Roomba: Automatic Validation, Correction and Generation of Dataset Metadata

Enhancing Dataset Search and Spam Detection

†‡ Ahmad Assaf, ‡ Aline Senart, and † Raphaël Troncy
†EURECOM, Sophia Antipolis, France
‡ SAP Labs, Sophia Antipolis, France
†raphael.troncy@eurecom.fr, ‡firstName.lastName@sap.com

ABSTRACT

Data is being published by both the public and private sectors and covers a diverse set of domains ranging from life sciences to media or government data. An example is the Linked Open Data (LOD) cloud which 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. Considering the significant variation in size, the languages used and the freshness of the data, one realizes that spotting spam datasets or simply finding useful datasets without prior knowledge is increasingly complicated. In this paper, we propose Roomba, a scalable automatic approach for extracting, validating, correcting and generating descriptive linked dataset profiles. While Roomba is generic, we target CKAN-based data portals and we validate our approach against a set of open data portals including the Linked Open Data (LOD) cloud as viewed on the DataHub. The results demonstrate that the general state of various datasets and groups, including the LOD cloud group, needs more attention as most of the datasets suffer from bad quality metadata and lack some informative metrics that are required to facilitate dataset search.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing

Keywords

Dataset Profile, Metadata, Data Quality, Linked Data

1. INTRODUCTION

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 [11]. It helps in assessing the importance of a dataset, in improving users' ability to search and reuse part of a dataset and in detecting irregularities to improve its quality. Data profiling includes typically several tasks:

- **metadata profiling** that provides general information on the dataset (dataset description, release and update dates), legal information (license information, openness), practical information (access points, data dumps);
- **statistical profiling** that provides statistical information about data types and patterns in the dataset, i.e. properties distribution, number of entities and RDF triples;
- **topical profiling** that provides descriptive knowledge on the dataset content and structure, e.g. tags and categories used to facilitate search and reuse.

The main entry point for discovering and identifying datasets is either through public data portals such as DataHub[1], the Europe's Public Data[2] or private data search engines such as Quandl[3], Engima[4]. Data on public portals is checked manually as administrators review, validate and correct datasets information and attach suitable metadata when available. The increasing number of datasets hinders the scalability of this process, affecting the correct and efficient spotting of datasets spam. CKAN-based data portals rely on metadata attached to datasets that enable search as they run a Solr index on the metadata schemas. Having missing or inconsistent information will affect the search results quality.

In this paper, we propose Roomba, a scalable automatic approach for extracting, validating, correcting and generating descriptive linked dataset profiles. Our framework consists of a processing pipeline that combines techniques for data portals identification, datasets crawling and a set of pluggable modules for the profiling tasks. Roomba validates dataset metadata against an aggregated standard set of information. Metadata fields are automatically corrected when possible, e.g. for adding a missing license URL. Moreover, a report describing all the issues and highlighting those that cannot be automatically fixed is created and 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 Roomba allows to easily

[1] `http://datahub.io`
[2] `http://publicdata.eu`
[3] `https://quandl.com/`
[4] `http://enigma.io/`

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742827.

add them as additional profiling modules. However, in this paper, we focus on the task of dataset metadata profiling and present our findings by running Roomba on the LOD cloud[5]. The results demonstrate that the general state of the LOD cloud needs more attention as most of the datasets suffer from bad quality metadata. The noisiest metadata are about the access information such as the license, the resource descriptions as well as problem related to resource availability.

2. RELATED WORK

Semantic sitemaps [4] and RDFStats [10] were one of the first to deal with RDF data statistics and summaries. However, there has been lately a plethora of tools like ExpLOD [8], LODStats [2], ProLOD++ [1], LODOP [6] and Aether [12] that compute some statistical information which is vital to applications dealing with query optimization and answering, data cleansing, schema induction and data mining [7, 9].

Topical and categorical information facilitates dataset search and reuse. Topical profiling focuses on content-wise analysis at the instances and ontological levels. In [9, 3, 5], the authors try to automatically infer information about the underlying dataset depending on external knowledge bases or the dataset's internal structure and ontological information.

The Open Data Dashboard project[6] tracks and measures how US government web sites 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 number of 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[7] gives an overview of Linked Data sources cataloged. 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 an overview on the state of the entire LOD cloud group while being very specific to the LOD cloud group rules and regulations.

3. FRAMEWORK ARCHITECTURE

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

3.1 System Overview

Roomba is built as a Command Line Interface (CLI) application using Node.js. Instructions on installing and running Roomba are available on its public Github repository[8] and explained in this short screencast[9]. Related functions are encapsulated into modules that can be easily plugged in/out the processing pipeline.

[5] `http://datahub.io/dataset?tags=lod`
[6] `http://labs.data.gov/dashboard/`
[7] `http://validator.lod-cloud.net/`
[8] `https://github.com/ahmadassaf/opendata-checker`
[9] `http://youtu.be/p7Y-mDN_Y2s`

3.2 Data Portal Identification

Data portals can be considered as data access points providing tools to facilitate data publishing, sharing, searching and visualization. CKAN is the world's leading open-source data portal platform powering web sites like the DataHub, Europe's Public Data and the U.S Government's open data. Modeled on CKAN, DKAN is a standalone Drupal distribution that is used in various public data portals as well. 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 sub-domains of `http://ckan.net`, for example, CKAN Brazil is at `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-based 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 at `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.

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

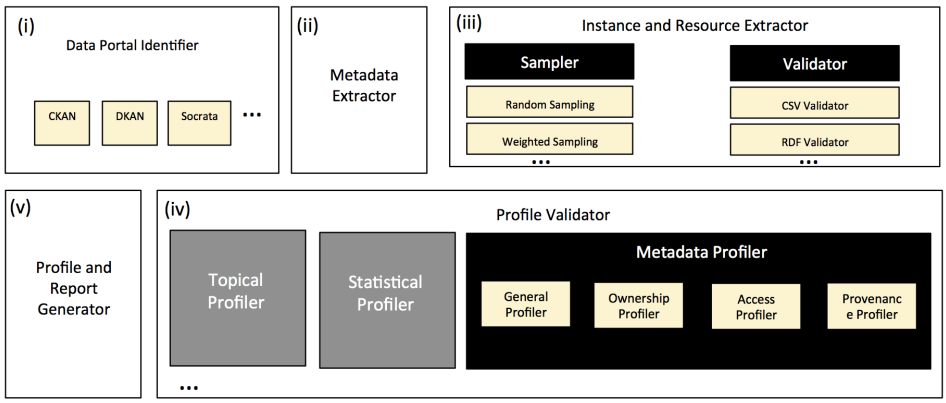

Figure 1: Processing pipeline for validating and generating dataset profiles

General information: General information about the dataset, e.g. title, description and ID. This general information is manually filled by the dataset owner. In addition to that, tags and group information are required for classification and enhancing dataset discoverability. This information can be entered manually or inferred using 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 also a set of attached metadata, e.g. resource name, URL, format, size.

Ownership information: Information about the ownership of the dataset, e.g. organization details, maintainer details, author. 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, e.g. creation and update dates, version information. Most of this information can be automatically filled up and tracked.

Although Roomba is generic and accepts any data model to check against, for this demo, we have used the CKAN standard model[10] as we do our experiments on the LOD cloud. 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, or all the datasets in a specific group (e.g. the LOD Cloud), or all the datasets in the portal.

3.4 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. The validation process issue an HTTP request to the resource and automatically fills up various missing information when possible, like the mime type and size by extracting them from the HTTP response header.

- **Format validation**: Validate specific resource formats against a linter or a validator.

Considering that some dataset contains large amounts of resources and the limited computation power of some machines on which Roomba might run on, a sampler module is introduced to execute various sample-based strategies discussed in [5] that were found to generate accurate results even with comparably small a sample size of 10%.

3.5 Profile Validation

The metadata validation process identifies missing information and is able 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 checks if each field is defined and if the value assigned is valid.

There are 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 and we issue an HTTP `HEAD` request in order to check whether a 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 the mime type and the content size.

From our experiments, we found out that datasets' license information is generally noisy. The license names, if found, are not standardized. For example, Creative Commons CC Zero can be CC0 or CCZero. Moreover, the license URI, if found and if de-referenceable, can point to different reference knowledge bases such as `http://opendefinition.org`. To overcome this issue, we have manually created a mapping file standardizing the set of possible license names and the reference knowledge base[11]. In addition, we have also used

[10] `http://demo.ckan.org/api/3/action/package_show?id=adur_district_spending`

[11] `https://github.com/ahmadassaf/opendata-checker/blob/master/util/licenseMappings.json`

the open source and knowledge license information[12] to normalize the license information and add additional metadata like the domain, maintainer and open data conformance.

3.6 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 provided 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[13]. Data portal administrators need an overall knowledge of the portal datasets and their properties. Roomba 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 one can aggregate the information needed when also having the set of values associated to it printed.

4. DEMONSTRATION AND CONCLUSION

During the demo, we will show how one can crawl a data portal, generate reports based on manual queries over the datasets metadata, validate a dataset profile and generate a new enriched profile with automatically fixed some problems. Moreover, users will be invited to try Roomba providing their own datasets hosted on any CKAN-powered portal and directly check the generated report.

Roomba is flexible and extensible. It can be plugged into data portals or used as a standalone tool to check for bad quality dataset metadata and identify possible spam. Automatically corrected profiles are of higher quality thus improving dataset search and retrieval. Further work includes introducing 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.

Acknowledgments

This research has been partially funded by the European Union's 7th Framework Programme via the project Apps4EU (GA No. 325090).

[12] `https://github.com/okfn/licenses`

[13] `https://github.com/ahmadassaf/opendata-checker/tree/master/results`

5. REFERENCES

[1] Z. Abedjan, T. Gruetze, A. Jentzsch, and F. Naumann. Profiling and mining RDF data with ProLOD++. In *30th IEEE International Conference on Data Engineering (ICDE)*, pages 1198–1201, 2014.

[2] S. Auer, J. Demter, M. Martin, and J. Lehmann. LODStats - an Extensible Framework for High-performance Dataset Analytics. In *18th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 353–362, Galway, Ireland, 2012.

[3] C. Böhm, G. Kasneci, and F. Naumann. Latent Topics in Graph-structured Data. In *21st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2663–2666, Maui, Hawaii, USA, 2012.

[4] R. Cyganiak, H. Stenzhorn, R. Delbru, S. Decker, and G. Tummarello. Semantic Sitemaps: Efficient and Flexible Access to Datasets on the Semantic Web. In *5th European Semantic Web Conference (ESWC)*, pages 690–704, Tenerife, Spain, 2008.

[5] 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 *11th European Semantic Web Conference (ESWC)*, 2014.

[6] B. Forchhammer, A. Jentzsch, and F. Naumann. LODOP - Multi-Query Optimization for Linked Data Profiling Queries. In *International Workshop on Dataset PROFIling and fEderated Search for Linked Data (PROFILES)*, Heraklion, Greece, 2014.

[7] A. Jentzsch. Profiling the Web of Data. In *13th International Semantic Web Conference (ISWC), Doctoral Consortium*, Trentino, Italy, 2014.

[8] S. Khatchadourian and M. P. Consens. ExpLOD: Summary-based Exploration of Interlinking and RDF Usage in the Linked Open Data Cloud. In *7th Extended Semantic Web Conference (ESWC)*, pages 272–287, Heraklion, Greece, 2010.

[9] S. Lalithsena, P. Hitzler, A. Sheth, and P. Jain. Automatic Domain Identification for Linked Open Data. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pages 205–212, 2013.

[10] A. Langegger and W. Woss. RDFStats - An Extensible RDF Statistics Generator and Library. In *20th International Workshop on Database and Expert Systems Application (DEXA)*, pages 79–83, 2009.

[11] H. Li. Data Profiling for Semantic Web Data. In *International Conference on Web Information Systems and Mining (WISM)*, pages 472–479, 2012.

[12] E. Mäkelä. Aether – Generating and Viewing Extended VoID Statistical Descriptions of RDF Datasets. In *11th European Semantic Web Conference (ESWC), Demo Track*, Heraklion, Greece, 2014.