Morph-CSV: Virtual Knowledge Graph Access for Tabular Data

David Chaves-Fraga[1], Luis Pozo-Gilo[1], Jhon Toledo[1],
Edna Ruckhaus[1], and Oscar Corcho[1]

Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
dchaves@fi.upm.es, luis.pozo@upm.es, ja.toledo@upm.es,
eruckhaus@fi.upm.es, ocorcho@fi.upm.es

Abstract. Virtual knowledge graph access has traditionally focused on providing ontology-based access to relational databases (RDB) proposing SPARQL-to-SQL query translation techniques and optimizations. With the advent of mapping languages or annotations such as RML or CSVW, these techniques have been applied over tabular data by considering each source as a single table that can be loaded into an RDB. However, such techniques do not take into account those characteristics that are normally present in real-world CSV files (e.g., normalization, constraints, joins). In this paper we present Morph-CSV, a framework for enhancing virtual knowledge graph access over a set of CSV files by using a combination of CSVW annotations and RML mappings with FnO transformation functions. Exploiting these inputs, the framework creates an enriched RDB representation of the CSV files together with the corresponding R2RML mappings, enabling the use of existing query translation (SPARQL-to-SQL) techniques and tools.

Keywords: Knowledge Graphs · CSV · RML · CSVW

1 Introduction

Semi-structured data formats, and particularly spreadsheets in the form of CSV or Excel files, are one of the most widely-used formats to publish data on the Web. There are several reasons why tabular formats are so popular for data publication. First, they are easy to generate by data providers. In many cases, they are even used as one of the main ways to manage data inside organizations. Second, they are easy to consume with common office tools (e.g., Excel, LibreOffice) and there are advanced tools that can be used to process them (e.g., OpenRefine, Tableau). However, more advanced consumers (e.g., application developers, knowledge workers) often have to face some relevant challenges when consuming tabular data: there is no standard way to query data in them as it can be done with other types of data formats, such as RDB, JSON or XML; data are difficult to integrate since data constraints and relationships across different files are not explicit; data are often difficult to understand since column names are generally heterogeneous.

Some of these challenges may be dealt following a Semantic Web approach. Virtual knowledge graphs (VKG) provide a unified view and common access to a set of data sources based on ontologies and mappings, translating SPARQL queries into queries that are supported by the underlying source. Although current proposals [7] provide support for querying this kind of formats, they treat each source as if it was a single not-normalized RDB table with no keys or integrity constraints, important elements that are used by SPARQL-to-SQL engines for efficient querying. Several languages have been proposed to specify annotations to deal with the heterogeneity of tabular datasets such as CSVW [8] metadata and RML+FnO [5] mapping rules, but engines or systems have to take them into account in their VKG access pipeline.

In this demo we present Morph-CSV, an open source engine[1] that extends the typical VKG workflow to enhance performance and query completeness over tabular datasets. Our approach exploits the information from CSVW annotations and RML+FnO mappings so as to obtain details on the underlying schema, required transformation functions, missing information, etc., pushing down their application directly over the tabular dataset. It generates and populates an enriched and normalized RDB schema from the CSV files, and translates RML+FnO to an equivalent function-free R2RML mapping document [4], so that existing SPARQL-to-SQL optimizations can be used to query them. Finally, we describe two real use cases from transport and biomedical domains where Morph-CSV is applied to enhance virtual KG access.

2 Tabular Annotations for VKG: RML+FnO and CSVW

There are specific challenges on querying tabular datasets using a VKG access approach that have not been tackled by existing techniques. The selection of the sources to answer a query, the normalization or heterogeneity of the dataset and the absence of indexes affect the performance and completeness of SPARQL-to-SQL engines. To deal with these challenges, RML [6] extends the R2RML W3C Recommendation to provide support beyond relational databases, such as XML, CSV, JSON, etc. Recently, RML has been integrated with the Function Ontology (FnO) to support other types of transformations [5]. Additionally, CSVW annotations [8] is a W3C Recommendation that provides metadata annotations for tabular data on the web. In Table 1 we summarized the properties of these two specifications and its related challenge(s). The manual and ad-hoc preparation of a tabular dataset for VKG access is usually the most time-consuming and less reproducible task. Exploiting available standard and declarative annotations allows its generalization and automatization, as well as ensuring query completeness and improving performance of SPARQL-to-SQL techniques.

[1] https://doi.org/10.5281/zenodo.3572132

Table 1. Properties of CSVW, and RML+FnO that can be used to address the challenges of dealing with tabular data in construction virtual knowledge graphs

Challenges	**Relevant Properties**
Describe the corresponding concept	rr:class, csvw:propertyUrl
Describe the corresponding property	rr:predicateMap, csvw:propertyUrl
Add header to a file	csvw:rowTitles
Column datatype	csvw:datatype
Constraining values	csvw:minimum, csvw:maximum
Specify the format of a column	csvw:format
Specify a join	rr:refObjectMap, csvw:foreignKeys
Transform value	fnml:functionValue
Support for multiple values in one cell	csvw:separator
Primary key	csvw:primaryKey
Default for missing values	csvw:default
Specify NULL values	csvw:null
Specify NOT NULL constraint	csvw:required
Specify columns to be tranformed	rr:reference, rr:template

3 The Morph-CSV engine

The Morph-CSV[2] open source engine exploits the typical inputs of a VKG process (query, metadata and mappings) to improve performance and query completeness over tabular sources, dealing with their identified challenges. More in detail, it extends the starting phase of a typical VKG access workflow to select the relevant sources from an input query, extract implicit constraints from RML+FnO [5] mappings and CSVW [8] metadata, pushing down their application directly to the selected sources and finally, it generates enriched inputs for a SPARQL-to-SQL process (R2RML mappings and an RDB instance). The architecture of Morph-CSV is shown in Figure 1, where we present the steps to exploit declarative annotations for enhancing SPARQL query translations over tabular data: i) **Source Selection:** Using the SPARQL query and the mapping rules, the engine selects only the relevant sources (and columns inside each source) that are relevant to answer the input query. ii) **Normalization:** Two functions for performing data normalization were implemented. The first one is the treatment of multi-values in columns while the second one is the treatment of multiple entities in the same source. iii) **Data Preparation:** In this step, three different functions are executed. First, it performs all the substitutions such as default values, NULL values and date formats, then, it creates a new column in the specific source applying the transformation function defined in RML+FnO and finally, the engine removes all duplicates in the raw data. iv) **Mapping Translation:** The mapping rules are translated accordingly to the generated

[2] `https://morph.oeg.fi.upm.es/tool/morph-csv`

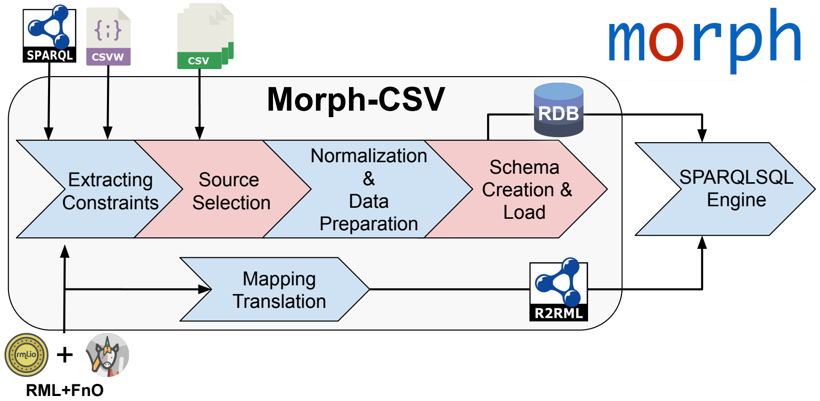

Fig. 1. Proposed workflow to enhance VKG access over tabular data

data from RML+FnO to a standard R2RML document [4]. v) **Schema Creation and Load:** An optimized SQL schema is generated applying integrity constraints (PK-FK), and the selected data sources are loaded.

4 Use Cases

In this demo we run Morph-CSV over two real use cases:

1. **Transport National Access Points (NAP).** Since 2019, most European countries are required to public transport data in accessible open query points called National Access Points or NAP[3]. The main issues related to access to transport data across Europe, will be how to deal with the heterogeneity of these access points and data formats, and how to efficiently query them. Using the *de-facto* standard for publishing open transport data, GTFS[4], which is composed by a set of tabular sources, our engine will exploit RML+FnO and CSVW annotations to enable efficient and complete access to GTFS data through SPARQL queries.
2. **Virtual KG over Bio2RDF.** Bio2RDF [1] is one of the most popular projects that integrates and publishes biomedical datasets using Semantic Web technologies. Although its community has actively contributed to the generation of these datasets, they perform the integration using ad-hoc programming scripts, which negatively affects the maintainability of the project, therefore, SPARQL queries may return outdated results. Selecting the tabular original sources of Bio2RDF, Morph-CSV constructs a virtual KG over them following a declarative approach, hence, improving the maintainability of the project and ensuring up to date results from the SPARQL queries.

[3] https://ec.europa.eu/transport/themes/its/road/action_plan/nap_en

[4] https://developers.google.com/transit/gtfs

The obtained results are shown in a landing page[5] and in a video[6]. Besides the real use cases, we present the results obtained in terms of performance and completeness with Morph-CSV, using two virtual knowledge graph benchmarks from the state of the art (BSBM [2] and GTFS-Madrid-Bench [3]), and two well known open source SPARQL-to-SQL engines (Morph-RDB and Ontop).

5 Conclusions and Future Work

Morph-CSV enhances virtual knowledge graph access over heterogeneous CSV files. It takes as input a set of CSV files, CSVW annotations, and an RML+FnO mapping, and generates as output an enriched RDB instance with data from the CSV files together with R2RML mappings, so that they can be used by any state-of-the-art R2RML-compliant OBDA engine. As part of our future work, we will improve its performance with new optimizations in the query-translation process. We will also extend it for other types of data (e.g., XML, JSON).

Acknowledgements. The work presented in this paper is supported by the Spanish Ministerio de Economía, Industria y Competitividad and EU FEDER funds under the DATOS 4.0: RETOS Y SOLUCIONES - UPM Spanish national project (TIN2016-78011-C4-4-R) and by an FPI grant (BES-2017-082511)

References

1. Belleau, F., Nolin, M.A., Tourigny, N., Rigault, P., Morissette, J.: Bio2RDF: towards a mashup to build bioinformatics knowledge systems. Journal of biomedical informatics **41**(5), 706–716 (2008)
2. Bizer, C., Schultz, A.: The Berlin SPARQL Benchmark. International Journal on Semantic Web and Information Systems (IJSWIS) **5**(2), 1–24 (2009)
3. Chaves-Fraga, D., Priyatna, F., Cimmino, A., Toledo, J., Ruckhaus, E., Corcho, O.: GTFS-Madrid-Bench: A Benchmark for Virtual Knowledge Graph Access in the Transport Domain. Journal of Web Semantics **65** (2020)
4. Corcho, O., Priyatna, F., Chaves-Fraga, D.: Towards a new generation of ontology based data access. Semantic Web **11**, 153–160 (2020)
5. De Meester, B., Maroy, W., Dimou, A., Verborgh, R., Mannens, E.: Declarative data transformations for Linked Data generation: the case of DBpedia. In: European Semantic Web Conference. pp. 33–48. Springer (2017)
6. 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: LDOW (2014)
7. Priyatna, F., Corcho, O., Sequeda, J.: Formalisation and experiences of R2RML-based SPARQL to SQL query translation using morph. In: Proceedings of the 23rd international conference on World wide web. pp. 479–490 (2014)
8. Tennison, J., Kellogg, G., Herman, I.: Model for tabular data and metadata on the web. W3C recommendation. World Wide Web Consortium (W3C) (2015)

[5] `https://morph.oeg.fi.upm.es/demo/morph-csv`

[6] `https://youtu.be/yzskzFSAMzA`