A Semi-Automatic Tool for Linked Data Integration

Benjamin Moreau, Nicolas Terpolilli, Patricia Serrano-Alvarado

A Semi-Automatic Tool for Linked Data Integration

Benjamin Moreau[1,2], Nicolas Terpolilli[1], and Patricia Serrano-Alvarado[2]

[1] OpenDataSoft {Name.Lastname}@opendatasoft.com
[2] Nantes University, LS2N, CNRS, UMR6004, 44000 Nantes, France
{Name.LastName@}univ-nantes.fr *

Abstract. Linked Data (LD) is a set of best practices to publish data in RDF format. Transforming structured datasets into RDF datasets is possible thanks to *RDF Mappings*. To be able to define such mappings, it is necessary to be familiar with the LD practices and to know perfectly concerned datasets. An obstacle to the democratisation of the LD is that few people satisfy these two conditions. We believe that tools making easy the process of LD integration will foster the LD growth. In this demonstration, we present a chatbot-like tool that can semi-automatically generate RDF mappings for existing structured datasets. The challenge is to automate part of the integration process that requires getting familiar with RDF.

1 Introduction and Motivation

Linked Data (LD) is a set of best practices to publish data in RDF[3] format. Data published as LD are described according to ontologies that represent relationships (i.e., properties) between concepts (i.e., classes) of a domain. RDFS[4] and OWL[5] are semantic web languages to describe ontologies. Using existing and widely used ontologies increases interoperability among LD datasets.

An *RDF Mapping* defines the transformation of a structured dataset (column-based, JSON, etc.) into an RDF dataset. It maps columns of a dataset to terms of an RDF graph.

Writing mappings is not easy. Consider Figure 1 that shows an excerpt of a structured dataset describing Roman Emperors and Figure 2 that represents an RDF mapping allowing to transform such dataset in RDF. Writing this mapping requires to answer several questions, for instance: (i) what concepts contain the *Name* and *Birth city* columns? In this case, *Name* contains entities that are Persons (emperors) and *Birth city* contains entites that are Places (cities). (ii) What are the relationships between these two concepts? Here, Places are birth places of Persons. (iii) Which existing ontologies are relevant to describe these concepts? In this example, DBpedia, GeoNames, etc.

[3] https://www.w3.org/RDF/
[4] https://www.w3.org/TR/rdf-schema/
[5] https://www.w3.org/TR/owl2-overview/

string	date	string	string	float	float
Name	**Birth**	**Birth City**	**Birth Province**	**Lat**	**Long**
Augustus	0062-09-23	Rome			
Caligula	0012-08-31	Antitum			
Claudius	0009-08-01	Lugdunum	Gallia Lugdunensis	47.932559	0.191854
...	...	...	...	...	...

Fig. 1. Excerpt of a structured and typed dataset describing Roman emperors.

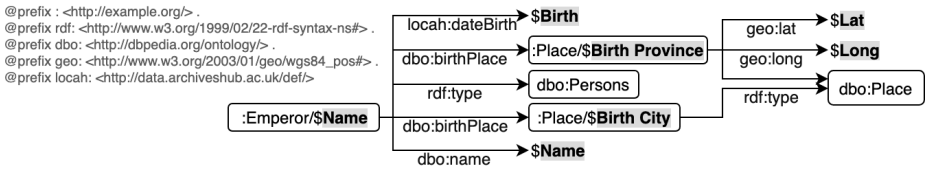

Fig. 2. RDF mapping for Roman emperors dataset. Text highlighted in grey and beginning with $ are references to a column of the dataset.

Answering these questions requires both, to know the dataset perfectly and to be familiar with RDF concepts such as RDFS, OWL and RDF mapping languages. Unfortunately, many data producers are not familiar with RDF and are not yet ready to invest time to integrate their data. In this work, we focus on *how to simplify as much as possible the integration of existing structured datasets as Linked Data*. The challenge we face is to automate part of the integration process that requires getting familiar with RDF.

RML [1] and SPARQL-Generate [6] are two RDF mapping languages. Even if there exist simplified and human-readable syntaxes of mapping languages like YARRRML [3], writing a mapping requires to be familiar with RDF. Recently, interesting tools like KARMA [2], RMLeditor [4] or Juma [5] have been proposed to assist users during the creation of an RDF mapping. However, these tools are not easy to use for users that are not familiar with RDF concepts.

We propose a chatbot-like tool that can generate an RDF mapping from a structured dataset by only asking simple questions to users about their dataset. Our tool can simply and quickly integrate datasets as Linked Data and can encourages new users to make their first steps into Linked Data.

2 A Chatbot-Like Tool for Linked Data Integration

To generate an RDF mapping from a structured dataset, our tool uses two knowledge graphs, DBpedia and YAGO, the ontologies of LOV[6], and the semantic web languages OWL and RDFS.

Roughly speaking, from a set of instances of each column, our tool searches corresponding entities in DBpedia and YAGO. The goal is to find a class corre-

[6] `https://lov.linkeddata.es/dataset/lov/`

sponding to each column. Then, similarly, LOV is used to find the most relevant properties that may correspond to column names, such that instances of a column correspond to the object of a property. To confirm and complete these correspondences, the tool asks simple questions to the user. User confirmed correspondences allow to generate a first RDF mapping. Finally, this mapping is saturated with entailment rules of OWL and RDFS.[7]

Using the Roman emperor dataset of Figure 1, in the *class correspondence* step the *Augustus* value of the *Name* column corresponds to the entity `http://dbpedia.org/resource/Augustus` of the class `dbo:Person` in DBpedia. Thus, *Augustus* is identified as an entity of the class `dbo:Person`. In this example, we obtain two class correspondences suggesting that columns *Name* and *Birth City* contain respectively entities of the classes `dbo:Person` and `dbo:Place`. These correspondences are suggested to the user with simple yes or no questions: "Does the column Name in your dataset contain Persons?". In order to hide URIs, questions are built using the `rdfs:label` property of classes.

In the *property correspondence* step, our tool obtains 5 property correspondences suggesting that columns *Name*, *Birth*, *Birth City*, *Lat* and *Long* are respectively objects of properties `dbo:name`, `locah:dateBirth`, `dbo:birthPlace`, `geo:lat` and `geo:long`. Again, these correspondences are suggested to the user with simple yes or no questions: "It seems that the column Lat is the latitude of a Spatial thing. Is it true?". Question is built using the `rdfs:label` and the `rdfs:domain` of the property.

To complete confirmed correspondences, the tool asks the user to select the column of the dataset that will correspond to the subject of the property. If the user confirms the `geo:lat` correspondence with column *Lat*, the tool asks "latitude is a characteristic of a Spatial Thing. Select the column that contains Spatial Thing.". In our example, if the user answers correctly, the column *Birth Province* is used as the subject of the `geo:lat` property.

Our tool uses heuristics to reduce the number of questions. It suggests at most one class and one property for each column. Only the class that corresponds to the most instances of a column is suggested. The property that is suggested for a column is the property that has the best popularity score in the LOV answer. Our tool does not suggest a property if its LOV score is lower than a fixed lower bound. Moreover, to improve the pertinence of suggested properties, the type of a column can also be added in the text search. In our example, it searches for the property *Birth date* instead of *Birth*.

From user confirmed correspondences, our tool generates a first RDF mapping. In a final step, our tool saturates this mapping by applying RDFS and OWL entailment rules. Using the range and domain of all properties (`rdfs:range` and `rdfs:domain`), new classes are infered. This is possible because the domain of a property represents the class of the subject and the range represents the type (i.e., a class or the literal datatype) of the object. In our example, for instance,

[7] We only consider rules 2, 3, 5, 7, 9 and 11 from RDFS: `https://www.w3.org/TR/rdf11-mt/#rdfs-entailment` and rules based on `owl:equivalentClass` and `owl:equivalentProperty` from OWL: `https://www.w3.org/TR/owl-ref`

the user defined *Birth Province* as the subject of the *latitude* property. In the GeoNames ontology, the `rdfs:domain` of this property is `geo:SpatialThing`. Thus, our tool infers that the column *Birth Province* contains entities of type `geo:SpatialThing`. Our tool also takes into account `owl:equivalentClass`, `owl:equivalentProperty`, `rdfs:subClassOf` and `rdfs:subPropertyOf` properties of concerned ontologies. The RDF mapping in YARRRML of our example is available at `https://git.io/fjKY6` and the result of the transformation in turtle is available at `https://git.io/fjKYo`.

3 Demonstration

We implemented a chatbot-like tool that is able to generate YARRRML mappings for datasets of the OpenDataSoft's data network[8]. We chose YARRRML because, at our knowledge, it is the most human readable and understandable RDF mapping syntax for users that are not familiar with LD. Source code of out tool is available at GitHub[9] under the MIT license. Our tool is available as a web service at `https://chatbot.opendatasoft.com/`. During the demonstration, attendees will be able to use the tool to generate RDF mappings for any structured dataset of the network as LD (e.g., Roman Emperors[10]).

References

1. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In: Workshop on Linked Data on the Web (LDOW) collocated with WWW (2014)
2. Gupta, S., Szekely, P., Knoblock, C.A., Goel, A., Taheriyan, M., Muslea, M.: Karma: A System for Mapping Structured Sources Into the Semantic Web. In: Extended Semantic Web Conference (ESWC), Poster&Demo (2012)
3. Heyvaert, P., De Meester, B., Dimou, A., Verborgh, R.: Declarative Rules for Linked Data Generation at Your Fingertips! In: Extended Semantic Web Conference (ESWC), Poster&Demo (2018)
4. Heyvaert, P., Dimou, A., Herregodts, A.L., Verborgh, R., Schuurman, D., Mannens, E., Van de Walle, R.: RMLEditor: a Graph-Based Mapping Editor for Linked Data Mappings. In: Extended Semantic Web Conference (ESWC) (2016)
5. Junior, A.C., Debruyne, C., O'Sullivan, D.: An Editor that Uses a Block Metaphor for Representing Semantic Mappings in Linked Data. In: Extended Semantic Web Conference (ESWC), Poster&Demo (2018)
6. Lefrançois, M., Zimmermann, A., Bakerally, N.: A SPARQL Extension For Generating RDF From Heterogeneous Formats. In: Extended Semantic Web Conference (ESWC) (2017)

[8] `https://data.opendatasoft.com`

[9] `https://github.com/opendatasoft/ontology-mapping-chatbot`

[10] Roman emperors dataset is available at `https://data.opendatasoft.com/explore/dataset/roman-emperors%40public/table/` and mapping of the dataset can be generated at `https://chatbot.opendatasoft.com/chatbot/roman-emperors@public`