FedSDM: Semantic Data Manager for Federations of RDF Datasets

Kemele M. Endris$^{1,2[0000-0001-9040-9421]}$,
Maria-Esther Vidal$^{2,1[0000-0003-1160-8727]}$, and
Sören Auer$^{1,2[0000-0002-0698-2864]}$

[1] L3S Research Center, Germany {endris, vidal,auer}@L3S.de
[2] TIB Leibniz Information Centre for Science and Technology, Germany

Abstract. Linked open data movements have been followed successfully in different domains; thus, the number of publicly available RDF datasets and linked data based applications have increased considerably during the last decade. Particularly in Life Sciences, RDF datasets are utilized to represent diverse concepts, e.g., proteins, genes, mutations, diseases, drugs, and side effects. Albeit publicly accessible, the exploration of these RDF datasets requires the understanding of their main characteristics, e.g., their vocabularies and the connections among them. To tackle these issues, we present and demonstrate FedSDM, a semantic data manager for federations of RDF datasets. Attendees will be able to explore the relationships among the RDF datasets in a federation, as well as the characteristics of the RDF classes included in each RDF dataset (`https://github.com/SDM-TIB/FedSDM`).

1 Introduction

As the RDF data model continues gaining popularity, publicly available RDF datasets are growing in terms of number and size [2,6]. One of the challenges emerging from this trend is how to efficiently and effectively execute queries over a set of autonomous RDF datasets, i.e., a federation of RDF datasets. RDF datasets in a federation are accessible via web services, e.g., SPARQL endpoints, and a federated query processing engine enables the execution of queries over these web services. Federated query engines are responsible of selecting the relevant sources of a query, as well as of the tasks of query planning and execution, both required to collect the data from the selected sources and to answer the query [9]. Existing federated SPARQL query engines include MULDER [5], ANAPSID [1], FedX [8], SPLENDID [7], and SemaGrow [3]. Albeit effective, a federated query engine requires user queries which should be expressed in terms of the vocabularies used in the sources of a federation, as well as respecting connections among them. Consider the SPARQL query in Listing 1.1, that collects the mutations of the type `'confirmed somatic variant'` located in transcripts which are translated as proteins that are transporters of the drug `Docetaxel`. To answer this query, a federated query engine should select two data sources, IASIS-KG and DrugBank. But for someone who does not have knowledge about the RDF vocabularies of these RDF datasets, writing this query may require a great effort. We tackle the problem of exploring RDF datasets in a federation, and present and demonstrate FedSDM, a semantic data manager for federations of RDF datasets. FedSDM relies on RDF Molecule Templates (RDF-MTs) [5], abstract representations of RDF classes in the RDF datasets of a federation, and their connections. FedSDM enables the exploration of RDF-MTs; during the demonstration, attendees will observe how an RDF-MT based analysis allows for the understanding of the concepts represented in a federation, as well as the main characteristics of a federation RDF datasets.

Listing 1.1: SPARQL Query

```
SELECT DISTINCT ?mutation ?transcript
WHERE {?mutation      rdf:type iasis:Mutation .
      ?mutation     iasis:mutation_somatic_status           'Confirmed␣somatic␣variant'.
      ?mutation     iasis:mutation_isLocatedIn_transcript   ?transcript .
      ?transcript   iasis:translates_as                     ?protein .
      ?drug         iasis:drug_interactsWith_protein        ?protein .
      ?protein      iasis:label                             ?proteinName .
      ?drug         iasis:label                             'docetaxel' .
      ?drug         iasis:externalLink                      ?drug1 .
      ?drug1        drugbank:transporter                    ?transporter .
      ?transporter  drugbank:gene-name                      ?proteinName .}
```

2 The FedSDM Architecture

The FedSDM architecture includes four basic components: Metadata Manager, Metadata Explorer, Graph Analyzer, and Federated Query Engine.
Metadata Manager: is responsible for creating and managing RDF-MTs in a federation. Given a set of RDF data sources, the metadata manager creates RDF-MTs for each data source. An RDF-MT *rm* is described in terms of: the RDF class of *rm*, cardinality, set of predicates and the cardinality of each predicate, and the links to other RDF-MTs in the federation or in the same RDF dataset. The set of predicates and the links will be used to either analyze the federation or to formulate a federated query. Intra-dataset links (i.e., links between RDF-MTs within the same data source) and inter-dataset links (i.e., links between RDF-MTs from different data sources) in a federation will be exploited by other components, such as a graph analyzer to compute the graph properties of an RDF-MT network, and the federated query engine for decomposition, source selection, and planning of a federated query. **Metadata Explorer:** uses RDF-MTs created by the metadata manager to generate different visualizations. For instance, it analyzes RDF-MT links to visualize the connectivity among datasets. In addition, it acts as a gateway to access the metadata stored for further analysis of the data sources, e.g., cardinality and predicates. **Graph Analyzer:** performs graph analysis of a graph created by using intra- and inter-dataset links among RDF-MTs. Properties such as graph density, number of connected components, transitivity, and clustering coefficient of an RDF-MT graph are generated using networkx[3] python library. **Federated Query Engine:** provides a unified view of the data sources in the federation. This component exploits the metadata

[3] https://networkx.github.io/

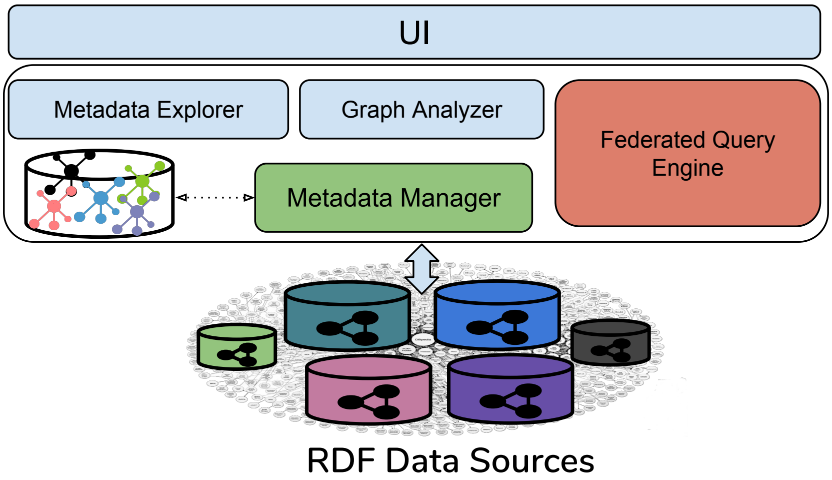
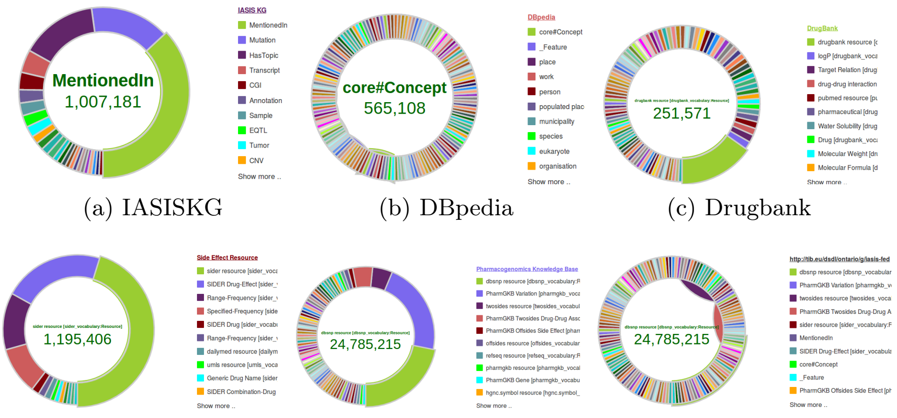


Fig. 1: **FedSDM Architecture**. Given a set of data source endpoints by the user, the metadata manager creates the RDF-MTs from each endpoint and store it to a triple store. Metadata Explorer and Graph Analyzer issues a SPARQL query to collect basic information about RDF-MTs in the federation to perform analysis and present to user via UI component. Finally, given a SPARQL query the Federated Query Engine selects relevant data sources in the federation and execute a federated query then predent the results to the user via the UI.

collected from the RDF datasets in a federation for decomposition and source selection. In FedSDM, MULDER [5] is integrated as the federated query engine.

3 Demonstrating Use Cases

We created a federation composed of five data sources; DBpedia (3.5.1), DrugBank (Bio2RDF), PharmGKB (Bio2RDF), Sider(Bio2RDF), and the IASIS-KG. Attendees will be able to explore the RDF-MTs of these RDF datasets and their connections. Specifically, we will demonstrate the following use cases:
Analysis of Datasets in a Federation. We will present analysis of datasets in the federation in terms of RDF-MT connectivity within a dataset and with other data sources in the federation. First, we will show the RDF-MT composition in different levels, per data source, e.g., Fig. 2a, 2b, 2c, 2d and 2e show concepts in IASIS-KG, DBpedia (3.5.1), Drugbank(Bio2RDF), Sider and PharmGKB, respectively. In addition, the federation in terms of RDF-MTs is depicted in Fig 2f. This gives the idea on how many concepts represented in a dataset and the number of unique entities per concept. Then, we will show the connectivity among sources via RDF-MTs as a force graph (e.g., Fig. 3) and circular(e.g., Fig. 4), demonstrating the connectivity among the RDF-MTs in the federation. Finally, we will show the graph properties, in numbers, of each data source and overall federation. Table 2 shows graph property values, such as



(a) IASISKG (b) DBpedia (c) Drugbank

(d) SIDER (e) PharmGKB (f) IASIS-Fed

Fig. 2: RDF-MT composition of data sources. Each colored portions represents an RDF-MT proportional to total distinct molecules in them.

Data Source	Nodes	Links	Density	C.C	Avg. Clus	Transitivity	Avg. Neighbours
IASIS-KG	31	36	0.07741	5	0.18817	0.265822	2.3225
DBpedia	467	8124	0.07466	20	0.12097	0.05968	34.79229
DrugBank	207	353	0.01655	22	0	0	3.4106
PharmaGKB	181	273	0.01675	39	0	0	3.01657
SIDER	27	43	0.12250	5	0	0	3.18518
ALL(Fed)	767	8698	0.0296	67	0.0811	0.0585	22.6805

Table 2: IASIS-Federation RDF-MTs Graph Properties. C.C - Connected Components, Avg. C - Average Clustering

density, connected components, transitivity, and average clustering coefficient, for each data sources and the overall federation. Average clustering coefficient assigns higher scores to low degree nodes, while the transitivity ratio places more weight on the high degree nodes.

Num of nodes	767
Num of edges	8698
Graph density	0.0296
Avg.num neighbors	22.6805
Connected components	67
Transitivity	0.0585
Clustering coefficient	0.0811

Table 1: Graph Metrics

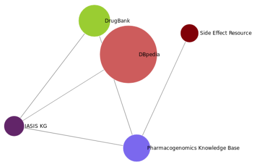


Fig. 3: Source Links

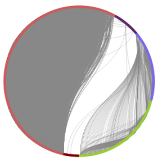


Fig. 4: links

Exploratory Metadata Analysis. In this use case, the attendee will explore the metadata of the federation to understand the characteristics of an RDF-MT, as in Fig. 5, and formulate a federated query, e.g., Fig. 6. After formulating the federated query by exploring RDF-MT properties, the query will be executed by a federated query processing engine integrated in FedSDM and results will be displayed, Fig. 7. Results can be exported as CSV, TSV, Excel, or PDF formats.

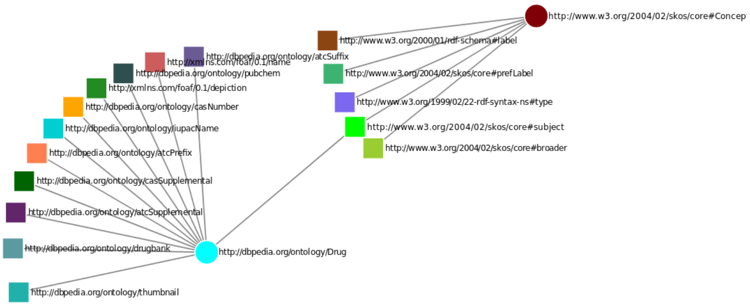
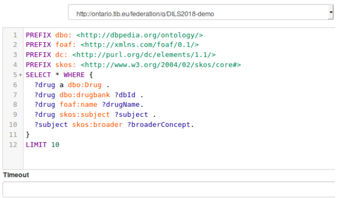
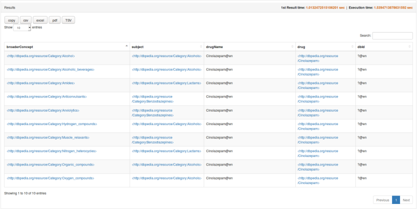

http://www.w3.org/2004/02/skos/core#Concep
http://www.w3.org/2000/01/rdf-schema#label
http://dbpedia.org/ontology/atcSuffix
http://xmlns.com/foaf/0.1/name
http://dbpedia.org/ontology/pubchem
http://www.w3.org/2004/02/skos/core#prefLabel
http://xmlns.com/foaf/0.1/depiction
http://dbpedia.org/ontology/casNumber
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/2004/02/skos/core#subject
http://dbpedia.org/ontology/iupacName
http://www.w3.org/2004/02/skos/core#broader
http://dbpedia.org/ontology/atcPrefix
http://dbpedia.org/ontology/casSupplemental
http://dbpedia.org/ontology/atcSupplemental
http://dbpedia.org/ontology/drugbank
http://dbpedia.org/ontology/Drug
http://dbpedia.org/ontology/thumbnail

Fig. 5: dbo:Drug predicates & links



http://ontario.tib.eu/federation/q/DILS2018-demo

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT * WHERE {
  ?drug a dbo:Drug .
  ?drug dbo:drugbank ?dbId .
  ?drug foaf:name ?drugName.
  ?drug skos:subject ?subject .
  ?subject skos:broader ?broaderConcept.
}
LIMIT 10
```



Results

1st Result time: 1.0132472515106201 sec | Execution time: 1.5394713878631592 sec

broaderConcept	subject	drugName	drug	dbId
<http://dbpedia.org/resource/Category:Alcohol>	<http://dbpedia.org/resource/Category:Alcohols>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en
<http://dbpedia.org/resource/Category:Alcoholic_beverages>	<http://dbpedia.org/resource/Category:Alcohols>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en
<http://dbpedia.org/resource/Category:Amides>	<http://dbpedia.org/resource/Category:Lactams>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en
<http://dbpedia.org/resource/Category:Anticonvulsants>	<http://dbpedia.org/resource/Category:Benzodiazepines>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en
<http://dbpedia.org/resource/Category:Anxiolytics>	<http://dbpedia.org/resource/Category:Benzodiazepines>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en
<http://dbpedia.org/resource/Category:Hydrogen_compounds>	<http://dbpedia.org/resource/Category:Alcohols>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en
<http://dbpedia.org/resource/Category:Muscle_relaxants>	<http://dbpedia.org/resource/Category:Benzodiazepines>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en
<http://dbpedia.org/resource/Category:Nitrogen_heterocycles>	<http://dbpedia.org/resource/Category:Lactams>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en
<http://dbpedia.org/resource/Category:Organic_compounds>	<http://dbpedia.org/resource/Category:Alcohols>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en
<http://dbpedia.org/resource/Category:Oxygen_compounds>	<http://dbpedia.org/resource/Category:Alcohols>	Cinolazepam@en	<http://dbpedia.org/resource/Cinolazepam>	?@en

Showing 1 to 10 of 10 entries

Fig. 6: SPARQL query based on the dbo:Drug metadata

Fig. 7: dbo:Drug query results (tabular)

4 Conclusions and Future Work

We present FedSDM, a semantic data manager for data federation and analysis. FedSDM provides a visual analysis of data sources and a federation by using RDF Molecule Templates. FedSDM provides an exploratory analysis on the metadata of the federation sources. In addition, FedSDM able to generate basic graph properties of RDF-MT graph. For future work, we plan to extend FedSDM to support domain specific visualization of SPARQL query results and analysis of data sources via sampling. Furthermore, FedSDM will be equipped with a component to define privacy and access control rules that must be enforced during federated query processing [4].

Acknowledgements This work has been funded by the EU H2020 RIA under the Marie Skłodowska-Curie grant agreement No. 642795 (WDAqua) and EU H2020 Programme for the projects with GA No. 727658 (IASIS).

References

1. M. Acosta, M. Vidal, T. Lampo, J. Castillo, and E. Ruckhaus. ANAPSID: an adaptive query processing engine for SPARQL endpoints. In *ISWC*, 2011.
2. W. Beek, J. D. Fernández, and R. Verborgh. Lod-a-lot: A single-file enabler for data science. In *Proceedings of the 13th International Conference on Semantic Systems, SEMANTICS 2017, Amsterdam, The Netherlands, September 11-14, 2017*, pages 181–184, 2017.
3. A. Charalambidis, A. Troumpoukis, and S. Konstantopoulos. Semagrow: Optimizing federated sparql queries. In *Proceedings of the 11th International Conference on Semantic Systems*, pages 121–128. ACM, 2015.
4. K. M. Endris, Z. Almhithawi, I. Lytra, M. Vidal, and S. Auer. BOUNCER: privacy-aware query processing over federations of RDF datasets. In *Database and Expert Systems Applications - 29th International Conference, DEXA 2018, Regensburg, Germany, September 3-6, 2018, Proceedings, Part I*, pages 69–84, 2018.
5. K. M. Endris, M. Galkin, I. Lytra, M. N. Mami, M. Vidal, and S. Auer. MULDER: querying the linked data web by bridging RDF molecule templates. In *DEXA*, 2017.
6. I. Fundulaki and S. Auer. Linked Open Data - Introduction to the special theme. *ERCIM News*, 2014(96), 2014.
7. O. Görlitz and S. Staab. SPLENDID: SPARQL endpoint federation exploiting VOID descriptions. In *COLD*, 2011.
8. A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt. FedX: Optimization Techniques for Federated Query Processing on Linked Data. In *ISWC*, 2011.
9. M. Vidal, S. Castillo, M. Acosta, G. Montoya, and G. Palma. On the selection of SPARQL endpoints to efficiently execute federated SPARQL queries. *Trans. Large-Scale Data- and Knowledge-Centered Systems*, 25:109–149, 2016.