or topical and accommodate different data types e.g. relational.

Data Catalog Vocabulary (DCAT) [14] and the Vocabulary of Interlinked Datasets (VoID) [10] are concerned with metadata about RDF datasets. There exist several tools aiming at exposing dataset metadata using these vocabularies. In [6] authors generate VoID descriptions limited to a subset of properties that can be automatically deduced from resources within the dataset. However, it still provides data consumers with interesting insights. Quality Assessment of Data Sources (Flemming’s Data Quality Assessment Tool)<sup>11</sup> provides basic

<sup>11</sup> <http://linkeddata.informatik.hu-berlin.de/LDSrcAss/datenquelle.php>

metadata assessment as it calculates data quality scores based on manual user input. The user assigns weights to the predefined quality metrics and answer a series of questions regarding the dataset. These include, for example, the use of obsolete classes and properties by defining the number of described entities that are assigned disjoint classes, the usage of stable URIs and whether the publisher provides a mailing list for the dataset. The ODI certificate<sup>12</sup> on the other hand provides a description of the published data quality in plain English. It aspires to act as a mark of approval that helps publishers understand how to publish good open data and users how to use it. It gives publishers the ability to provide assurance and support on their data while encouraging further improvements through an ascending scale. ODI comes as an online and free questionnaire for data publishers focusing on certain characteristics about their data. Although these approaches try to perform metadata profiling, they are either incomplete or manual. In our framework, we propose a more automatized and complete approach.

The Project Open Data Dashboard<sup>13</sup> tracks and measures how US government websites implement the Open Data principles to understand the progress and current status of their public data listings. A validator analyzes machine readable files e.g. JSON files for automated metrics like the resolved URLs, HTTP status and content-type. However, deep schema information about the metadata is missing like description, license information or tags. Similarly on the LOD cloud, the Data Hub LOD Validator<sup>14</sup> gives an overview of Linked Data sources cataloged on the Data Hub. It offers a step-by-step validator guidance to check a dataset completeness level for inclusion in the LOD cloud. The results are divided into four different compliance levels from basic to reviewed and included in the LOD cloud. Although it is an excellent tool to monitor LOD compliance, it still lacks the ability to give detailed insights about the completeness of the metadata and overview on the state of the whole LOD cloud group and is very specific to the LOD cloud group rules and regulations.

Although the above mentioned tools are able to provide various information about a dataset, there exist no approach that is extensible to combine further information coming from various profiling tools.

Dataset search can be done without relying on attached metadata (tags and categories). For example, there exist several approaches to create LOD indexes. In [2], authors used VoID descriptions to optimize query processing by determining relevant query-able datasets. In [17], authors created an approximate index structure (QTree) and an algorithm for answering conjunctive queries over Linked Data. SchemEX [18] is a stream-based approach leveraging type and property information of RDF instances to create schema-level indexes.

Semantic search engines like Sindice [12], Swoogle [13] and Watson [11] help in entities lookup but are not designed specifically for dataset search. In [24], authors utilized the sig.ma index [27] to identify appropriate data sources for

<sup>12</sup> <https://certificates.theodi.org/>

<sup>13</sup> <http://labs.data.gov/dashboard/>

<sup>14</sup> <http://validator.lod-cloud.net/>

interlinking. However, the current main source for dataset search and discovery is via data portals. CKAN and DKAN powered data portals rely on attached metadata to provide dataset search features as they run a Solr index on the metadata schemas. Having missing or inconsistent information will affect the search results quality.

### 3 Profiling Data Portals

In this section, we provide an overview of the processing steps for validating and generating dataset profiles. Figure 1 shows the main steps which are the following: (i) Data portal identification; (ii) metadata extraction; (iii) instance and resource extraction; (iv) profile validation (v) profile and report generation.

Our framework is built as a Command Line Interface (CLI) application using Node.js. Instructions on installing and running the framework are available on its public Github repository<sup>15</sup>. Related functions are encapsulated into modules that can be easily plugged in/out the processing pipeline. The various steps are explained in details below.

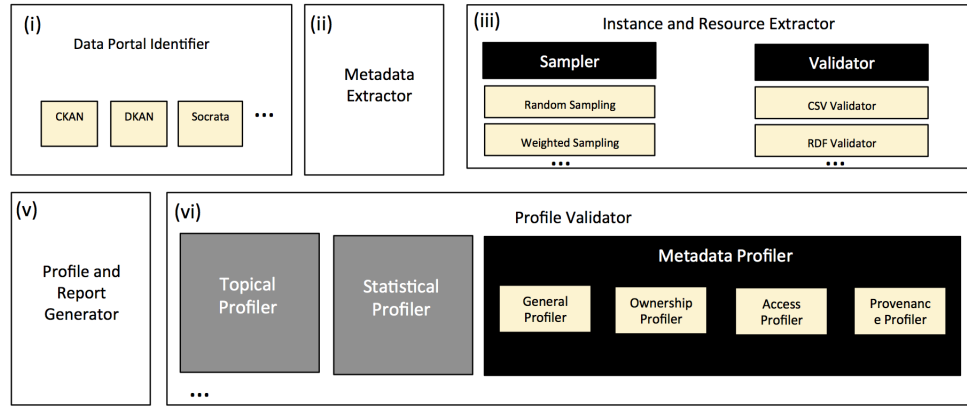


Fig. 1: Processing pipeline for validating and generating dataset profiles

#### 3.1 Data Portal Identification

Data portals can be considered as data access points providing tools to facilitate data publishing, sharing, searching and visualization. CKAN<sup>16</sup> is the world's leading open-source data portal platform powering websites like the DataHub, Europe's Public Data and the U.S Government's open data. Modeled on CKAN, DKAN<sup>17</sup> is a standalone Drupal distribution that is used in various public data

<sup>15</sup> <https://github.com/ahmadassaf/opendata-checker>

<sup>16</sup> <http://ckan.org>

<sup>17</sup> <http://drupal.org/project/dkan>

portals as well. Socrata<sup>18</sup> helps public sector organizations improve data-driven decision making by providing a set of solutions including an open data portal. In addition to these tradition data portals, there is a set of tools that allow exposing data directly as RESTful APIs like Datatank<sup>19</sup> and Database-to-API<sup>20</sup>.

Identifying the software powering data portals is a vital first step to understand the API calls structure. Web scraping is a technique for extracting data from Web pages. We rely on several scraping techniques in the identification process which includes a combination of the following:

- **URL inspection:** Check the existence of certain URL patterns. Various CKAN based portals are hosted on subdomains of the `http://ckan.net`. For example, CKAN Brazil (`http://br.ckan.net`).
- **Meta tags inspection:** The `<meta>` tag provides metadata about the HTML document. They are used to specify page description, keywords, author, etc. Inspecting the `content` attribute can indicate the type of the data portal. We use CSS selectors to check the existence of these meta tags. An example of a query selector is `meta[content*="ckan"]` (all meta tags with the attribute content containing the string *CKAN*). This selector can identify CKAN portals whereas the `meta[content*="Drupal"]` can identify DKAN portals.
- **Document Object Model (DOM) inspection:** Similar to the meta tags inspection, we check the existence of certain DOM elements or properties. For example, CKAN powered portals will have DOM elements with class names like `ckan-icon` or `ckan-footer-logo`. A CSS selector like `.ckan-icon` will be able to check if a DOM element with the class name `ckan-icon` exists. The list of elements and properties to inspect is stored in a separate configurable object for each portal. This allows the addition and removal of elements as deemed necessary.

The identification process for each portal can be easily customized by overriding the default function. Moreover, adding or removing steps from the identification process can be easily configured.

After those preliminary checks, we query one of the portal's API endpoints. For example, DataHub is identified as CKAN, so we will query the API endpoint on `http://datahub.io/api/action/package_list`. A successful request will list the names of the site's datasets, whereas a failing request will signal a possible failure of the identification process.

### 3.2 Metadata Extraction

Data portals expose a set of information about each dataset as metadata. The model used varies across portals. However, a standard model should contain

<sup>18</sup> <http://www.socrata.com>

<sup>19</sup> <http://thedataatank.com>

<sup>20</sup> <https://github.com/project-open-data/db-to-api>

information about the dataset’s title, description, maintainer email, update and creation date, etc. We divided the metadata information into the following:

**General information:** General information about the dataset. e.g. title, description, ID, etc. This general information is manually filled by the dataset owner. In addition to that, tags and group information is required for classification and enhancing dataset discoverability. This information can be entered manually or inferred modules plugged into the topical profiler.

**Access information:** Information about accessing and using the dataset. This includes the dataset URL, license information i.e. license title and URL and information about the dataset’s resources. Each resource has as well a set of attached metadata e.g. resource name, URL, format, size, etc.

**Ownership information:** Information about the ownership of the dataset. e.g. organization details, maintainer details, author, etc. The existence of this information is important to identify the authority on which the generated report and the newly corrected profile will be sent to.

**Provenance information:** Temporal and historical information on the dataset and its resources. For example, creation and update dates, version information, version, etc. Most of this information can be automatically filled and tracked.

Building a standard metadata model is not the scope of this paper, and since we focus on CKAN-based portals, we validate the extracted metadata against the CKAN standard model<sup>21</sup>.

After identifying the underlying portal software, we perform iterative queries to the API in order to fetch datasets metadata and persist them in a file-based cache system. Depending on the portal software we can issue specific extraction jobs. For example, in CKAN based portals, we are able to crawl and extract the metadata of a specific dataset, all the datasets in a specific group e.g. LOD Cloud or all the datasets in the portal.

### 3.3 Instance and Resource Extraction

From the extracted metadata we are able to identify all the resources associated with that dataset. They can have various types like a SPARQL endpoint, API, file, visualization ,etc. However, before extracting the resource instance(s) we perform the following steps:

- **Resource metadata validation and enrichment:** Check the resource attached metadata values. Similar to the dataset metadata, each resource should include information about its mimetype, name, description, format, valid de-referenceable URL, size, type and provenance. The validation process issue an HTTP request to the resource and automatically fills up various missing information when possible, like the mimetype and size by extracting them from the HTTP response header. However, missing fields like name and description that needs manual input are marked as missing and will appear in the generated summary report.

<sup>21</sup> [http://demo.ckan.org/api/3/action/package\\_show?id=adur\\_district\\_spending](http://demo.ckan.org/api/3/action/package_show?id=adur_district_spending)

- **Format validation:** Validate specific resource formats against a linter or a validator. For example, `node-csv`<sup>22</sup> for CSV files and `n3`<sup>23</sup> to validate N3 and Turtle RDF serializations.

Considering that certain dataset contains large amounts of resources and the limited computation power of some machines on which the framework might run on, a sampler module is introduced to execute various sample-based strategies detailed in [15] where they were found to generate accurate results even with comparably small sample size of 10%.

### 3.4 Profile Validation

A dataset profile should include descriptive information about the data examined. In our framework, we have identified three main profiling information. However, the extensibility of our framework allows for additional profiling techniques to be plugged in easily i.e. a quality profiling module reflecting the dataset quality. In this paper, we focus on the task of metadata profiling.

Metadata validation process identifies missing information and the ability to automatically correct them. Each set of metadata (general, access, ownership and provenance) is validated and corrected automatically when possible. Each profiler task has a set of metadata fields to check against. The validation process check if each field is defined and if the value assigned is valid.

There exist a bunch of special validation steps for various fields. For example, for ownership information where the maintainer email has to be defined, the validator checks if the email field is an actual valid email address. A similar process is done to URLs whenever found. However, we also issue an `HTTP HEAD` request in order to check if that URL is reachable or not. For the dataset resources, we use the `content-header` information when the request is successful in order to extract, compare and correct further metadata values like `mimetype` and `content size`.

Despite the legal issues surrounding Linked Data licenses [25], it is still considered a gold mine for organizations who are trying to leverage external data sources in order to produce more informed business decisions [8]. In [21] the authors see the potential economic effect unfolding in education, transportation, consumer products, electricity, oil and gas, health care and consumer finance. They estimate the potential annual value enabled by Open Data in these domains to be 3 trillion US Dollars across seven domains. As a result, validating license related information is vital. However, from our experiments, we found out that datasets' license information is noisy. The license names if found are not standardized. For example, Creative Commons CCZero can be also CC0 or CCZero. Moreover, the license URI if found and if de-referenceable can point to different reference knowledge bases e.g. `http://opendefinition.org`. To overcome this issue, we have manually created a mapping file standardizing the set

<sup>22</sup> <https://github.com/wdavidw/node-csv>

<sup>23</sup> <https://github.com/RubenVerborgh/N3.js>



of possible license names and the reference knowledge base<sup>24</sup>. In addition, we have also used the open source and knowledge license information<sup>25</sup> to normalize the license information and add extra metadata like the domain, maintainer and open data conformance.

---

```
{
  "license_id" : ["ODC-PDDL-1.0"],
  "disambiguations" : ["Open Data Commons Public Domain Dedication and License (PDDL)"]
},
{
  "license_id" : ["CC-BY-SA-4.0", "CC-BY-SA-3.0"],
  "disambiguations" : ["cc-by-sa", "CC BY-SA", "Creative Commons Attribution Share-Alike"]
}
```

---

Listing 1.1: License mapping file sample

### 3.5 Profile and Report Generation

The validation process highlights the missing information and presents them in a human readable report. The report can be automatically sent to the dataset maintainer email if exists in the metadata.

In addition to the generated report, the enhanced profiles are represented in JSON using the CKAN data model and are publicly available<sup>26</sup>.

Data portal administrators need an overall knowledge of the portal datasets and their properties. Our framework has the ability to generate numerous reports of all the datasets by passing formatted queries. There are two main set of aggregation tasks that can be run:

- **Aggregating meta-field values:** Passing a string that corresponds to a valid field in the metadata. The field can be flat like `license_title` (aggregates all the license titles used in the portal or in a specific group) or nested like `resource>resource_type` (aggregates all the resources types for all the datasets). Such reports are important to have an overview of the possible values used for each metadata field.
- **Aggregating key:object meta-field values:** Passing two meta-field values separated by a colon : e.g. `resources>resource_type:resources>name`. These reports are important as you can aggregate the information needed when also having the set of values associated to it printed.

For example, the meta-field value query `resource>resource_type` run against the LODCloud group will result in an array containing `[file, api, documentation...]` values. These are all the resource types used to describe all the datasets of the group. However, to be able to know also what are the datasets containing resources corresponding to each type, we issue a key:object meta-field query `resource>resource_type:name`. The result will be a JSON object having the `resource_type` as the key and an array of corresponding datasets titles that has a resource of that type.

<sup>24</sup> <https://github.com/ahmadassaf/opendata-checker/blob/master/util/licenseMappings.json>

<sup>25</sup> <https://github.com/okfn/licenses>

<sup>26</sup> <https://github.com/ahmadassaf/opendata-checker/tree/master/results>

Metadata Report
group information is missing. Check organization information as they can be mixed sometimes organization_image_url field exists but there is no value defined
Tag Statistics
There is a total of: 21 [undefined] vocabulary_id fields 100.00%
License Report
License information has been normalized !
Resource Statistics
There is a total of: 10 [missing] url-type fields 100.00%
There is a total of: 9 [missing] created fields 90.00%
There is a total of: 10 [undefined] cache_last_updated fields 100.00%
There is a total of: 10 [undefined] webstore_last_updated fields 100.00%
There is a total of: 10 [undefined] size fields 100.00%
There is a total of: 10 [undefined] hash fields 100.00%
There is a total of: 10 [undefined] mimetype_inner fields 100.00%
There is a total of: 7 [undefined] mimetype fields 70.00%
There is a total of: 10 [undefined] cache_url fields 100.00%
There is a total of: 6 [undefined] name fields 60.00%
There is a total of: 9 [undefined] webstore_url fields 90.00%
There is a total of: 9 [undefined] last_modified fields 90.00%
There is one [undefined] format field 10.00%
Resource Connectivity Issues
There are 2 connectivity issues with the following URLs: – <a href="http://dbpedia.org/void/Dataset">http://dbpedia.org/void/Dataset</a>
Un-Reachable URLs Types
There are: 1 unreachable URLs of type [file]

Listing 1.2: Excerpt of the DBpedia validation report

## 4 Experiments and Evaluation

In this section, we provide the experiments and evaluation of the proposed framework. All the experiments are reproducible by our tool and their results are available on the its Github repository.

We have run the framework on the LOD cloud containing 259 datasets at the time of writing this paper. We ran the instance and resource extractor in order to cache the metadata files for these datasets locally and ran the validation process which took around one and a half hour on a 2.6 Ghz Intel Core i7 processor with 16GB of DDR3 memory machine.

A CKAN dataset metadata describes three main sections in addition to the core dataset’s properties. Those are the groups, tags and resources. Each section contains a set of metadata corresponding to one or more metadata type. For example, a dataset resource will have general information such as the resource name, access information such as the resource url and provenance information

such as creation date. The framework generates a report aggregating all the problems in all these sections, fixing field values when possible. Errors can be the result of missing metadata fields, undefined field values or field value errors e.g. unreachable URL or incorrect email address.

Figures 2 and 3 show the percentage of errors found in metadata fields by section and by information type respectively. We found out that the most erroneous information for the dataset core information were ownership related as 41% were missing or undefined. Datasets resources have the poorest metadata. 64% of the general metadata, all the access information and 80% of the provenance information contained missing or undefined values. Table 1 shows the top metadata fields errors in each metadata information type.

Metadata Field		Error %	Section	Error Type	Auto Fix
General	group	100%	Dataset	Missing	-
	vocabulary_id	100%	Tag	Undefined	-
	url-type	96.82%	Resource	Missing	-
	mimetype_inner	95.88%	Resource	Undefined	Yes
	hash	95.51%	Resource	Undefined	Yes
	size	81.55%	Resource	Undefined	Yes
Access	cache_url	96.9%	Resource	Undefined	-
	webstore_url	91.29%	Resource	Undefined	-
	license_url	54.44%	Dataset	Missing	Yes
	url	30.89%	Resource	Unreachable	-
	license_title	16.6%	Dataset	Undefined	Yes
Provenance	cache_last_updated	96.91%	Resource	Undefined	Yes
	webstore_last_updated	95.88%	Resource	Undefined	Yes
	created	86.8%	Resource	Missing	Yes
	last_modified	79.87%	Resource	Undefined	Yes
	version	60.23%	Dataset	Undefined	-
Ownership	maintainer_email	55.21%	Dataset	Undefined	-
	maintainer	51.35%	Dataset	Undefined	-
	author_email	15.06%	Dataset	Undefined	-
	organization_image_url	10.81%	Dataset	Undefined	-
	author	2.32%	Dataset	Undefined	-

Table 1: Top metadata fields error % by type

We notice that 42.85% of the top metadata problems can be fixed automatically. 44.44% of these problems can be fixed by our tool while the others need tools that are plugged into the data portal. We further present and discuss the results grouped by metadata information type below.

#### 4.1 General information

34 datasets (13.13%) did not have valid **notes** values. **tags** information for the datasets were complete except for the **vocabulary\_id** as it was missing from

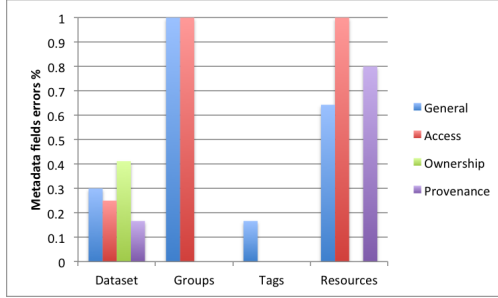


Fig. 2: Error % by section

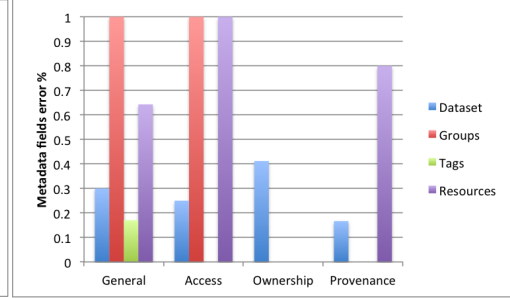


Fig. 3: Error % by information type

all the datasets' metadata. All the datasets **groups** information were missing `display_name`, `description`, `title`, `image.display_url`, `id`, `name`. After manual examination, we noticed a clear overlap between group and organization information. Many datasets like **event-media** used the organization field to show group related information (being in LOD Cloud) instead of the publishers details.

#### 4.2 Access information

25% of the datasets access information (being the dataset URL and any URL defined in its groups) has issues related to them (missing or unreachable URLs). Three datasets (1.15%) did not have a URL defined (**tip**, **uniprotdatabases**, **uniprotcitations**) while 45 datasets (17.3%) defined URLs were not accessible at the time writing this paper. One dataset did not have resources information (**bio2rdfchebi**) while the other datasets had a total of 1068 defined resources.

On the datasets resources level, we noticed wrong or inconsistent values in the **size** and **mimetype** fields. 20 (1.87%) resources had incorrect **mimetype** defined, while 52 (4.82%) had incorrect **size** values. These values have been automatically fixed based on the values defined in the HTTP response header. However, 44 datasets have valid **size** field values and 54 have valid **mimetype** field values where they were not reachable, thus providing incorrect information.

15 (68%) fields of all the other access metadata are missing or have undefined values. Looking closely, we noticed that most of these problems can be easily fixed automatically by tools that can be plugged to the data portal. For example, the top six missing fields are the `cache_last_updated`, `cache_url`, `urltype`, `webstore_last_updated`, `mimetype_inner` and `hash` which can be computed and filled automatically. However, the most important missing information which require manual entry are the dataset's **name** and **description** were missing from 817 (76.49%) and 98 (9.17%) resources respectively. A total of 334 resources (31.27%) URLs were not reachable, thus affecting highly the availability of these datasets. CKAN resources can be of various predefined types (*file*, *file.upload*, *api*, *visualization*, *codeanddocumentation*). The framework also breaks down these unreachable resources according to their types.

211 (63.17%) resources did not have valid `resource_type`, 112 (33.53%) were files, 8 (2.39%) and one (0.029%) metadata, example and documentation types.

To have more details about the resources URL types, we created a *key : objectmeta—fieldvalues* group level report on LOD cloud with `resources>format:title`. This will aggregate the resources format information for each dataset. We found out that only 161 (62.16%) of the datasets valid URLs have SPARQL endpoints defined by `api/sparql` resource format. 92.27% provided RDF example links and 56.3% provided direct links to RDF down-loadable dumps.

The noisiest part of the access metadata was license information. A total of 43 datasets (16.6%) did not have a defined `license_title` and `license_id` fields, where 141 (54.44%) had missing `license_url` field. However, we managed to normalize 123 (47.49%) of the datasets' license information using the manual mapping file.

### 4.3 Ownership information

Ownership information is divided into direct ownership (author and maintainer) and organization information. Four fields (66.66%) of the direct ownership information were missing or undefined. The breakdown for the missing information is: 55.21% `maintainer_email`, 51.35% `maintainer`, 15.06% `author_email`, 2.32% `author`. Moreover, our framework performs checks to validate existing email values. 11 (0.05%) and 6 (0.05%) of the defined `author_email` and `maintainer_email` fields were not valid email addresses respectively.

For the organization information, two field values (16.6%) were missing or undefined. 1.16% of the `organization_description` and 10.81% of the `organization_image_url` information with two out of these URLs were unreachable.

### 4.4 Provenance information

80% of the resources provenance information were missing or undefined. However, most of the provenance information e.g. `metadata_created`, `metadata_modified` can be computed automatically by tools plugged into the data portal. The only field requiring manual entry is the `version` field which was found to be missing from 60.23% of the datasets.

## 5 Conclusion and Future Work

In this paper, we proposed a scalable automatic approach for extracting, validating, correcting and generating descriptive linked dataset profiles. This approach applies several techniques in order to check the validity of the metadata provided and to generate descriptive and statistical information for a particular dataset or for an entire data portal. Based on our experiments running the tool on the LOD cloud, we discovered that the general state of the datasets needs attention as most of them lack informative access information and their resources suffer low availability. These two metrics are of high importance for enterprises looking

to integrate and use external linked data.

It has been noticed that the issues surrounding metadata quality affect directly dataset search as data portals rely on such information to power their search index. We noted the need for tools that are able to identify various issues in this metadata and correct them automatically. We found out that 32.25% of all the metadata information can be automatically fixed, on which 50% of them can be directly fixed by our framework. The rest are mainly provenance information that requires special treatment.

As part of our future work, we plan to introduce workflows that will be able to correct the rest of the metadata either automatically or through intuitive manually-driven interfaces. We also plan to integrate statistical and topical profilers to be able to generate full comprehensive profiles. We also intend to suggest a ranked standard metadata model that will help generate more accurate and scored metadata quality profiles. We also plan to run this tool on various CKAN based data portals, schedule periodic reports to monitor the evolution of datasets metadata. Finally, at some stage, we plan to extend this tool for other data portal types like DKAN and Socrata.

## References

1. Z. Abedjan, T. Gruetze, A. Jentzsch, and F. Naumann. Profiling and mining rdf data with `prolog++`. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 1198–1201, March 2014.
2. K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets - on the design and usage of void, the 'vocabulary of interlinked datasets'. In *WWW 2009 Workshop: Linked Data on the Web (LDOW2009)*, Madrid, Spain.
3. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
4. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, Sept. 2009.
5. C. Böhm, G. Kasneci, and F. Naumann. Latent topics in graph-structured data. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 2663–2666, 2012.
6. C. Böhm, J. Lorey, and F. Naumann. Creating void descriptions for web-scale data. *Web Semant.*, 9(3):339–345, Sept. 2011.
7. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08*, pages 1247–1250, 2008.
8. D. Boyd and K. Crawford. Six provocations for big data. *A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society*, pages 1–17, 2011.
9. M. Brümmer, C. Baron, I. Ermilov, M. Freudenberg, D. Kontokostas, and S. Hellmann. DataID: Towards semantically rich metadata for complex datasets. In *Proceedings of the 10th International Conference on Semantic Systems*, 2014.
10. R. Cyganiak, J. Zhao, M. Hausenblas, and K. Alexander. Describing linked datasets with the VoID vocabulary. W3C note, W3C, Mar. 2011. <http://www.w3.org/TR/2011/NOTE-void-20110303/>.

11. M. d'Aquin and E. Motta. Watson, more than a semantic web search engine. *Semant. web*, 2(1), Jan. 2011.
12. R. Delbru, N. Toupikov, and M. Catasta. Hierarchical link analysis for ranking web data. *The Semantic Web: Research and Applications*, 2010.
13. L. Ding, T. Finin, A. Joshi, R. Pan, and R. Cost. Swoogle: A semantic web search and metadata engine. *CIKM04*, 2004.
14. J. Erickson and F. Maali. Data catalog vocabulary (DCAT). W3C recommendation, W3C, Jan. 2014. <http://www.w3.org/TR/2014/REC-vocab-dcat-20140116/>.
15. B. Fetahu, S. Dietze, B. Pereira Nunes, M. Antonio Casanova, D. Taibi, and W. Nejdl. A scalable approach for efficiently generating structured dataset topic profiles. In *The Semantic Web: Trends and Challenges*, volume 8465 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014.
16. M. Frosterus, E. Hyvönen, and J. Laitio. Datafinland - a semantic portal for open and linked datasets. In *ESWC (2)'11*, pages 243–254, 2011.
17. A. Harth, K. Hose, M. Karnstedt, A. Polleres, K.-U. Sattler, and J. Umbrich. Data summaries for on-demand queries over linked data. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 411–420, 2010.
18. M. Konrath, T. Gottron, S. Staab, and A. Scherp. Schemex - efficient construction of a data catalogue by stream-based indexing of linked data. *Web Semant.*, 16:52–58, Nov. 2012.
19. S. Lalithsena, P. Hitzler, A. Sheth, and P. Jain. Automatic domain identification for linked open data. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 205–212, Nov 2013.
20. H. Li. Data profiling for semantic web data. In F. Wang, J. Lei, Z. Gong, and X. Luo, editors, *Web Information Systems and Mining*, volume 7529 of *Lecture Notes in Computer Science*, pages 472–479. Springer Berlin Heidelberg, 2012.
21. J. Manyika and E. A. Doshi. Open data: Unlocking innovation and performance with liquid information. Technical report, 2013.
22. P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems, I-Semantics '11*, 2011.
23. E. Mäkelä. Aether – generating and viewing extended void statistical descriptions of rdf datasets. In *Proceedings of the ESWC 2014 demo track, Springer-Verlag*, 2014.
24. A. Nikolov, M. d'Aquin, and E. Motta. What should i link to? identifying relevant sources and classes for data linking. In *The Semantic Web*, volume 7185 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012.
25. K. J. Prateek Jain, Pascal Hitzler and C. Venkatramani. There's no money in linked data. 2013.
26. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW '07*, pages 697–706, 2007.
27. G. Tummarello, S. Danielczyk, R. Cyganiak, R. Delbru, M. Catasta, E. P. Federale, and S. Decker. Sig.ma: Live views on the web of data. In *In Proc. WWW-2010*, pages 1301–1304. ACM Press, 2010.