FAGI-tr: A tool for aligning geospatial RDF vocabularies

Giorgos Giannopoulos[1], Thomas Maroulis[2], Dimitrios Skoutas[1], Nikos Karagiannakis[1], and Spiros Athanasiou[1]

[1] IMIS Institute, "Athena" Research Center
[2] Imperial College, London

Abstract. In this paper, we present FAGI-tr, a tool for aligning RDF vocabularies with respect to their geospatial aspect. The tool provides a framework for (a) loading a source and a target geospatial RDF dataset, (b) identifying vocabularies for representing geospatial RDF data, (c) selecting, from both datasets, the representations to be considered for processing, (d) selecting a target vocabulary and transforming all geospatial triples from both datasets into the respective format and (e) outputting the two datasets for further processing. The outcome of the process is datasets that follow exactly the same vocabulary and, also, are cleansed from possible duplicate triples containing geospatial metadata, which is the case when an RDF dataset adopts more than one vocabularies to describe spatial data. The tool is tested with DBpedia data and performs rather efficiently.

1 Introduction

The Semantic Web and Linked Data practices have been gaining increasing interest the last years. More and better technologies and tools are becoming available for producing RDF datasets that adhere to common, widely adopted schemata and vocabularies, so that the contained information can be searched and integrated in a more automated and principled manner. However, it is rarely the case that there exists a single, commonly used schema or ontology for a given domain. Often, several overlapping or complementary schemata may have evolved in parallel and be used by different communities. Moreover, users may be unaware of or unwilling to use an existing schema, resorting instead to custom schemas and vocabularies when producing RDF data. This is also the case in the geospatial data domain, where several vocabularies have been proposed and utilized for describing geospatial features in RDF, such as Basic Geo or GeoRSS [4], although GeoSPARQL [8] is lately becoming a more widely accepted standard.

In addition, it is often the case that different data sources, although describing the same real world entities, provide different views of them, either by providing information on different subsets of attributes or even by providing different values on the same attributes. Typical reasons for this is that some sources may be outdated or may serve a different purpose and have different focus. As a result, information for the same real world entities is often spread across several heterogeneous datasets, each one providing partial and/or contradicting views of it, which then need to be fused in order to acquire a richer, cleaner and universal dataset.

In this paper, we focus on fusion of geospatial RDF data and specifically, the first necessary step of the process: the alignment of RDF vocabularies. FAGI-tr, the first component of our envisioned framework for *Fusion and Aggregation of Geospatial Information*, allows the configuration of matching rules that identify different geospatial RDF vocabularies, the efficient application of such rules on RDF datasets and the transformation of the data from one vocabulary to another.

2 Related Work

There are several approaches for transforming conventional data to RDF. Indicatively, some approaches are presented next. However, to the best of our knowledge, this work is the first one addressing RDF-to-RDF transformations on geospatial RDF vocabularies.

In [10], the authors present SPARQL2XQuery, a framework that provides a mapping model for the expression of OWL-RDF/S to XML Schema mappings as well as a method for SPARQL to XQuery translation. Through the framework, XML datasets can be turned into SPARQL endpoints. Sparqlify [3] is a SPARQL-SQL query rewriter that allows the definition of RDF views using a *Sparqlification Mapping Language*. This way, it enables SPARQL queries on relational databases, emphasizing on the LinkedGeoData framework [11] which utilizes Sparqlify to provide access to OpenStreetMap data in RDF form, through SPARQL endpoints and dowloadable data dumps. Finally, TripleGeo [12] is an ETL utility that can extract geospatial features from various sources (shapefiles and DBMSs) and transform them into Basic Geo or GeoSPARQL compatible RDF triples for subsequent loading into RDF stores.

3 Vocabulary Transformations in FAGI

In this Section, we present FAGI-tr (FAGI for transformations) the module of FAGI that handles the recognition of different RDF representations of spatial features in RDF, i.e. different vocabularies, literal (feature values) formats and coordinate reference systems, as well as the transformation of these representations from one to another.

FAGI-tr is implemented in Java, as a desktop application, and provides a graphical user interface. It takes as input SPARQL endpoints from where source and target datasets are loaded and stored into the underlying RDF store. For the latter, we have used Virtuoso[3]. Next, the two datasets are parsed, and preconfigured regular expressions that recognize different RDF representations of triples involving geospatial data are applied. The regular expressions are organized in distinct configuration files, that, currently, need to be manually editer by the user in order to create new matching rules. For each dataset, the identified vocabularies are presented to the user in order to select the types of triples (i.e., the respective vocabularies) that are to be processed further. At the final step, the user selects a target vocabulary (from all the available/defined vocabulary matching rules) and all selected geospatial triples from both datasets are transformed into the respective vocabulary. The output is written either on the same datasets or new datasets can be created, so that the original ones are kept for future use. The source code of FAGI-tr is publicly available, and also available as a jar file for execution [2]. In what follows, we describe in more detail the tool components, the rule matching configuration, and we demonstrate the usage of the tool.

[3] `http://virtuoso.openlinksw.com/`

3.1 Components

FAGI-tr consists of four basic components, described next:

- *GUI component*. It consists of three parts, implementing the user interfaces.
- *CORE component*. This component is responsible for fetching both conventional (non-spatial) triples and identified spatial triples from a source dataset and storing them into Virtuoso. It also handles vocabulary rule matching and provides matching metadata (e.g. number of matched geospatial triples to a specific vocabulary rule).
- *GEOMETRY component*. This component implements all the necessary geospatial functionality. It provides parsing and transformation functions for handling geometry serializations and coordinate reference systems.
- *RULES component*. This component handles the synthesis of vocabulary matching rules in the form of regular expressions and their translation into SPARQL queries to be applied on the RDF datasets both for matching vocabularies and for transforming from one vocabulary to another.

3.2 Supported data sources and formats

Currently, FAGI-tr supports loading data from SPARQL endpoints where the actual endpoint and the graph URI of the dataset are required. The supported RDF triples format is N-triples. The output of the tool is written into the underlying Virtuoso RDF store. As far as RDF vocabularies for geospatial features representation are concerned, we currently have defined rules for three vocabularies: GeoRSS [7], Basic Geo Vocabulary [6], and GeoSPARQL. Implementation of support of RDF files as input/output sources, other RDF triple formats (RDF/XML, Turtle, etc.) and definition of additional vocabulary matching rules are part of ongoing work. We note that, defining new rules is possible by defining proper regular expressions into the configuration files of the tool.

3.3 Configuration rules

Rules for matching and transforming triples are expressed in the form of triple restrictions and are defined in five separate configuration files. We define four types of rules. The first three rule types (property, class, object) are helper rules intended to improve the readability and formulation of full triple rules. The only rules that will be matched are the full triple rules. The values of any helper rule will be substituted into the full triple rule internally by the rule parser. Due to lack of space, we briefly present the four rule types and provide an example of a full configuration rule. Detailed description of the rules syntax can be found in [5].

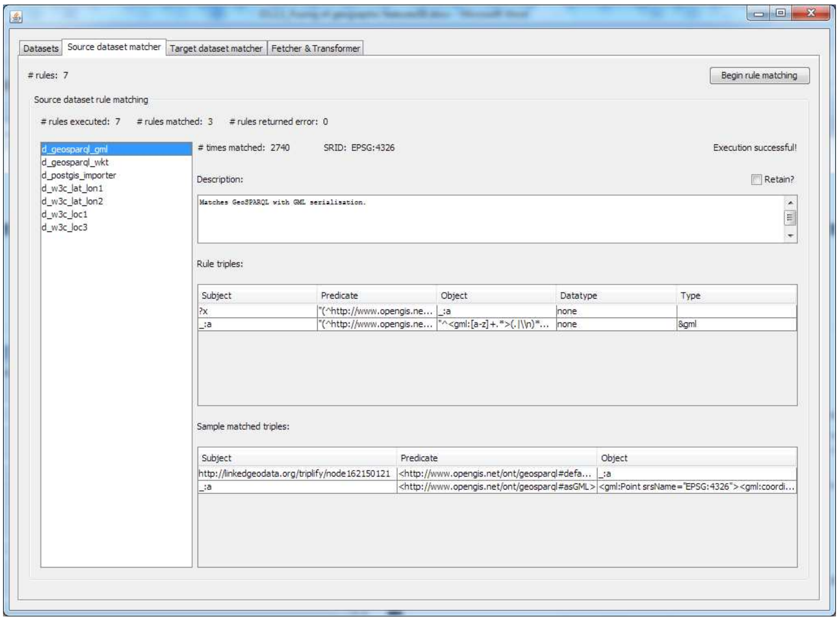
- *property*: Configuration file ’property’ is used for the definition of rules that match RDF properties.
- *class*: Configuration file ’class’ is used for the definition of rules that match classes.
- *object*: Configuration file ’object’ is used for the definition of rules that match object literals.
- *full triple rules*: Configuration files ’triple_default’ and ’triple_user’ are used for the definition of the full triple rules that will be used for matching. These rules reference property, object and class rules. The former contains predefined rules, while the latter contains user added rules. Both files use the same syntax and are handled the same way internally.

An example rule is given below. In the first line, the rule *id*, *description* and *number of triples* are given. Next, the three triple expressions comprising the rule are provided, referring property and object rule ids (corresponding to the respective regular expressions) from the rest configuration files.

```
<k_w3c_loc2> "WGS84 identification rule" 3
?x <p_wgs84_loc> _:a
_:a <p_wgs84_lat> <o_lat>
_:a <p_wgs84_long> <o_long> .
```

3.4 Tool demonstration

In the first step, the user specifies the datasets to be processed and where the results are to be stored. For both source and target datasets, the SPARQL endpoint and the graph URI of the dataset are required. Then, the rule matching process is executed on both datasets. This is illustrated in Figure 1. The left panel displays all available rules. Upon selecting one of them, the user can see whether the rule was matched and, if so, with how many triples. Also, information for the matched rule is presented, including its description, the structure of the rule, and a sample matching set of triples from the dataset. The user is able to select which kind of triples to retain for further processing, e.g. to retain only the triples matching a specific vocabulary rule. This allows the user to keep only certain vocabulary versions of the geospatial triples, saving processing effort for the next steps, as well as clearing out possible erroneously contained triples.



rules: 7

Source dataset rule matching

rules executed: 7   # rules matched: 3   # rules returned error: 0

d_geosparql_gml
d_geosparql_wkt
d_postgis_importer
d_w3c_lat_lon1
d_w3c_lat_lon2
d_w3c_loc1
d_w3c_loc3

times matched: 2740   SRID: EPSG:4326   Execution successful!

Description: □ Retain?

Matches GeoSPARQL with GML serialisation.

Rule triples:

Subject	Predicate	Object	Datatype	Type
?x	"(^http://www.opengis.ne…	_:a	none	
_:a	"(^http://www.opengis.ne…	"^<gml:[a-z]+.*>(.|\\n)*…	none	&gml

Sample matched triples:

Subject	Predicate	Object
http://linkedgeodata.org/triplify/node162150121	<http://www.opengis.net/ont/geosparql#defa…	_:a
_:a	<http://www.opengis.net/ont/geosparql#asGML>	<gml:Point srsName="EPSG:4326"><gml:coordi…

Fig. 1. Rule matching panel

Rule matching statistics for the source and target datasets are presented in a separate panel, the fetching and transforming panel. This allows to choose a target vocabulary rule, so that all retained geospatial triples from both datasets are transformed according to the vocabulary specified by the rule. The transformed triples, along with the unmodified non-spatial triples are then written into the RDF store, in different output graphs, depending on the user selection on the dataset loading panel. There is also limited support for changing the CRS, which will be extended in the future. A video that demonstrates FAGI-tr is provided in the link below (copy-paste the link to your browser).

`http://web.imis.athena-innovation.gr/%7egiann/FAGI-tr.mp4`

3.5 Evaluation

The correctness of the transformation process has been verified by examining all possible vocabulary transformations and the triples produced. Thus, our evaluation focused on assessing the efficiency of the tool, that is, the total time required to match geospatial vocabularies, as well as to transform triples from one vocabulariy to another. We used a DBpedia dataset[4] containing 2M triples, of which 1M triples corresponded to 500K geometries in Basic Geo vocabulary. The task was to transform these triples into the GeoSPARQL format. We measured the time required for the following subprocesses run withing FAGI-tr: (a) Vocabulary rules matching, (b) Fetching of non-spatial triples, (c) Loading of matched spatial triples, (d) Transformation of matched spatial triples. We can see (Table 1) that the whole process requires less than a minute to run.

Process	Time (sec)
Rule matching	3.524
Non-spatial metadata fetching	4.782
Spatial triples loading	23.393
Spatial triples transformation	23.596

Table 1. Run times for FAGI-tr functions.

4 Conclusion

In this paper, we presented FAGI-tr, a tool for vocabulary transformations that focuses on matching and transforming geospatial RDF vocabularies. We presented the system's components and functionality, we assessed its efficiency and showcased its usage. To the best of our knowledge this is the first tool that specializes on aligning geospatial RDF vocabularies. Our next steps involve enriching the input and output formats of the tool, as well as increasing the tool's scalability and efficiency.

Acknowledgments This work was supported by a grant from the EU's 7th Framework Programme (2007-2013) provided for the project GeoKnow (GA no. 318159).

References

1. Bleiholder, J. and Naumann, F. Declarative data fusion - syntax, semantics, and implementation. In *Proc. of the 9th East European conference on Advances in Databases and Information Systems*, pp. 58-73, 2005.
2. FAGI-tr. Available at https://github.com/GeoKnow/FAGI-tr
3. Sparqlify. Available at https://github.com/AKSW/Sparqlify
4. GeoKnow EU/FP7 project. Market and Research Overview. Available at http://svn.aksw.org/projects/GeoKnow/Public/D2.1.1_Market_and_Research_Overview.pdf
5. GeoKnow EU/FP7 project. Fusing of geographic features. Available at http://svn.aksw.org/projects/GeoKnow/Public/D3.2.1_Fusing_of_geographic_features.pdf
6. Basic Geo (WGS84 lat/long) Vocabulary. Available at http://www.w3.org/2003/01/geo/
7. Open Geospatial Consortium. An Introduction to GeoRSS. Whitepaper, 2006
8. Open Geospatial Consortium Inc. OGC GeoSPARQL standard - A geographic query language for RDF data. Available at https://portal.opengeospatial.org/files/?artifact_id=47664
9. PostGIS - Spatial and Geographic objects for PostgreSQL. Available at http://postgis.net/
10. Bikakis, N. and Tsinaraki, C. and Stavrakantonakis, I. and Gioldasis, N. and Christodoulakis, S. The SPARQL2XQuery interoperability framework. In *World Wide Web Journal*, pp. 1-88, 2014.
11. Stadler, C. and Lehmann, J. and Höffner, K. and Auer, S. LinkedGeoData: A Core for a Web of Spatial Open Data. In *Semantic Web Journal*, v. 3:4, pp. 333-354, 2012.
12. Patroumpas, K. and Alexakis, M. and Giannopoulos, G. and Athanasiou, S. TripleGeo: an ETL Tool for Transforming Geospatial Data into RDF Triples. In *Proc. of LWDM'14, EDBT/ICDT Workshops*, 2014 (to appear) - http://www.dblab.ece.ntua.gr/pubs/uploads/TR-2014-2.pdf.

[4] `http://downloads.dbpedia.org/3.9/en/geo_coordinates_en.nt.bz2`