Crowd-Sourcing the Large-Scale Semantic Mapping of Tabular Data

Ivan Ermilov
AKSW/BIS, Universität Leipzig
PO Box 100920, 04009
Leipzig, Germany
iermilov@informatik.uni-leipzig.de

Sören Auer
AKSW/BIS, Universität Leipzig
PO Box 100920, 04009
Leipzig, Germany
auer@informatik.uni-leipzig.de

Claus Stadler
AKSW/BIS, Universität Leipzig
PO Box 100920, 04009
Leipzig, Germany
cstadler@informatik.uni-leipzig.de

ABSTRACT

Governments and public administrations started recently to publish large amounts of structured data on the Web, mostly in the form of tabular data such as CSV files or Excel sheets. Various tools and projects have been launched aiming at facilitating the lifting of tabular data to reach semantically structured and linked data. However, none of these tools supported a truly incremental, pay-as-you-go data publication and mapping strategy, which enables effort sharing between data owners, community experts and consumers. In this article, we present an approach for enabling the crowd-sourcing of the large-scale semantic mapping of tabular data. We devise a simple mapping language for tabular data, which is easy to understand even for casual users, but expressive enough to cover the vast majority of potential tabular mappings use cases. Default mappings are automatically created and can be revised by the community using a semantic wiki. The mappings are executed using a sophisticated streaming RDB2RDF conversion. We report about the deployment of our approach at the Pan-European data portal PublicData.eu, where we transformed and enriched almost 10,000 datasets accounting for 7.3 billion triples.

Author Keywords

Tabular data, RDF, mapping, crowd-sourcing

ACM Classification Keywords

D.2.12 Interoperability: Data mapping; H.3.5 Online Information Services: Data sharing

INTRODUCTION

Integrating and analyzing large amounts of data plays an increasingly important role in todays society. Often, however, new discoveries and insights can only be attained by integrating information from dispersed sources. Despite recent advances in structured data publishing on the Web (such as RDFa and the schema.org initiative) the question arises how larger datasets can be published, described in order to make them easily discoverable and facilitate the integration as well as analysis.

One approach for addressing this problem are data portals, which enable organizations to upload and describe datasets using comprehensive metadata schemes. Similar to digital libraries, networks of such data catalogs can support the description, archiving and discovery of datasets on the Web. Recently, we have seen a rapid growth of data catalogs being made available on the Web. The data catalog registry *datacatalogs.org*, for example, lists already 285 data catalogs worldwide. Examples for the increasing popularity of data catalogs are Open Government Data portals, data portals of international organizations and NGOs as well as scientific data portals.

Governments and public administrations started to publish large amounts of structured data on the Web, mostly in the form of tabular data such as CSV files or Excel sheets. Examples are the data portals of the US[1], the UK[2] or the European Commission[3] as well as numerous other local, regional and national data portal initiatives.

The Semantic Web and Linked Data communities are advocating the use of RDF and Linked Data as a standardized data publication format facilitating data integration and visualization. Despite its unquestioned advantages, only a tiny fraction of open data is currently available as RDF. At the Pan-European data portal PublicData.eu, which aggregates dataset descriptions from numerous other European data portals, for example, only 459 out of more than 17.000 datasets (i.e. just 3%) are available as RDF. This can be mostly attributed to the fact, that publishing data as RDF requires additional effort in particular with regard to identifier creation, vocabulary design, reuse and mapping.

Various tools and projects have been launched aiming at facilitating the lifting of tabular data to reach semantically structured and interlinked data. Examples are *Any23*[4], *Tripli-*

WebSci 2012, June 22–24, 2012, Evanston, Illinois, USA.
Copyright 2012 ACM 978-1-4503-1228-8...$10.00.

[1] http://www.data.gov/
[2] http://data.gov.uk/
[3] http://open-data.europa.eu/
[4] http://any23.apache.org/

fy/Sparqlify [1], *Tabels*[5], *RDF Refine*[6]. However, none of these tools supported a truly incremental, pay-as-you-go data publication and mapping strategy, which enabled effort sharing between data owners and consumers. The lack of such an *architecture of participation* with regard to the mapping and transformation of tabular data to semantically richer representations hampers the creation of an ecosystem for open data publishing and reuse. In order to realize such an ecosystem, we have to enable a large number of potential stakeholders to effectively and efficiently collaborate in the data lifting process. Small contributions (such as fine-tuning of a mapping configuration or the mapping of an individual column) should be possible and render an instant benefit for the respective stakeholder. The sum of many such small contributions should result in a comprehensive Open Data knowledge space, where datasets are increasingly semantically structured and interlinked.

In this article, we present an approach for enabling the crowd-sourcing of the large-scale semantic mapping of tabular data. We formalize the canonical form of tabular data and survey possible deviations from this canonical form. We devise a mapping language for tabular data, which able to cope with typical deviations (e.g. repeated headers, empty rows, columns), is easy to understand even for casual users, but expressive enough to cover the vast majority of potential tabular mappings use cases. Our approach involves automatically creating default mappings for all datasets registered at PublicData.eu, which can be revised using a semantic wiki thus facilitating the crowd-sourcing. The mappings are interactively executed after mapping changes using our sophisticated *Sparqlify* streaming RDB2RDF conversion technique, which is able to transform large datasets with minimal resources. We report about the deployment of our approach at the Pan-European data portal PublicData.eu, where we transformed and enriched almost 10,000 datasets accounting for 7.3 billion triples.

The paper is structured as follows: In Section 1 we perform an inventory of the datasets being made available through local, regional and national data portals and being aggregated at PublicData.eu. In Section 2 we define a canonical model of tabular data and possible survey deviations from this model. Section 3 describes our transformation approach from tabular data to RDF. In Section 4 we describe the deployment at PublicData.eu and the crowd-sourcing of mappings. Section 5 summarizes the results of mapping and transforming almost 10,000 tabular datasets at PublicData.eu. We survey related work in Section 6 and conclude with an outlook on future work in Section 7.

PUBLICDATA.EU INVENTORY

PublicData.eu is a data catalog aiming to become a one stop shop for open-data in Europe. The rationale is to increase public access to high-value, machine-readable datasets generated by the European, national, regional as well as local governments and public administrations. This is achieved by harvesting and exposing datasets from various European data catalogs (currently 19 catalogs are harvested[7]). The communication with other data catalogs is performed by employing the *DCAT* vocabulary [8], an RDF vocabulary well-suited to represent information in data catalogs. PublicData.eu is, as well as many other data catalogs, based on the open-source platform CKAN[8].

Figure 1: PublicData.eu tag cloud.

CKAN exposes metadata about datasets in a catalog and allows to publish, share, find and use the registered datasets. Figure 4 shows a CKAN entry of a dataset at PublicData.eu. CKAN provides means for users and developers to easily access the published datasets. The registered datasets can be explored by end-users through free-text and faceted search based on various attributes, dataset groups and tagging. The *CKAN API* provides programmatic access to the metadata stored about datasets in a CKAN instance.

At PublicData.eu, dataset metadata is only available in read-only mode, not allowing users to modify the metadata (since it is harvested from other data catalogs). At the time of writing PublicData.eu comprises 17,027 datasets. These are categorized by categories, groups, license, geographical coverage and format. Comprehensive statistics gathered from the PublicData.eu are summarized in Table 1.

The information in Table 1 does not reflect all 17,027 datasets, because metadata is not available for all datasets. The good coverage of the UK with 7,798 datasets (45.80%) can be attributed to the comprehensive *data.gov.uk* Open Data portal, which contributed together with smaller UK data portals overall 9,099 datasets (53.44%) published under the UK OGL license. Metadata about categories (2,332 datasets – 13.70%) and groups (2,049 datasets – 12.03%) is much more sparse and insufficient to obtain a comprehensive picture of Open

[5] http://idi.fundacionctic.org/tabels

[6] http://refine.deri.ie/

[7] http://www.datacatalogs.org/dataset?groups=publicdata-eu

[8] http://ckan.org/

Categories	#	Groups	#	License	#	Coverage	#
Health and Social Care	568	Social	229	OGL	9,099	UK	7,798
		Health	83	N/A	6,489	N/A	951
Economy	277	Finance	436	CC-BY	541	UK Global (overseas)	555
		Economy	118	Other (Attribution)	349	Wiener Gemeindebezirk	6
People and Places	258	Geography	15	CC-Zero	200	Wien	4
		Culture	10	Other (Not Open)	128	Wiener Gemeindebezirke	2
Children, Education, Skills	220	Education	194	Other (Public Domain)	102		
Population	218	Population	145	ODbL	67		
Agriculture, Environment	162	Environment	227	CC BY-SA	27		
		Agriculture	181	Other (Open)	10		
Business and Energy	157	Services	66	Other (Non-Commercial)	6		
Crime and Justice	139			CC NC (Any)	5		
Travel and Transport	135	Transport	199	PDDL	4		
Government	101	Politics	69	GNU FDL	2		
Labour Market	97	Employment	77				

Table 1: PublicData.eu categories, groups, licenses and geographical coverage.

Format	#	Format	#
N/A	23,772	TXT	919
CSV	12,255	ZIP	697
Other tabular data	8,330	RDF	459
HTML	6,271	Geographical data	409
PDF	1,270	DOC	172
XML	1,132	Other	163

Table 2: Distribution of file formats at PublicData.eu.

Data at the European scale. However, tags are associated with 15,578 datasets (91.49%). There are overall 17,988 distinct tags, which are used 96,507 times. Figure 1 shows a tag cloud generated from all tags used more than 15 times.

Each dataset can comprise several data resources and there are overall 55,849 data resources available at PublicData.eu. Data resources can represent the same data in various formats, contain example data, schemata or linksets. Statistics on formats are summarized in Table 2.

A large part of the datasets at PublicData.eu are in tabular format, such as, for example, CSV, TSV, XLS, XLSX. These formats do not preserve much of the domain semantics and structure. Also, tabular data represented in the above mentioned formats can be syntactically quite heterogeneous[9] and leaves many semantic ambiguities open, which make interpreting, integrating and visualizing the data difficult. In order to support the exploitation of tabular data, it is necessary to transform the data to standardized formats facilitating the semantic description, linking and integration, such as RDF. To guide an automatic transformation to RDF, we define in the following section a canonical format for tabular data and categorize any possible deviations. By analysing a set of 100 dataset resources we compile a comprehensive list of such issues.

A CANONICAL MODEL OF TABULAR DATA

The following definitions formalize our concept of canonical tabular data.

DEFINITION 1. *A table $T = (H, D)$ is a tuple consisting of a header H and data D, where:*

- *the header $H = \{h_1, h_2, \ldots, h_n\}$ is an n-tuple of header elements h_i.*
- *the data* $D = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} \end{pmatrix}$ *is a (m, n) matrix consisting of n columns and m rows.*

Note, that the number of elements in the header and each row has to be the same. Given this definition of a tabular data table, we can describe deviations from this canonical model in three categories: (1) table level, (2) header level and (3) data level.

On the table level possible deviations are:

- *T-Metadata.* Metadata is embedded above or below the table. Publishers tend to append information about geographical location, time range, license etc. beside the table.
- *T-Whitespace.* The table has preceding or succeeding empty rows or columns.
- *T-Multiple.* Several semantically distinct tables are represented as one syntactic table.

The header level problems are:

- *H-Missing.* The header is empty or missing: $H = \{\}$
- *H-Duplicate.* The header is repeated:

[9] Informational RFC for CSV: http://www.ietf.org/rfc/rfc4180.txt

$$H = \begin{pmatrix} h_1 & h_2 & \cdots & h_n \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_n \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}$$

- *H-Multiple-column-cell.* One or several header cells occupy multiple columns: $H = \{h_1, h_2, h_2, h_2, h_3 \cdots h_n\}$
- *H-Incomplete.* One or several header cells are empty: $H = \{h_1, h_2, empty, h_4 \cdots h_n\}$
- *H-Multiple-header-rows.* The header is spread across several rows:
  $$H = \begin{pmatrix} h_{11} & h_{11} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \end{pmatrix}$$
- *H-Cardinality.* The header cardinality does not match the cardinality of the rows: $|H| \neq |R|$

On the data level encountered deviations are:

- *D-Duplicate.* The data row is repeated:
  $$D = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{11} & c_{12} & \cdots & c_{1n} \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}$$
- *D-Incomplete.* One or several data cells are empty:
  $$D = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & empty & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}$$
- *D-Missing.* The data row is empty or missing. Special case of *D-Incomplete*.
- *D-Multiple-column-cell.* One or several data cells occupy multiple columns:
  $$D = \begin{pmatrix} c_{11} & c_{11} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}$$
- *D-Multiple-row-cell.* Omitting duplicate value in the next row or column:
  $$D = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ empty & c_{22} & \cdots & c_{2n} \\ empty & c_{32} & \cdots & c_{3n} \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}$$

We have chosen 100 random CSV resources from PublicData.eu and checked them manually for these issues. Most of these resources (i.e. 62) contain no deviations from our canonical format and can be processed as is. 26 resources have table level problems. 23 have header and data level problems. Only 4 out of those 23 have problems other than empty rows or embedded metadata. Therefore, the main challenge is to identify the borders of the tabular data. The complete results of our survey are summarized in Table 3.

Deviation	**in # of CSV files**
No deviations	62
D-Missing	21
T-Metadata	20
T-Whitespace	11
T-Multiple	3
H-Cardinality	2
H-Multiple-header-rows	1
D-Multiple-row-cell	1

Table 3: Deviations from the canonical tabular model in 100 randomly selected CSV files.

TABULAR DATA TO RDF TRANSFORMATION

In this section we outline a formal approach for mapping tabular data to RDF. For this purpose, we first briefly summarize fundamental concepts of the RDF data model.

Preliminaries

The RDF primitives are:

- $\mathcal{U}$ is the set of URIs
- $\mathcal{B}$ is the set of all blank nodes
- $\mathcal{L}$ is the set of all literals
- $\mathcal{V}$ is the set of all variables
- $\mathcal{T}$ is the set of all *RDF term*s, defined as $\mathcal{U} \cup \mathcal{B} \cup \mathcal{L}$.

Furthermore, we make use of the following notions:

- $\mathcal{J}$ is the joint set of RDF terms and variables, defined as $\mathcal{T} \cup \mathcal{V}$.
- $\mathcal{Q}$ is the set of all *quads*, defined as $\mathcal{J} \times \mathcal{J} \times \mathcal{J} \times \mathcal{J}$.
- A *quad pattern* Q is defined as $Q \subset \mathcal{Q}$
- $\mathcal{R}$ is the set of all quad patterns, thus the powerset of $\mathcal{Q}$, denoted by $\mathcal{P}(\mathcal{Q})$
- A *quad* q is defined as $q \in \mathcal{Q}$.
- $vars(Q)$ is the set of variables appearing in Q
- A *concrete quad (pattern)* is a variable free quad (pattern).

Finally, for ease of discussion, we introduce the function σ that transforms all rows of a canonical table C into a *logical table* L, which is a set of corresponding partial functions from headings to data, i.e. a table

$$((id, name), \{(1, Anne), (2, John)\})$$

is transformed to:

$$\{\{(id, 1), (name, Anne)\}, \{(id, 2), (name, John)\}\}$$

Let $\mathcal{C}$ and $\mathcal{L}$ be the set of all canonical and logical tables, respectively.

$$\sigma : \mathcal{C} \rightarrow \mathcal{L}$$

$$\sigma(C) := \left\{ \bigcup_{1 \leq i \leq |C.H|} \{(C.H_i, d_i)\} \middle| d \in C.D \right\}$$

Generating RDF from logical tables

Based on the previously introduced primitives, we are now able to formally capture the nature of RDF mapping approaches for tabular data.

A relational data to RDF (*R2R*) mapping m is a three-tuple (P, L, f):

- P is *quad pattern* which acts as the *template* for the construction of triples and relating them to named graphs. The template is instantiated once for each row of the logical table. We use the notation V_P for referring to the set of SPARQL variables used in the template.
- L is the logical table to be converted to RDF.
- f is mapping of signature $L \to (\mathcal{V} \to \mathcal{T})$: f yields for each element of the logical table L a partial function that binds the variables of the template P to RDF terms in $\mathcal{T}$. Note that we do not require all variables of P to be bound, which enables us to support NULL values in the source data.

An R2R mapping is valid, if its *evaluation* yields a concrete quad pattern that conforms to an *RDF dataset*[10].

Given a quad pattern $Q \subset \mathcal{Q}$ and a partial function $a : \mathcal{V} \to \mathcal{T}$, we define the substitution operator

$$\rho_{[a]} : \mathcal{R} \to \mathcal{R}$$

$\rho_{[a]}(Q)$ yields a new concrete quad pattern Q' with all variables replaced in accordance with a. Any quads of Q with unbound variables in a are omitted in Q'.

An evaluation of a mapping m proceeds by passing each row of L as an argument to f, thereby obtaining the bindings for $vars(P)$, which are used to instanciate the template P for finally creating concrete quads. Let $\mathcal{M}$ be the set of all mappings, then function $eval$ can then be defined as:

$$eval : \mathcal{M} \to \mathcal{R}$$

$$eval(m) = \bigcup_{l \in m.L} \{\rho_{[m.f(l)]}(m.P)\}$$

Implementation

The RDF generation approach is implemented in the *Sparqlify-CSV* tool which is part of the *Sparqlify* project. This project also features a novel mapping language for expressing mappings between tabular and RDF representations, namely *Sparqlify-ML*. An example Sparqlify-ML view definition is shown in Listing 1.

Listing 1: A simple view definition that creates URIs and plain literals from the first and second column of a table respectively.

```
Prefix ex: <http://example.org/>
Create View Template person-mapping As
  Construct {
    ?s
      a ex:Person ;
      ex:name ?n
  }
  With
    ?s = uri(concat(ex:, ?1))
    ?n = plainLiteral(?2)
```

It is noteworthy, that this syntax closely follows the previously introduced formal model: the `Construct` part corresponds to Q, the `With` part to f, and the logical table L is constructed from a tabular data source, such as a CSV file, that is specified as a command line argument for Sparqlify-CSV. Additionally, the Sparqlify-ML grammar re-uses many production rules of the original SPARQL 1.0 grammar[11] as building blocks, which significantly simplified the implementation of the language.

In general, the mapping f can be arbitrarily implemented. However, Sparqlify was originally designed for SPARQL to SQL rewriting and thus, at present, only allows f to be defined in terms of expressions making use of a limited number of operator symbols and function names. In general, each canonical table can be seen as an SQL table. Expressions specifies the RDF term type to generate from the underlying SQL expression.

Listing 2: Excerpt of valid expressions for the Sparqlify-ML WITH part. Note that the same notion is used for column references and SPARQL variables

```
withPart
  : (var '=' rdfTermCtorExpr)*
  ;

// plainLiteral: (value, optional languageTag)
// typedLiteral: (value, datatype)
rdfTermCtorExpr
  : BNODE '(' sqlExpr ')'
  | URI '(' sqlExpr ')'
  | PLAINLITERAL '(' sqlExpr (',' sqlExpr)? ')'
  | TYPEDLITERAL '(' sqlExpr ',' sqlExpr ')'
  ;

sqlExpr
  : sqlLiteral
  | columnRef
  | CONCAT '(' sqlExpr ')'
  | URLENCODE '(' sqlExpr ')'
  | URLDECODE '(' sqlExpr ')'
  ;

columnRef
  : '?' NAME
  ;
```

These expressions, as the name suggest, are used for constructing RDF terms from literals, function symbols and column references to a logical table.

In the future we could distinguish between ETL and SPARQL-SQL rewriting profiles, and allow more powerful expressions and transformations in the former case.

[10] http://www.w3.org/TR/sparql11-query/

[11] http://www.w3.org/TR/rdf-sparql-query/

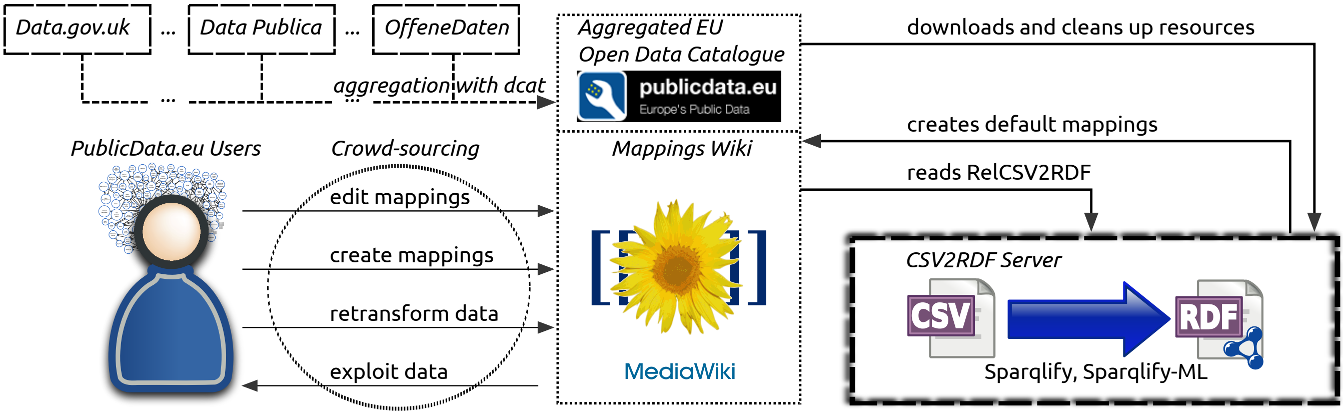


Figure 2: Overall architecture of our CSV2RDF extension for PublicData.eu.

CROWD-SOURCING OF MAPPINGS

The completely automatic RDF transformation as well as the detection and correction of tabular data deviations is not feasible. Therefore, we devise an approach where the effort is shared between machines and human users. Our mapping authoring environment is based on the popular *MediaWiki*[12] system. The resulting *mapping wiki* located at *wiki.publicdata.eu* operates together with PublicData.eu and helps users to map and convert tabular data to RDF in a meaningful way.

To leverage the wisdom of the crowd, mappings are created automatically first and can then be revised by human users. Thus, users improve mappings by correcting errors of the automatic conversion and the cumbersome process of creating mappings from scratch can be avoided in the most cases. In order to realize the automatic conversion, our implementation downloads and cleans resources available on PublicData.eu. In a next step it extracts the header of the tabular data file, creates a default mapping automatically and converts the data based on this mapping to RDF using Sparqlify-CSV as described in the previous section. Finally, a page on wiki.publicdata.eu is created for each resource containing the mappings, links to rerun the transformation routine and download links for the resulting RDF files. An overview of the entire application is depicted in Figure 2.

At the time of writing PublicData.eu contains 12,255 CSV resources. Our automatic transformation crawls these CSV resources (we work on extending our implementation to be able to deal with other tabular data formats such as XLS). 2060 (16.8%) of the CSV resources were not available due to response time-outs, server errors or missing files. 218 (1.78%) resources have invalid URIs, for example, URI schemes such as ttp or hhttp as well as typos and trailing whitespace. 609 (4.97%) resources do not contain tabular data in CSV format. 81 (0.66%) resources contain several tables inside one archive file, which makes it difficult to create an explicit identifier for the given resource. The crawl run statistics are summarized in Table 4.

CSV resources	12,255
HTTP status code 200	9,977
HTTP status code 4xx or 5xx	2,060
Broken links	161
HTML / XML pages	591
Archives containing one file	81
Archives with more than one file	55
XLS / XLSX files	146
Torrent files	10
Other problems	8
CSV resources after validation	9,370
Amount of data	33 GB

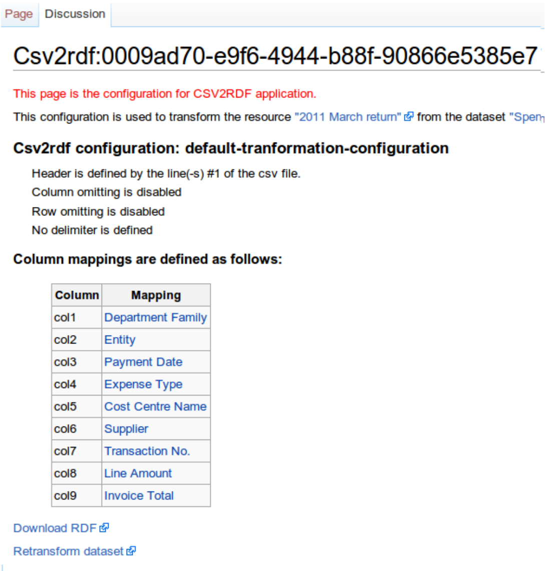
Table 4: CSV data collection and cleaning summary.

The second step after validation is the automatic creation of the default mapping and conversion to RDF. In order to obtain an RDF graph from a table T we essentially use the *table as class* approach [2] (as formally described in the last section), which generates triples as follows: subjects are generated by prefixing each row's id (in the case of CSV files this by default is the line number) with the corresponding CSV resource URL. The headings become properties in the ontology name space. The cell values then become the objects. Note that we avoid inferring classes from the CSV file names, as the file names too often turned out to be simply labels rather than meaningful type names. Listing 3 shows the default mapping expressed in Sparqlify-ML syntax.

Listing 3: Sparqlify-ML default mapping. Note that ?rowId and ?headingName{index} are special variable names that get assigned appropriate values by the Sparqlify-CSV engine.

```
Prefix pdd: <http://data.publicdata.eu/>
Prefix pdo:
   <http://wiki.publicdata.eu/ontology/>
Create View Template DefaultMapping As
   Construct {
     ?s
       ?p1 ?o1 ;
       ?p2 ?o2 ...
   } With
     ?s = uri(concat(pdd:,'csv-path/',?rowId))
     ?p1 = uri(concat(pdo:, ?headingName1))
     ?o1 = plainLiteral(?1)
     ?p2 = ...
```

[12] http://www.mediawiki.org/



Csv2rdf:0009ad70-e9f6-4944-b88f-90866e5385e7

This page is the configuration for CSV2RDF application.

This configuration is used to transform the resource "2011 March return" from the dataset "Spen

Csv2rdf configuration: default-tranformation-configuration

Header is defined by the line(-s) #1 of the csv file.
Column omitting is disabled
Row omitting is disabled
No delimiter is defined

Column mappings are defined as follows:

Column	Mapping
col1	Department Family
col2	Entity
col3	Payment Date
col4	Expense Type
col5	Cost Centre Name
col6	Supplier
col7	Transaction No.
col8	Line Amount
col9	Invoice Total

Download RDF

Retransform dataset

```
1  {{CSV2RDFHeader}}
2
3  ...
4
5  {{RelCSV2RDF
6   | name      = default-mapping
7   | header    =  1
8   | omitRows = -1
9   | omitCols = -1
10  | delimiter =
11  | col1 = Department Family
12  | col2 = Entity
13  | col3 = Payment Date
14  | col4 = Expense Type
15  | col5 = Cost Centre Name
16  | col6 = Supplier
17  | col7 = Transaction No.
18  | col8 = Line Amount
19  | col9 = Invoice Total
20 }}
```

Figure 3: Dataset resource page on wiki.publicdata.eu with the mapping definition (left) and the wiki text mark up for the mapping (right).

We utilize *Sparqlify*[13] for conversion to RDF. The framework communicates with Sparqlify-CSV via its command-line interface which is passed two parameters: the file to convert and a mapping in *Sparqlify-ML*[14] mapping language syntax. Although the Sparqlify-ML syntax should not pose any problems to users familiar with SPARQL, it is yet too complicated for novice users and therefore less suitable for being crowd-sourced. To even lower the barrier, we define a simplified mapping format, which releases users from dealing with the Sparqlify-ML syntax. Our format is based on MediaWiki templates and thus seamlessly integrates with MediaWiki. We created a template called *RelCSV2RDF*, which defines the following parameters (line numbers correspond to Figure 3):

- (line 13) *name*: a string, which identifies the mapping and must be unique within the scope of one resource;
- (line 14) *header*: an integer or an integer range, which determines the position of header row(s);
- (line 15) *omitRows* and *omitCols*: integer ranges, which determine rows and columns to be omitted from the conversion;
- (line 17) *delimiter*: a symbol, defining the column delimiter for the tabular data file;
- (lines 18-26) *col1, col2, col3 etc.*: strings, which specify RDF properties to be used for the conversion of each column of the table.

The default mapping, which our automatic conversion process generates, uses CSV column headers as identifiers for respective properties. These properties are then instantiated for each of the respective column values. A consequence of this approach is, that CSV files using the same column header will produce RDF containing the same properties. We argue, that in the majority of the cases this behavior is desirable, especially, if multiple datasets were exported to CSV from the same backend system and have the same structure and headers. However, this automatic mapping can also result in incorrect property identification in cases, where columns in CSV files have the same header label, but different meaning. Our crowd-sourcing approach enables to quickly resolve such problems once identified.

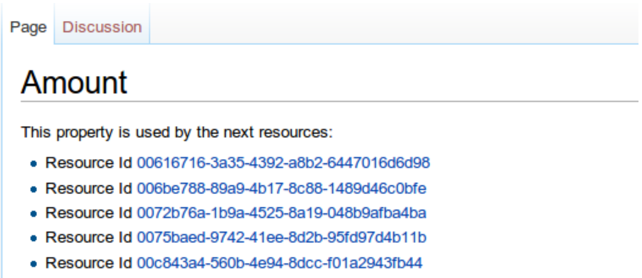
At the end of the transformation a page is created for each resource on the mappings wiki at wiki.publicdata.eu (e.g. Figure 3). The resource page comprises links to the corresponding resource and dataset on PublicData.eu as well as one or several mappings. Each mapping is rendered using the RelCSV2RDF template into a human-readable description of the parameters including links for transformation rerun and RDF download.

The mapping wiki uses the *Semantic MediaWiki* [6] (SMW) extension, which enables semantic annotations and embedding of search queries over these annotation within wiki pages. The RelCSV2RDF template utilizes SMW and automatically attaches semantic links (using `has_property`) from mappings to respective property pages. This allows users to navigate between dataset resources which use the same properties, that is dataset resources are connected through the properties used in their mappings. For each property we created a page in the mapping wiki with the list of dataset resources, that utilize the corresponding property. The example page located at `http://wiki.publicdata.eu/wiki/Amount` is depicted in Figure 5. To construct these pages we use a simple template with the following embedded SMW query:

```
{{#ask: [[has property::{{PAGENAME}}]]
  | mainlabel = Resource Id |
  | format = ul }}
```

[13] `http://sparqlify.org/`

[14] `http://sparqlify.org/wiki/Sparqlify_mapping_language`



Amount

This property is used by the next resources:

- Resource Id 00616716-3a35-4392-a8b2-6447016d6d98
- Resource Id 006be788-89a9-4b17-8c88-1489d46c0bfe
- Resource Id 0072b76a-1b9a-4525-8a19-048b9afba4ba
- Resource Id 0075baed-9742-41ee-8d2b-95fd97d4b11b
- Resource Id 00c843a4-560b-4e94-8dcc-f01a2943fb44

Figure 5: Property page with the list of resources.

This query retrieves all pages, which have the following property: `[[has property::{{PAGENAME}}]]`, adds the label `Resource Id` to each retrieved page and formats the output as a list. At the time of writing the page `Amount` returns 3,151 links to the dataset resources using this property. Users can also write custom queries to narrow the result set. For example, only 118 dataset resources, which use `Amount` in conjunction with `Entity` and `Expenditure Type`, are retrieved using the following SMW query:

```
#ask: [[has property::Amount]]
      [[has property::Entity]]
      [[has property::Expenditure Type]]
```

In order to navigate to the wiki page every dataset and resource page on PublicData.eu has an RDF link as depicted on Figure 4.

RESULTS

We downloaded and cleaned 9,370 CSV files, that consume in total 33 GB of disk space. The distribution of the file sizes in Figure 6 shows, that the vast majority (i.e. 85%) of the published datasets are less than 100 kB in the size. A small amount of the resources at PublicData.eu (i.e. 14.5%) are between 100 kB and 50 MB. Only 44 resources (i.e. 0.5%) are large and very large files above 50 MB, with the largest file comprising 3.3 GB. As a result, the largest 41 out of the 9,370 converted RDF resources account for 7.2 (i.e. 98.5%) out of overall 7.3 billion triples.

During the automatic conversion our framework created 9,370 wiki pages on the mappings wiki. The `has_property` property is used 80,676 times and maps to 13,490 distinct properties. The 10 most used properties are:

Property	**Occurrences**
Entity	3,593
Supplier	3,505
Amount	3,151
Date	3,104
Expense Type	2,352
Expense Area	2,240
Department Family	2,036
Transaction Number	1,849
Transaction number	1,425
Expense type	1,395

The results of the transformation process are summarized in Table 5. Our efficient Sparqlify RDB2RDF transformation engine is capable to process CSV files and generate approx. 4.000 triples per second on a quad core 2.2 GHz machine. As a result, we can process CSV files up to a file size of 50MB within a minute. This enables us to re-transform the vast majority of CSV files on demand, once a user revised a mapping. For files larger than 50MB, the transformation is currently queued and processed in batch mode.

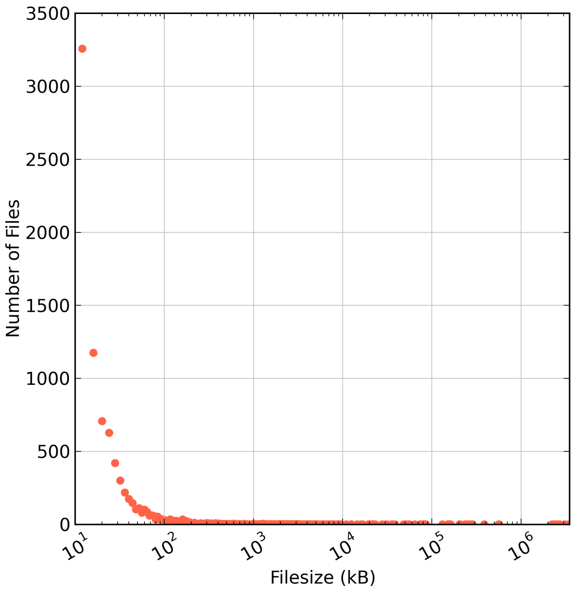


Figure 6: File size distribution of CSV files available at PublicData.eu.

CSV resources converted	9,370
CSV resources volume	33 GB
Number of generated triples	7.3 billions
Number of entity descriptions	154 millions
Avg. number of properties per entity	47
Generated default mappings	9,370
Overall properties	80,676
Distinct properties	13,490

Table 5: Transformation results summary.

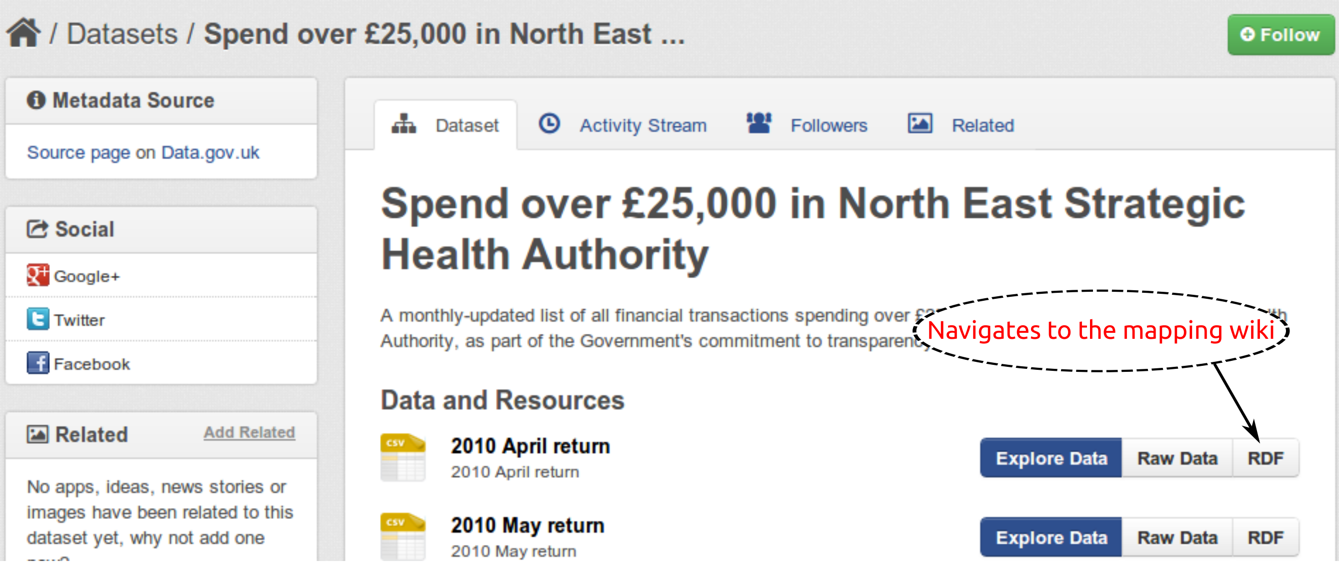
RELATED WORK

We can roughly classify related work into approaches for tabular data to RDF conversion, lifting and linking open governmental data as well as tabular data extraction.

Tabular data to RDF conversion

There is a plethora of work on tools for converting various data formats to RDF. Tim Lebo maintains a github wiki page[15] listing as many as 37 tools for this purpose. These tools differ in supported input formats (CSV, Excel, XML), mapping language (syntax, expressivity) and implementation programming language (e.g. Java, XSLT, PHP). Specifically for tabular data, one of the most advanced tools in this area is

[15] https://github.com/timrdf/csv2rdf4lod-automation/wiki/Alternative-Tabular-to-RDF-converters



No apps, ideas, news stories or images have been related to this dataset yet, why not add one

Spend over £25,000 in North East Strategic Health Authority

A monthly-updated list of all financial transactions spending over £
Authority, as part of the Government's commitment to transparen

Data and Resources

2010 April return
2010 April return

2010 May return
2010 May return

Figure 4: Dataset description page at PublicData.eu showing the integration with the mapping wiki (highlighted red).

Tables[16], which offers the *Tables Language*. This language is similar to Sparqlify-ML in the sense that it re-uses syntactic constructs already known from SPARQL. However, it introduces additional features specifically for CSV-RDF transformations, such as loops for iterating over CSV files in ZIP archives and workbooks and pages in Excel spreadsheets.

An effort to standardize a mapping language for expressing the conversion of data stored in relational databases (of which CSV files can be seen as a special case) to RDF is *R2RML*[17] which recently became a W3C recommendation. A closely related recommendation is the *Direct Mapping*[18]. which standardizes rules for obtaining default RDF graphs from relational data in absence of a user defined mapping.

In regard to the generation of RDF terms for tables, there is strong evidence for Sparqlify-ML to feature the same expressivity as R2RML: For every R2RML test case[19] it was possible to manually create corresponding Sparqlify-ML view definitions yielding the expected output.

Lifting and linking Open Government Data

The *Data-Gov Wiki* project[20] is the one of the largest initiatives with regard to the publishing of Linked Open Government Data (LOGD). At the time of writing the Data-Gov Wiki hosts 417 RDF datasets, covering the content of 703 out of the 5,762 datasets released at *data.gov* and contributing 6.46 billion RDF triples to the LOD cloud [3, 4]. The conversion process is described in [7] and divided into two steps: (1) row-based raw conversion to RDF and (2) RDF enhancement. In the first step well-formed CSV files with headers are automatically converted to RDF without linking entities to the existing ontologies. The second step is supervised by experts and results in an enrichment of the converted RDF without deleting the automatically converted triples. Our approach differs from the one followed by the Data-Gov Wiki in that we base our conversion on a formalized canonical tabular data model and employ a deliberately simple mapping language embedded in Semantic Wiki pages in order to facilitate the crowd-sourcing of mappings.

In [10] the authors criticize the naive automatic conversion used by the Data-Gov Wiki. They propose an approach for automatic mapping of column headers to classes from an appropriate ontology, linking cell values to entities and discovering or identifying relationships between columns. However, the work is restricted only to well-formed CSV, while our mapping language aims to also deal with deviations from canonical tabular data.

In [9] the authors represent a Government Linked Data publishing pipeline, based on Google Refine. However, Google Refine is not a collaborative platform and thus the publishing process described in the paper can not be crowd-sourced easily.

Tabular data extraction

A methodology for automatic transformation and generation of semantic (F-Logic) frames from table-like structures is presented in [11]. The authors implement the *TARTAR* (Transforming ARbitrary TAbles into fRames) system, which processes HTML tables. The methodology is based upon the table model described by Hurst in [5] and distinguish three types of the tables: (1) 1-dimensional tables, (2) 2-dimensional tables and (3) complex tables. The authors point out some deviations (H-Duplicate, H-Multiple-column-cell, D-Multiple-column-cell, T-Multiple) of the tabular data as features of complex tables, but do not provide a formalization. Some of the deviations are also described on the Data-Gov Wiki.[21]

[16] http://idi.fundacionctic.org/tabels/
[17] http://www.w3.org/TR/r2rml/
[18] http://www.w3.org/TR/rdb-direct-mapping
[19] http://www.w3.org/2001/sw/rdb2rdf/test-cases/
[20] http://data-gov.tw.rpi.edu/
[21] http://data-gov.tw.rpi.edu/wiki/Category:Issue_Report

CONCLUSIONS AND FUTURE WORK

In this article we presented a formalization of tabular data as well as its mapping and transformation to RDF. We implemented our approach in such a way, that the mapping creation can be easily crowd-sourced in order to make the large-scale transformation of tabular data registered at Open Data catalogs such as PublicData.eu possible.

Our approach is currently only capable to deal with one dimensional tabular data for which RDF entities are generated per row. However, statistical data is represented in tables with a region comprising additional dimensions on the left-hand side. For such tables RDF entities have to be created for every cell. Also, an additional dimension in a table results in additional possible deviations, which have to be identified and classified. The automatic header recognition in the CSV files is one of the most important problems. According to our analysis 20% of the CSV files have *T-Metadata* deviations, where metadata is embedded before the table. In such cases the location of the header is currently not properly determined. Also, 3% of the CSV files have *T-Multiple* deviation, which aggravates the identification of the header and data. The header recognition problem is a classification problem and can be solved using supervised machine learning. Possible features to be employed for learning are: the frequency of the words in the column, position of the line, ratio between the overall number of entries in the line, divided by non-empty entries in the line.

The work on crowdsourcing the semantification of data portals described in this article is only the first step in a larger research and development agenda. Ultimately, we envision to semantically enrich and interlink and integrate data portals into a distributed human development data warehouse. Just as data warehouses and business intelligence are now integral parts of every larger enterprise, data portals can be the nucleus for a human development data warehouse. In such a human development data warehouse, a large number of statistical data and indicators are published by different organizations that could be integrated automatically or semi-automatically in order to obtain a more interactive picture of the development in a certain region, country or even on the globe. Currently, the indicators (e.g. the Human Development Index) are very coarse-grained, mainly referring to countries. By integrating semantified ground-truth data made available through data portals, such indicators can be computed on a much more fine-grained level, such as for cities and regions as well as with regard to different groups of people (e.g. per gender, ethnicity, education level). Policy making would be based on more rational, transparent and observable decisions as it is advocated by evidence-based policy.

Acknowledgment

We would like to thank Open Knowledge Foundation's Andreea Bonea, Sean Hammond and John Martin for their help with the PublicData.eu integration. This work was supported by a grant from the European Union's 7th Framework Programme provided for the project LOD2 (GA no. 257943).

REFERENCES

1. Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., and Aumueller, D. Triplify: Light-weight linked data publication from relational databases. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, J. Quemada, G. León, Y. S. Maarek, and W. Nejdl, Eds., ACM (2009), 621–630.

2. Berners-Lee, T. Relational databases on the semantic web. Design Issues, http://www.w3.org/DesignIssues/RDB-RDF.html.

3. Ding, L., DiFranzo, D., Magidson, S., McGuinness, D. L., and Hendler, J. The data-gov wiki: A semantic web portal for linked government data. In *Proceedings of the 8th ISWC (poster/demo session)* (Oct 2009).

4. Ding, L., Lebo, T., Erickson, J. S., DiFranzo, D., Graves, A., Williams, G. T., Li, X., Michaelis, J., Zheng, J., Shangguan, Z., Flores, J., J, D. L. M., and Hendler, J. A. Twc logd: A portal for linked open government data ecosystems. *Web Semantics: Science, Services and Agents on the World Wide Web 9*, 3 (2011).

5. Hurst, M. *The Interpretation of Tables in Texts*. University of Edinburgh, PhD Thesis, 2000.

6. Krötzsch, M., Vrandecic, D., Völkel, M., Haller, H., and Studer, R. Semantic wikipedia. *Journal of Web Semantics 5* (September 2007), 251–261.

7. Lebo, T., and Williams, G. T. Converting governmental datasets into linked data. In *Proceedings of the 6th International Conference on Semantic Systems*, I-SEMANTICS '10, ACM (New York, NY, USA, 2010), 38:1–38:3.

8. Maali, F., Cyganiak, R., and Peristeras, V. Enabling interoperability of government data catalogues. In *Proceedings of the 9th IFIP WG 8.5 international conference on Electronic government*, EGOV'10, Springer-Verlag (Berlin, Heidelberg, 2010), 339–350.

9. Maali, F., Cyganiak, R., and Peristeras, V. A publishing pipeline for linked government data. In *ESWC*, E. Simperl, P. Cimiano, A. Polleres, s. Corcho, and V. Presutti, Eds., vol. 7295 of *Lecture Notes in Computer Science*, Springer (2012), 778–792.

10. Mulwad, V., Finin, T., and Joshi, A. Automatically Generating Government Linked Data from Tables. In *Working notes of AAAI Fall Symposium on Open Government Knowledge: AI Opportunities and Challenges*. November 2011.

11. Pivk, A., Cimiano, P., Sure, Y., Gams, M., Rajkovič, V., and Studer, R. Transforming arbitrary tables into logical form with tartar. *Data Knowl. Eng. 60*, 3 (Mar. 2007), 567–595.