An Approach for Accessing Linked Open Data for Data Mining Purposes

Andreas Nolle*, German Nemirovski*, Álvaro Sicilia°,
Joan Pleguezuelos Porqueras°

*Albstadt-Sigmaringen University
{nolle,nemirovskij}@hs-albsig.de
°ARC Enginyeria i Arquitectura La Salle, Universitat Ramon Llull
{asicilia,jpleguezuelos}@salle.url.edu

Abstract

In the recent time the amount of semantic data publicly available online grows continuously. This data is a valuable source of information that can be inferred by means of different technologies. Apart of reasoning on data semantics classical OLAP or probabilistic data mining techniques can be applied for pattern recognition, prediction or clustering. However the conversion of linked (open) data basically stored as RDF triples to data structures appropriate for data mining, such as propositional representations or flat tables, appears to be a challenging task. In this paper we propose an approach addressing this task. The approach has been implemented in form of an operator for the RapidMiner software, a data mining environment widely used in the industry and research.

1 Introduction

The idea to close the gap between two technologies for data analysis, i.e. ontology based inference on the one hand and data mining on the other hand, wins interest of scientists increasingly. While the statistical methods comprised under the umbrella term "data mining" are known since decades, the birth of so called semantic technologies embracing ontology-based knowledge representation and inference is connected with the paper of Gruber [2]. This paper was published in 1995 and introduces the term *ontology* in the knowledge representation context related to engineering sciences.

One year later Fayyad et al. [1] brought both terms together. However, ontologies are only mentioned in the concluding part of this paper as a perspective technology "allowing the use of prior human knowledge about the underlying process by the KDD[1] system". One of the first attempts to integrate ontology-based knowledge representation into the data mining process has been presented two years later by Simeon & Maher [13]. In their application case an ontology provided "context, structure and relationships for representation and integration of discovered patterns" to be identified in multimedia data related to the domain of building design.

In recent time the interdisciplinary research on the edge of data mining and ontology-based inference has been progressing significantly. One of the reasons for this progress is doubtless substantial growth of the Linked Open Data cloud [5]. This data is basically stored in RDF format and accessible over numerous SPARQL endpoints. In this connection a challenging task is to query the RDF data and to convert it to structures and formats which can be processed by data mining tools. The approach described in this paper is addressing this task. In particular it targets the retrieval and integration of RDF data into the processes designed using RapidMiner, a data mining environment widely used in the industry and research.

The paper is structured as following: In the section 2 it gives an overview of approaches related to the proposed one. Section 3 contains the technical description of the work done. Finally the section 4 describes conclusions and new challenges related to the preprocessing of RDF data for data mining purposes.

2 Related Work

In spite of the high interest of the scientific community and strong demand in various application domains, currently there exist not many practically applicable approaches targeting querying of RDF data sources and transforming query answers into formats required by data mining processes, e.g. by RapidMiner processes.

One of these approaches, RMonto [8,9,11], is an extension for RapidMiner. RMonto facilitates accessing semantic data and its further processing using machine learning algorithms. Since the data isn't transformed into the internal data format of RapidMiner this approach is more similar to semantic data mining [4]. Due to differences in data formats numerous data mining operators available in RapidMiner cannot be combined with RMonto. Therefore the existing operators have to be adapted or new operators have to be developed

[1] KDD stands for Knowledge Discovery in Databases that is roughly spoken a comprehensive process comprising apart of data analysis e.g. by data mining, data selection and data transformation of data as well as interpretation of analysis results. Though, in the recent time terms KDD and data mining in colloquial language are often used as synonyms.

in order to facilitate sophisticated data mining processes basing on RMonto, that is a significant limitation of the approach.

Another approach called rapidminer-semweb has been developed by Khan et al. [3,10]. The implementation in form of a RapidMiner operator is available online. Though, this approach is strongly related to the task of RDF data retrieval, it is, unfortunately, not mature enough to be applied in practice. Our evaluation of this operator has shown that both methods i) RDF file access, i.e. loading of RDF files from a local directory as well as ii) sending requests to SPARQL endpoints ended up with an error.

The third approach implemented by Paulheim & Frnkranz [7] rather focuses on enrichment of statistical data with ontology based "semantic" metadata available online, e.g. within the Linked Open Data cloud. However such metadata isn't used as immediate input for data mining processes. Therefore this approach doesn't address the task of RDF data retrieval directly.

Since this survey of available technologies did not deliver any satisfying results the demand on development of software making the RDF data available for a data mining processes in particular for processes developed in the RapidMiner environment is evident.

3 Technical Description

The RapidMiner operator proposed in this paper has been developed to query RDF data sources accessible over SPARQL endpoints and to transform the query answers into proprietary RapidMiner format "understandable" for all other operators available in the RapidMiner environment.

The operator has three parameters (Figure 1). The first one specifies the URL of the SPARQL endpoint that has to be queried. The second parameter contains the SPARQL query itself and the third one is to specify variables of the SPARQL query that have to be interpreted as polynominals.

Figure 1: Input parameters for the RDF retrieval operator

To query more than one SPARQL endpoints a federation engine like ELITE [6] can be used. Another option for querying multiple data sources simultaneously is to use SPARQL 1.1's federated queries [14]. Since in practice, formulation of SPARQL queries is a sophisticated process and most of data mining experts aren't expected to have sufficient experience in this field, we

have developed an assistance tool to support non-ontology experts to formulate a SPARQL query. The goal of this tool is to identify data properties of the ontology, which contain the data required to carry out the data mining processes. To do so, the tool displays the ontology structure in a user-friendly manner enabling users to navigate through the ontology classes and properties.

Assume that the TBox of an ontology contains the following statements (namespace specifications are omitted):

$$
\begin{aligned}
&\exists \texttt{hasBuildingGeometry} \sqsubseteq \texttt{Building} \\
&\exists \texttt{hasBuildingGeometry}^{-} \sqsubseteq \texttt{BuildingGeometry} \\
&\exists \texttt{hasNumberOfCompleteStoreys} \sqsubseteq \texttt{BuildingGeometry} \\
&\exists \texttt{hasNumberOfCompleteStoreys}^{-} \sqsubseteq \texttt{NumberOfCompleteStoreys} \\
&\exists \texttt{numberOfCompleteStoreysValue} \sqsubseteq \texttt{NumberOfCompleteStoreys} \\
&\texttt{Range(numberOfCompleteStoreysValue)} \equiv \texttt{xsd:integer} \\
&\exists \texttt{hasNumberOfRooms} \sqsubseteq \texttt{BuildingGeometry} \\
&\exists \texttt{hasNumberOfRooms}^{-} \sqsubseteq \texttt{NumberOfRooms} \\
&\exists \texttt{numberOfRoomsValue} \sqsubseteq \texttt{NumberOfRooms} \\
&\texttt{Range(numberOfRoomsValue)} \equiv \texttt{xsd:integer} \\
&\exists \texttt{hasGroundFloor} \sqsubseteq \texttt{BuildingGeometry} \\
&\exists \texttt{hasGroundFloor}^{-} \sqsubseteq \texttt{GroundFloor} \\
&\exists \texttt{hasGroundFloorArea} \sqsubseteq \texttt{GroundFloor} \\
&\exists \texttt{hasGroundFloorArea}^{-} \sqsubseteq \texttt{GroundFloorArea} \\
&\exists \texttt{groundFloorAreaValue} \sqsubseteq \texttt{GroundFloorArea} \\
&\texttt{Range(groundFloorAreaValue)} \equiv \texttt{xsd:decimal} \\
&\exists \texttt{hasGroundFloorHeight} \sqsubseteq \texttt{BuildingGeometry} \\
&\exists \texttt{hasGroundFloorHeight}^{-} \sqsubseteq \texttt{GroundFloor} \\
&\exists \texttt{groundFloorHeightValue} \sqsubseteq \texttt{GroundFloorHeight} \\
&\texttt{Range(groundFloorHeightValue)} \equiv \texttt{xsd:decimal}
\end{aligned}
\tag{1}
$$

To formulate a query the user begins with determining of a concept by typing its name for instance “Building” in the input box (Figure 2). To support the user, a suggest feature has been implemented as a drop-down list which is activated while the user is typing the input. The drop-down list is filled with ontology concept names that match the user’s input. The user can select one of the matches to retrieve its object properties which are shown in brackets (see Figure 2 below). When the user selects a concept name its annotations properties (e.g. descriptions, comments, and references) are shown. In Turn each object property of the selected concept can be selected to demonstrate property’s ranges (ontology concepts). When selecting the latter one the user gets demonstrated object and data properties of the selected range concept. For example, when the class `BuildingGeometry` is in focus its properties are shown, among others `NumberOfCompleteStoreys`, `NumberOfRooms`, or `GroundFloor`. By clicking one of these object properties, their ranges and further the object properties of that ranges user can navigate through the whole ontology.

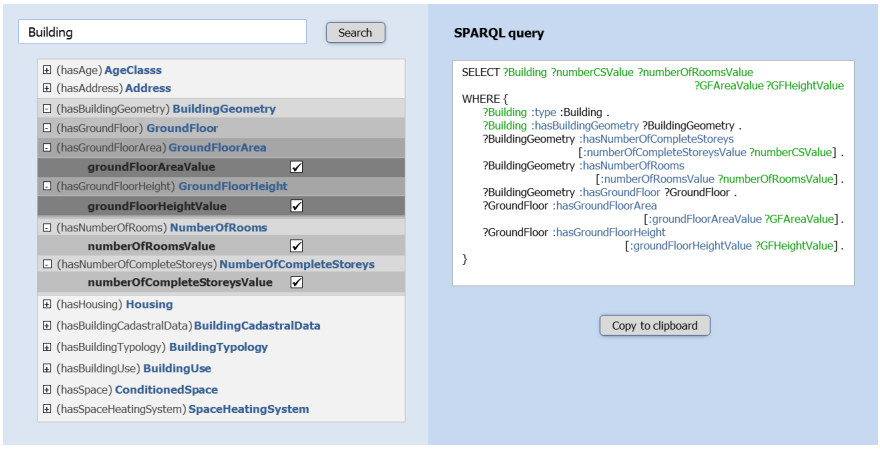

Figure 2: Query generation tool

Moreover the user also can select data properties by activating the corresponding check boxes. The selected data properties are used to generate a SPARQL query. The tool records the paths that the user has been followed by selecting object properties and their ranges. In particular, this path is used to generate the `SELECT` and `WHERE` clauses of the SPARQL query. Finally, the generated query is submitted to the SPARQL endpoint by means of the RapidMiner operator described in this paper to retrieve the desired data.

A SPARQL query to get the data related to the ontology part shown in (1) looks like the following (in the query namespace specifications are omitted):

```
SELECT ?Building ?numberCSValue ?numberOfRoomsValue          (2)
       ?GFAreaValue ?GFHeightValue
WHERE {
   ?Building :type :Building .
   ?Building :hasBuildingGeometry ?BuildingGeometry .
   ?BuildingGeometry :hasNumberOfCompleteStoreys ?NumberCS .
   ?NumberCS :numberOfCompleteStoreysValue ?numberCSValue .
   ?BuildingGeometry :hasNumberOfRooms ?NumberOfRooms .
   ?NumberOfRooms :numberOfRoomsValue ?numberOfRoomsValue .
   ?BuildingGeometry :hasGroundFloor ?GroundFloor .
   ?GroundFloor :hasGroundFloorArea ?GFArea .
   ?GFArea :groundFloorAreaValue ?GFAreaValue .
   ?GroundFloor :hasGroundFloorHeight ?GFHeight .
   ?GFHeight :groundFloorHeightValue ?GFHeightValue .
}
```

After a query has been evaluated by a SPARQL endpoint and the evaluation result has been retrieved, it is transformed into the internal representation of RapidMiner. The data type of each result variable is analyzed and transformed into the corresponding RapidMiner type. For instance Boolean variables are transformed into binominals and URIs of individuals are transformed into strings.

Since polynominal variables can't be identified on the basis of the RDF data type analysis, additional information is required to solve this issue. To identify polynominal variables we use the semantics of data, i.e. particular conceptualization aspects expressed over the ontology's TBox. For this purpose, we have defined a special ontology concept that subsumes all concepts whose individual's URIs should be converted to polynomial variables. Identified polynominals are delivered to the operator using the appropriated operator parameter.

The meta data and the data itself which is received by execution of the SPARQL query mentioned above and transformed into the internal representation of RapidMiner is depicted in Figure 3.

ExampleSet (98 examples, 1 special attribute, 5 regular attributes)

Role	Name	Type	Missings
id	ID	integer	0
regular	Building	text	0
regular	numberCSValue	integer	0
regular	numberOfRoomsValue	integer	0
regular	GFAreaValue	real	0
regular	GFHeightValue	real	0

ExampleSet (98 examples, 1 special attribute, 5 regular attributes)

ID	Building	numberCSValue	numberOfRoomsValue	GFAreaValue	GFHeightValue
0	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/1	1	4	60	3
1	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/10	1	4	61	3
2	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/11	2	4	54	3.500
3	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/12	2	4	40	3
4	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/13	2	5	64	3
5	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/14	2	5	50	3.500
6	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/15	2	5	68	3
7	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/16	2	5	53	3.500
8	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/17	2	5	65	3.500
9	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/18	2	5	68	3
10	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/19	2	5	50	3.500
11	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/2	2	5	39	3
12	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/20	2	5	56	3.500
13	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/21	2	5	56	3.500
14	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/22	2	5	55	3
15	http://arcdev.housing.salle.url.edu/semanco/repository/sap/building/23	2	4	54	3.500

Figure 3: Retrieved data transformed in RapidMiner representation format

Having the data transformed into the internal representation of RapidMiner, the data can be further processed with conventional operators.

To avoid recurrent execution of SPARQL queries, which may be time consuming, we store the query results in a database (Figure 4, Part A). Doing so, the data retrieved from the RDF sources previously can be simply reloaded and processed (Figure 4, Part B).

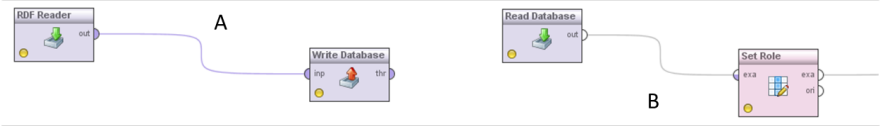

Figure 4: RapidMiner processes to retrieve RDF data and store as well as load the transformed data into a database

4 Conclusion and Further Work

In this paper we have described a RapidMiner operator for retrieval of RDF data distributed in sources exposing a SPARQL endpoint. After retrieving data in RDF format the operator transforms it into the internal RapidMiner representation and so facilitates processing of this data by means of other operators available in the RapidMiner environment.

The described operator has been successfully implemented and is now applied in data mining processes for retrieval of data related to the scope of SEMANCO project (`http://semanco-project.eu`) focusing on carbon reduction in urban planning.

In our future work we purpose to integrate the tools for query generation and ontology analyses mentioned above in the described RapidMiner operator described in this paper.

Furthermore, apart of probabilistic data mining techniques we plan to facilitate data processing using OLAP technologies. To do so issues related to data mart generations need to be addressed, e.g. through exploiting formally specified data semantics for the automation of data cube design, in particular tackling such challenges as selection and combination of data cube dimensions or summarizability problem as already shown by Romero & Abelló [12].

Acknowledgments.

The main contribution of this work has been developed within the SEMANCO project, which is being carried out with the support of the Seventh Framework Programme "ICT for Energy Systems" 2011–2014, under the grant agreement no. 287534.

References

[1] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.

[2] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International journal of human computer studies*, 43(5):907–928, 1995.

[3] Mansoor Ahmed Khan, Gunnar Aastrand Grimnes, and Andreas Dengel. Two pre-processing operators for improved learning from semanticweb data. In *First RapidMiner Community Meeting And Conference (RCOMM 2010)*, volume 20, 2010.

[4] Nada Lavrač, Anže Vavpetič, Larisa Soldatova, Igor Trajkovski, and Petra Kralj Novak. Using ontologies in semantic data mining with segs and g-segs. In *Discovery Science*, pages 165–178. Springer, 2011.

[5] LinkingOpenData W3C SWEO Community Project. `http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`.

[6] Andreas Nolle and German Nemirovski. ELITE: An Entailment-based Federated Query Engine for Complete and Transparent Semantic Data Integration. In *26th International Workshop on Description Logics*, 2013.

[7] Heiko Paulheim and Johannes Fümkranz. Unsupervised generation of data mining features from linked open data. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, page 31. ACM, 2012.

[8] Jedrzej Potoniec and Agnieszka Lawrynowicz. RMonto: ontological extension to RapidMiner. In *Demo session at the International Semantic Web Conference*, 2011.

[9] Jedrzej Potoniec and Agnieszka Lawrynowicz. RMonto-towards KDD workflows for ontology-based data mining. *eCML PKDD 2011*, page 11, 2011.

[10] rapidminer-semweb. `http://code.google.com/p/rapidminer-semweb`.

[11] RMonto: RapidMiner ontological extension. `http://semantic.cs.put.poznan.pl/RMonto/doku.php`.

[12] Oscar Romero and Alberto Abelló. Generating multidimensional schemas from the semantic web. In *Proc. of CAiSE Forum07, Poster*, volume 247, pages 69–72. Citeseer.

[13] Simeon J Simoff and Mary Lou Maher. Ontology-based multimedia data mining for design information retrieval. In *Proceedings of ACSE Computing Congress*, volume 320. Cambridge, MA: ACSE, 1998.

[14] SPARQL 1.1 Federated Query. `http://www.w3.org/TR/sparql11-federated-query`.