GovLOD: Towards a Linked Open Data Portal

Octavian Rinciog and Vlad Posea

Politehnica University of Bucharest
{octavian.rinciog, vlad.posea}@cs.pub.ro

Abstract. Nowadays, governments and public agencies publish open data at an exponentially growing rate on dedicated portals. These open data sources provide semi-structured or unstructured data, because the focus is on publishing data and not on how they are later used. GovLOD is a platform that aims to transform the information found in these heterogeneous files in Linked Open Data using RDF triples.

Keywords: Linked Open Data, RDF, OCR, SPARQL

1 Introduction

Most developed countries governments have implemented public open data portals. The data for this portals are supplied by the governments and public institutions, helping citizens to find, download and use information generated and held by these public institutions. Portals that publish public data exist in countries like USA, UK or Germany. Romania started an open data portal in 2013. This portal now holds 441 datasets from 71 institutions, less than 2% of the portal content in UK for example.

The problem with these open data portals and implicitly with the data delivered by these is that, in most cases [2], provided files do not have any structure, being very hard to work with. The information found in these files can be classified as: a) *Semi-structured data:* Most published files are semi-structured, such as CSV or XLS. These files do not have a formal defined structure, so the users of public data must download each document they want to use and must understand how data are presented in that file. b) *Unstructured data:* Unfortunately, there are published files that do not even have a structure, such as scanned PDF files, doc or zip files, so these documents must be processed before being actually used.

The problems which developers are facing using these published data are:

Heterogeneous data The same information can be found in multiple formats on same data portal, being published by different institutions. This heterogeneity of used formats and structures prevents data consumption.

No data linking - There is not a clear link between the published data. For example, if a user wants to find all the information published about a city or about a particular industry he must download all files published and seek every possible file that could contain information about what is interesting to him.

Further, in this article we present GovLOD, a platform which aims to change the paradigm of open data portals by publishing the data as Linked Open Data with a well-defined structure. Thus, we propose a semi-automatic solution for ingestion, processing and publication of information that is hidden in files published on open data portals governments. By publishing the data found in files in Linked Open Data format, our platform also proposes a very easy method for developers to build applications without the need to process the initial files.

2 Platform

Our platform consists of several layers, each having a clearly defined role. Some of these layers are automatic, for example: ingestion or processing layer, others are currently not: vocabulary choosing. The purpose of the platform is to transform the open data files into Linked Open Data, according to requirements from [3].

2.1 Architecture

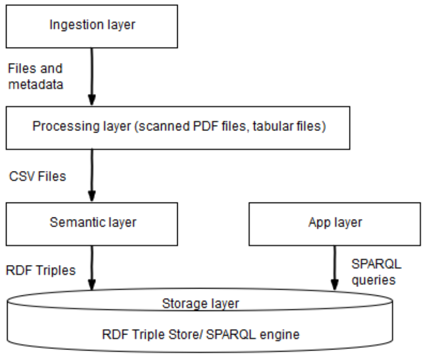

Fig. 1. GovLOD Architecture

The platform architecture is shown in Figure 1 and is structured according to the following workflow: First, files are taken from open data portals. The second step is processing these files in order to have a well-defined structure. In the third step the information from the files is converted to RDF triples and stored in a semantic repository. Developers who want to use public data can use SPARQL queries for accessing the data, without the need of processing the initial files.

Ingestion layer Most open data portals are implemented across CKAN platform that provides APIs for information retrieval from the portals. Our system connects to this API and processes the new information published since the last query. This process is an automatic one: this layer keeps track of all downloaded files and in the current step it only downloads the newly added files from the last query time.

Processing layer In the second step, the files retrieved pass through a stage of processing, whose purpose is to identify and define the structure of each file. Currently we have defined modules for processing PDF and tabular files, such as CSV or XLS. From the PDF files, whether they are scanned or not, our platform extracts the tables and makes them available as structured data. This process is based on techniques that identify the table structure within the pages of the file, identify the text by applying OCR on each of the table cells and afterwards export the tables as CSV files. Also in this step data cleaning and normalization is undergone by identifying the incorrectly formatted numbers and strings.

Semantic layer Data taken from CSV files returned in the previous stage are classified into two categories: a) statistical data b) physical entities information (e.g. monuments, hospitals, schools, churches and people). In this phase, existing data from files are converted to RDF triples. The URI schema is consistent, so each time when an entity is recognized, it will be mapped to the same URI. Files containing statistical data are converted using RDF Data Cube Vocabulary[1]. Thus we map every dimension of data in one semantic dimension of the cube. Because all statistical open data must be located in time and space, they must share these two common dimensions. The other dimensions are variable and the user must identify the properties to be mapped to each dimension. Also, we must identify an ontology in which we map each data about physical entities. At this time, in our system this mapping is done manually, but in the future we plan to have it done automatically or semi-automatically. If an existing ontology cannot be found on the given subject the user has the possibility to define new properties inside the platform. Also in this step, we augment existing data using several external web services, such as geolocation for physical entities. Then we link the created resources with existing resources from the Linked Open Data Cloud. Currently the system supports linking of resources through <owl:sameAs> property, with existing resources in dbpedia.org. Similar resources are identified by matching exactly the property <dbp:name> from dbpedia.org with <rdf:name> from our created resources. In the future we intend to integrate a framework of automatic retrieval of connections, such as Silk[5].

Storage layer RDF triples transformed in the above step are stored in Apache Marmotta platform which contains a SPARQL engine and a reasoning one[2]. For effective data storage Apache Marmotta can use a H2 or a PostgreSQL database. We use the last option as it provides additional scalability.

App layer As already mentioned, using the SPARQL engine from the previous layer, developers can implement applications only by running SPARQL queries, without having to process the initial files. The major advantage of this layer is that the data structure is documented and well-defined. Each data set that was transformed into RDF triples has attached one wiki page[3], which explains the properties of the RDF triples and gives a few SPARQL queries samples about how to access these data.

[1] https://www.w3.org/TR/vocab-data-cube

[2] http://opendata.cs.pub.ro/repo/sparql

[3] http://opendata.cs.pub.ro/wiki

3 Results

In order to test this platform, we used data taken from Romanian national public data portal[4]. As already mentioned, in this portal 441 datasets are published, comprising 3,231 files. From there, we selected a number of 117 documents, divided into 23 datasets. They share information in the following areas: health (data about hospitals and pharmacies), education (schools), culture (museums, archaeological sites and churches), and statistical data (types of cars registered in this country). From these files converted to RDF, our platform stored 165,279 resources and 2.4 million RDF triples.

The easy use of the data provided by this platform is evidenced by the application Romanian Linked Open Data Map[5], whose purpose is to display the nearest museums, hospitals or pharmacies to the user's location. This application was implemented using only data provided by our platform, using SPARQL queries.

4 Conclusion

In this article, we presented GovLOD, a solution that ingests and transforms files from national open data portals and publishes RDF triples into a semantic repository. The platform also provides a SPARQL engine that developers can use for implementing applications without the need to process the initial documents.

In recent years, a number of research projects aimed to transform open data in Linked Open Data. For example, City Data Pipeline [1] creates a single model under which it aggregates all available data about a given city and DataLift [4] is a tool that converts CSV files and relational databases taken from city open data portals into RDF triples.

Our platform differs from these solutions by automatically ingesting files that contain open data by transforming structured data taken from scanned PDF files and statistical data into RDF triples.

References

1. Bischof, S., Polleres, A., et al.: City data pipeline. Proc. of the I-SEMANTICS (2013)
2. Böhm, C., Freitag, M., Heise, A., et al.: Govwild: integrating open government data for transparency. In: Proceedings of the 21st International Conference on World Wide Web. pp. 321–324. ACM (2012)
3. Heath, T., Hausenblas, M., Bizer, C., Cyganiak, R., Hartig, O.: How to publish linked data on the web. In: Tutorial in the 7th International Semantic Web Conference, Karlsruhe, Germany (2008)
4. Scharffe, F., Atemezing, G., Troncy, R., et al.: Enabling linked-data publication with the datalift platform. In: Proc. AAAI workshop on semantic cities (2012)
5. Volz, J., Bizer, C., et al.: Discovering and maintaining links on the web of data. In: International Semantic Web Conference. pp. 650–665. Springer (2009)

[4] http://data.gov.ro

[5] http://opendata.cs.pub.ro:3000