RICDaM: Recommending Interoperable and Consistent Data Models

Daniela Oliveira and Mathieu d'Aquin

Data Science Institute, Insight SFI Research Centre for Data Analytics, NUI Galway
{first,last}@insight-centre.org

Abstract One of the core functionalities of Knowledge Graphs is that data is not required to adhere to strictly defined data models. Nonetheless, the RDF model provides specifications for publishing data on the Web that not only describe entities and relationships but also focus on enabling interoperability between datasets. However, finding the right ontologies to model a dataset is a challenge since several valid data models exist and there is no clear agreement between them. We present a demonstration of an interface that allows users to customise a data model based on recommendations obtained with the RICDaM framework. This framework focuses on ranking candidates based on three metrics that measure the relevancy of the candidate, the interoperability of the neighbourhood of the candidate, and the overall consistency of the model proposed. The interface allows the user to refine the recommended data model and complete it when the framework is lacking or not directly fitting the intention of the user.
The demo can be found at http://afel.insight-centre.org/ricdam/

Keywords: Knowledge Graph · Ontologies · Linked Open Data.

1 Introduction

Data models have an important role in data integration on the Web because they define how data is connected and stored. The RDF data model uses subject-predicate-object statements (i.e., triples) to model datasets. The data model is usually supported by one or more ontologies that provide standard nomenclature to describe concepts. From the ontologies, classes are used to annotate the entity types of subjects and objects in an RDF dataset, while datatype and object properties are used to model predicates. A survey conducted over several years with Linked Data providers found that the third most common barrier to publishing Linked Data was *Selecting appropriate ontologies to represent our data* [8]. If data publishers cannot find appropriate ontologies to model their data, they tend to create their own ontologies or extend upper-level ontologies to meet their requirements. Therefore, knowledge from similar domains ends up following different standards or data models that not always focus on interoperability with other datasets in the same domain, making it challenging to integrate data from multiple, existing knowledge graphs, as well as to model new data consistently. This issue can be found in different domains [2, 6].

We propose the RICDaM framework (Recommending Interoperable and Consistent Data Models) which produces a ranked list of candidate data models that not only fit the data but are also interoperable with a Knowledge Graph of multiple published RDF data sources. The output of the framework is the correspondence between a list of triple patterns (i.e., domain, property, range triple) from an input dataset and a ranked list of candidate triple patterns. We exploit the content and graph structure of this Knowledge Graph to compute scores that consider the accuracy, interoperability, and consistency of the candidates. These scores are combined into a single score that is weighted according to the user's preferences or use-case. In this demo, we present an interface that explores the use-case of aligning two datasets in the library data domain with published RDF library data models. The demonstrator shows the output of the framework and allows the user to choose the weights of the scores and manually customise the automatic recommendations generated by the framework.

The problem of creating a data model for a dataset is related to schema matching, which maps relationships and concepts. A variety of schema matching solutions have been proposed for different types of data (e.g., [7, 5, 4, 1]) with the purpose of integrating heterogeneous datasets within a domain. Mapping languages are also popular to align heterogeneous data sources with the RDF data model. The RML [3] is an example of a mapping language that enables the conversion of heterogeneous data formats to RDF. Contrary to these approaches, our framework recommends data models based on existing Knowledge Graphs by focusing not only on accuracy but also boost candidates that are the most interoperable within the graph.

2 Framework Overview

The RICDaM framework, illustrated in Figure 1, has three stages: (1) Building Background Knowledge, (2) Candidate Generation, and (3) Candidate Ranking.

2.1 Building Background Knowledge

The goal of this stage is to build the background knowledge structures that will facilitate the next stages of the framework. The first task includes creating a single KG from multiple existing RDF datasets. The entities of the KG are collected in a document store and indexed using an inverted index to enable full-text search. These entities maintain their original predicates but are also connected to external entities via the underlying ontology graph that facilitates candidate ranking. We enrich this ontology graph by inferring missing information (e.g., entity type of `owl:SameAs` relationships) and via ontology matching.

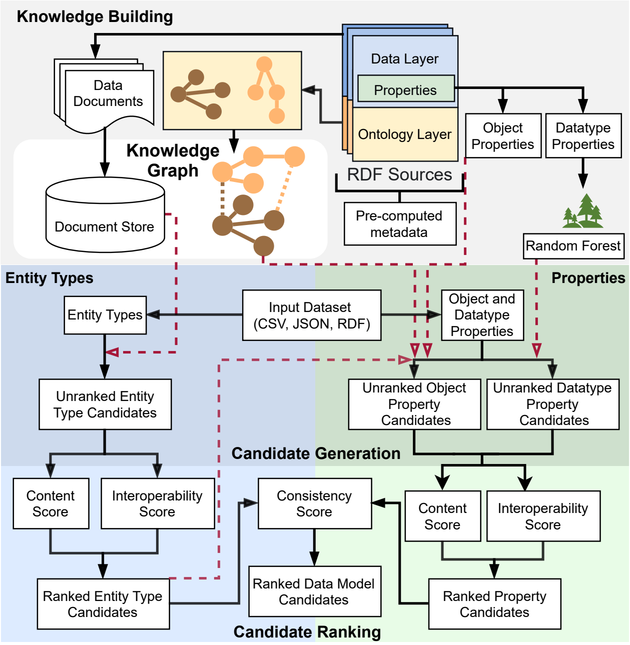

Figure 1. Workflow diagram.

In a final step, we extract metadata from the document store (e.g., entity type frequency), and we train a Random Forest model on datatype property values of the KG to facilitate the generation of datatype property candidates.

2.2 Candidate Generation and Ranking

This stage generates a list of entity type and property candidates for an input dataset. For the entity types, the candidate list is obtained by searching literals in the inverted index of the document store. Datatype properties are obtained with the Random Forest classification model, which produces datatype property predictions for input literal values. Object properties are inferred from the relationships between the entity type candidates in the ontology graph.

We propose to rank candidates using a Content score and an Interoperability Score. The Content Score combines metrics based on string similarity, search results frequency, and graph distances. This score assesses the appropriateness of a candidate to match an entity type or property from an input dataset The Interoperability score contains information from the document store and from a sub-graph that restricts edges to relationships of equivalence, subsumption, or relatedness extracted via ontology matching techniques. This score measures how interoperable a candidate is considering the frequency of a candidate in the KG and the graph neighbourhood of the candidate. Higher Interoperability scores translate into a more connected and relevant candidate. This score uses the *interoperability metric* and the *neighbourhood size*. The neighbourhood size represents the total number of related neighbours up to a distance in the graph, while interoperability metric counts the frequency of these neighbours in the document store. In contrast with the interoperability metric, the neighbourhood size rewards neighbours that do not appear in the RDF data sources but are still connected to the candidate in the ontology graph.

Until now, candidates are ranked independently of each other. Therefore, the final metrics compute the Consistency score. This score increases the likelihood of the same candidate being suggested for the same input entity type or property and, at the same time, boosts triples that are more commonly encountered in the Knowledge Graph. This consistency is achieved by ensuring that triples that appear together (co-occur) in the KG are rewarded, while also guarantees that the same triple elements are assigned the same entity types or properties throughout the data model. Therefore, the consistency combines Content, Interoperability, and co-occurrence frequencies into a single score that ranks candidate data model triples in terms of their adequacy and interoperability.

3 Demonstration

This demonstration presents the top candidate data model proposed to the user in response to an input dataset in the library domain. The interface gives an overview of the best ranked candidates for each triple but also allows the user to adapt the data model to their preference and use-case.

The main functionalities included in the demo are (1) an overview of the output of the framework, (2) customisation of the data model, (3) tuning of the parameters to produce different candidate rankings, and (4) exporting the data model as a set of mappings between the input and the produced data model.

When the user makes a manual change to the data model, they can choose to propagate that change to maintain consistency across the dataset or keep the change locally to the modified cell. The tuning parameters allow the user to customise the ranking of the candidates, obtaining different top data models that can speed up the modelling process by suggesting the candidates the user is looking for more easily. Finally, the user can export the data model produced and apply the mappings to translate their original input data to an RDF dataset that is potentially more interoperable with the datasets in the KG.

Table 1 presents the datasets used in the demo. The dashed line separates datasets used to build the background KG (top) and datasets used as input (bottom). Overall, the datasets contain a variety of records such as books, audio records, and periodicals. For example, for `pgterms:ebook` in Project Gutenberg, the suggestion is to use `bibo:Document` as the most interoperable, relevant, and consistent entity type, together with properties such as `dcterms:contributor`. Through the interface, it can be changed, for example, to `schema:Book`, which is often used as well in the background KGs.

Table 1. Summary of the source RDF libraries chosen.

Library name	Handle	URL	# Entities
British Library	British	`https://www.bl.uk`	17 289 195
Bibliothèque Nationale de France	French	`https://www.bnf.fr`	38 100 563
Deutsche Nationalbibliothek	German	`https://www.dnb.de`	50 340 156
Biblioteca Nacional de Portugal	Portuguese	`http://www.bnportugal.gov.pt`	2 437 096
Biblioteca Nacional de España	Spanish	`http://www.bne.es`	20 752 087
Project Gutenberg (RDF)	Gutenberg	`https://www.gutenberg.org`	856 476
Open Library (JSON)	OpenL	`https://openlibrary.org`	54 678 367

4 Conclusion

In general, designing a data model is not a trivial task. When considering integration with existing datasets, the task becomes more complex. Our demo facilitates the task of finding the best possible data model according to certain criteria by producing a ranked list of candidates to match entity types and properties in a dataset. In the future, a complete tool using this framework would allow the user to search the indexed ontologies or add new ontologies to the graph to complete the model when the data model fails to find the desired class. For JSON or CSV input datasets, it would also be possible to generate a RML mapping file to facilitate the process of translating the data to RDF.

References

1. Alserafi, A., Abelló, A., Romero, O., and Calders, T.: Keeping the Data Lake in Form: Proximity Mining for Pre-Filtering Schema Matching. ACM Trans. Inf. Syst. 38(3), 26:1–26:30 (2020)
2. d'Aquin, M., Adamou, A., and Dietze, S.: Assessing the educational linked data landscape. In: Proceedings of the 5th Annual ACM Web Science Conference. WebSci '13, pp. 43–46. Association for Computing Machinery, Paris, France (2013)
3. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., and Van de Walle, R.: RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In: LDOW (2014)
4. Ermilov, I., and Ngomo, A.-C.N.: TAIPAN: Automatic Property Mapping for Tabular Data. In: Knowledge Engineering and Knowledge Management, pp. 163–179. Springer International Publishing, Cham (2016)
5. Limaye, G., Sarawagi, S., and Chakrabarti, S.: Annotating and searching web tables using entities, types and relationships. Proc. VLDB Endow. 3(1), 1338–1347 (2010)
6. Park, H., and Kipp, M.: Library Linked Data Models: Library Data in the Semantic Web. Cataloging & Classification Quarterly 57(5), 261–277 (2019)
7. Pei, J., Hong, J., and Bell, D.: A Novel Clustering-Based Approach to Schema Matching. In: Advances in Information Systems. LNCS, pp. 60–69. Springer, Heidelberg (2006)
8. Smith-Yoshimura, K.: Analysis of 2018 International Linked Data Survey for Implementers. The Code4Lib Journal (2018)