SDM - LAB Assignment
Knowledge Graphs

By

Josu Bernal
Stefanos Kypritidis

Teachers:
Oscar Romero Moral
Anna Queralt Calafat

Barcelona, May 2025

B.1 TBOX definition

The graphical representation of the TBOX was created using several tools. Initially, we used Python along with the RDFLib and pydotplus libraries and also matplotlib to generate and visualize the graph. However, due to the high number of nodes and edges, the resulting graphs were quite cluttered and difficult to interpret. To improve readability, we recreated the graph using NetworkX and exported it as a *.graphml* file. This format allowed us to import the graph into Gephi, a visualization tool that offers more flexibility. With Gephi, we were able to customize the layout, reposition nodes and edges, and overall produce a much clearer and more readable visualization.

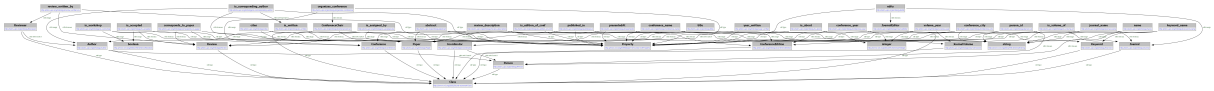

Figure 1: Graphical representation using pydotplus

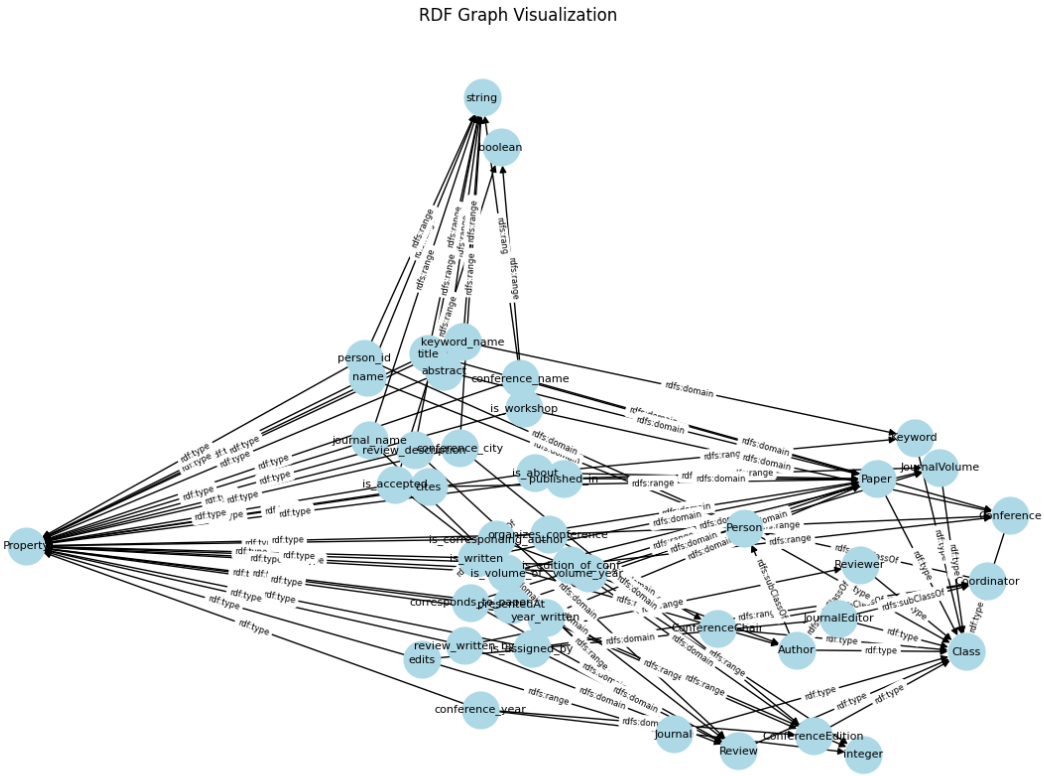

Figure 2: Graphical representation using matplotlib

For best visualization, we recommend opening the file *output_files/gephi_project_final_representation.gephi* inside Gephi and hovering over the nodes to see which nodes are connected and via which edges. Class nodes are with light red and property nodes are with grey. Additionally, the properties that link classes are in dark grey, and the properties that link a class to a literal value are with light grey.

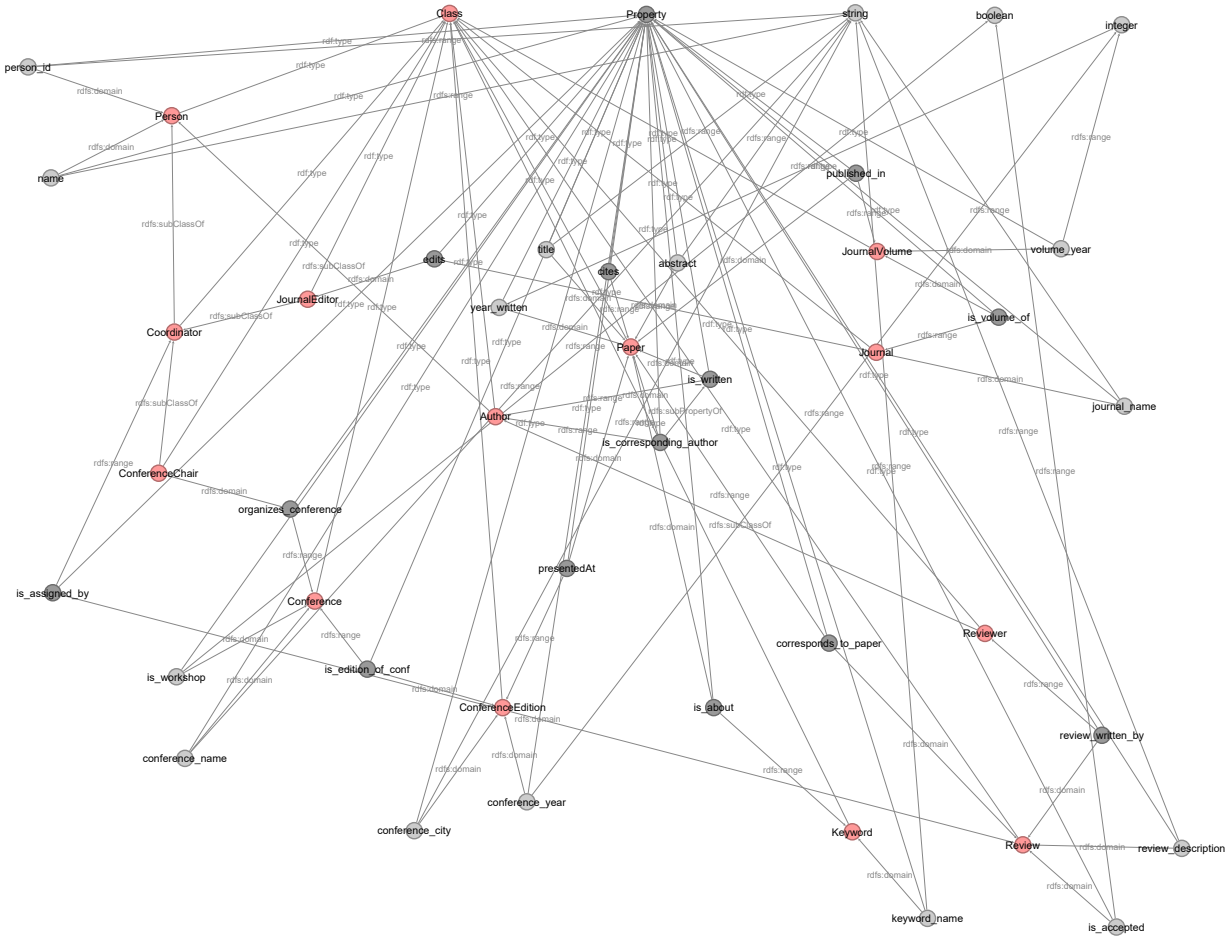

Figure 3: Graphical representation using Gephi

B.2 ABOX definition

B.2.1 ABOX methodology

For the ABox, we utilized the RDFLib library as proposed in the statement. We processed the various CSV files generated from the previous project, ensuring that the appropriate triples were created. While the process is straightforward, it is labor-intensive due to the necessity of employing multiple data processing techniques, such as joining, hashing (for URI generation of instances that did not have an ID beforehand), or casting, to refine the information into the required semantic format.

B.2.3 Inference regime entailment

Inference is the key feature of this project. We utilized GraphDB, as recommended in the project statement; however, we also explored various other methods of inference, including specific Python libraries designed for the task. Ultimately, we chose GraphDB due to its simplicity and speed.

The primary benefit of using inference is that it allows us to avoid writing redundant triples that can be inferred from other triples. Let's review the triples that we managed to avoid writing explicitly.

1. We avoided writing the links *is_written* for the corresponding authors, due to the fact that *is_corresponding_author* is a subproperty of the property *is_written*

2. We skipped all the links *rdf:type* of the ABox since having explicitly defined every domain and range of every property made the inference already possible.

This gave us a total of 381,132 statements from which, 277,811 were explicitly written and 103,321 were inferred, resulting in a 1.37 expansion ratio.

B.2.4 Summary stats

After importing the data into GraphDB, we utilized the SPARQL Query Editor to extract key knowledge graph statistics. Specifically, we calculated the number of classes, the number of properties, the number of instances for each class, and the number of triples for each property. See Appendix for the detailed queries.

The tables below showcase the resulting statistics:

Table 1: Summary Table

Metric	**Total Number**
Classes	13
Properties	27

Table 2: Class Instance Counts

Class	**Instance Count**
sdm:Author	19636
sdm:Person	20443
sdm:Conference	119
sdm:ConferenceChair	119
sdm:Coordinator	807
sdm:ConferenceEdition	398
sdm:Journal	688
sdm:JournalEditor	688
sdm:JournalVolume	2541
sdm:Keyword	225
sdm:Paper	8642
sdm:Review	25926
sdm:Reviewer	14348

Table 3: Triplet Counts per Property

Property	**Triplet Count**
sdm:abstract	8642
sdm:cites	9669
sdm:conference_city	398
sdm:conference_name	119
sdm:conference_year	398
sdm:corresponds_to_paper	25926
sdm:edits	688
sdm:is_about	26131
sdm:is_accepted	25926
sdm:is_assigned_by	25926
sdm:is_corresponding_author	8575
sdm:is_edition_of_conf	398
sdm:is_volume_of	2541
sdm:is_workshop	119
sdm:is_written	33344
sdm:journal_name	688
sdm:keyword_name	225
sdm:name	20443
sdm:organizes_conference	119
sdm:person_id	20443
sdm:presentedAt	814
sdm:published_in	3072
sdm:review_description	25926
sdm:review_written_by	25926
sdm:title	8642
sdm:volume_year	2541
sdm:year_written	8642

Table 4: Infered triples breakdown A-Box

Triple type	**Why?**	**Triplet Count**
rdf:type	One per instance	94,580
sdm:is_written	One per paper	8,642

We can notice that this does not add up. This is because in the T Box, some triplets are being inferred as well; since this is not the focus of the project, we won't delve deeper into those.

B.3 Querying the ontology

The first query finds the average number of citations made to a paper by year and by publication type, and can tell us a lot about how research evolves over time in terms of journals,

workshops, and conferences. It helps us see patterns and compare whether journal articles get more long-term attention than conference or workshop papers. Also, this kind of insight can help researchers choose the best place for them to publish their work.

Average citations made to a paper by year and by publication type

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://sdm_upc.org/ontology/>

SELECT ?typeLabel ?year (COALESCE(AVG(?edgeCount), 0) AS ?citesAvg)
WHERE {
  VALUES ?typeLabel { "Workshop" "Conference" "Journal" }
  VALUES ?year {
    2000 2001 2002 2003 2004 2005 2006 2007 2008
    2009 2010 2011 2012 2013 2014 2015 2016 2017
    2018 2019 2020 2021 2022
  }
    OPTIONAL
  {
    {SELECT ?node ?typeLabel ?year (COUNT(?other) AS ?edgeCount)
    WHERE {
      ?other dbo:cites ?node .
      ?node dbo:presentedAt ?x .
      ?x dbo:conference_year ?year .
      ?x dbo:is_edition_of_conf ?conf .
      ?conf dbo:is_workshop ?type .

      BIND(IF(?type = true, "Workshop", "Conference") AS ?typeLabel)
    }
    GROUP BY ?node ?typeLabel ?year}
 UNION
{SELECT ?node ?typeLabel ?year (COUNT(?other) AS ?edgeCount)
WHERE {
      ?other dbo:cites ?node .
      ?node dbo:published_in ?x .
      ?x dbo:volume_year ?year .
      ?x dbo:is_volume_of ?conf .

  BIND("Journal" AS ?typeLabel)
}
GROUP BY  ?node ?typeLabel ?year}
  }
}
GROUP BY ?typeLabel ?year
ORDER BY ?year
```

An interesting query to run is identifying the most cited paper for each keyword. This can be particularly helpful for researchers looking to find the most influential work in a specific area, for libraries aiming to highlight key papers by topic, and for students exploring new or unfamiliar subjects. One important detail to note is that the query may return multiple rows for a single keyword in cases where there's a tie in citation count.

Most cited paper per keyword

```
PREFIX sdm: <http://sdm_upc.org/ontology/>

# Outer query selects the most cited paper for each keyword
SELECT ?keyword_name ?paper_title ?citationCount
WHERE {
  {
    # Subquery - returns the citation count per pair(paper, keyword)
    SELECT ?keyword ?paper (COUNT(?citingPaper) AS ?citationCount)
    WHERE {
      ?paper sdm:is_about ?keyword .
      ?citingPaper sdm:cites ?paper .
    }
    GROUP BY ?keyword ?paper
  }

  {
    # Subquery - gets max citation count per keyword
    SELECT ?keyword (MAX(?count) AS ?maxCount)
    WHERE {
      SELECT ?keyword ?paper (COUNT(?citingPaper) AS ?count)
      WHERE {
        ?paper sdm:is_about ?keyword .
        ?citingPaper sdm:cites ?paper .
      }
      GROUP BY ?keyword ?paper
    }
    GROUP BY ?keyword
  }


  # Filter to only return the most cited paper per keyword
  FILTER(?citationCount = ?maxCount)
  # Get the corresponding title and keyword name
  ?paper sdm:title ?paper_title .
  ?keyword sdm:keyword_name ?keyword_name .

}
ORDER BY DESC(?citationCount)
```

Lastly, a bonus-query that was noteworthy to include identifies the authors who have collaborated with the largest number of peers. These individuals are likely to possess extensive professional networks, which may facilitate the widespread dissemination of important information. The query retrieves the top 10 most collaborative authors, along with the number of unique collaborators each has worked with.

Top collaborative authors

```
PREFIX sdm: <http://sdm_upc.org/ontology/>

SELECT ?author_name (COUNT(DISTINCT ?coauthor) AS ?collaborators)
WHERE {
   ?paper sdm:is_written ?author.
   ?paper sdm:is_written?coauthor .
   ?author sdm:name ?author_name
   FILTER(?author != ?coauthor)
}
GROUP BY ?author ?author_name
ORDER BY DESC(?collaborators)
LIMIT 10
```

Appendix

SparQL queries for graph stats:

Total number of nodes

```
PREFIX sdm: <http://sdm_upc.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT (COUNT(DISTINCT ?class) AS ?classCount)
WHERE {
  ?class a rdfs:Class .
  FILTER(STRSTARTS(STR(?class), STR(sdm:)))
}
```

Total number of properties

```
PREFIX sdm: <http://sdm_upc.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT (COUNT(DISTINCT ?property) AS ?propertyCount)
WHERE{
    ?property a rdf:Property .
        FILTER(STRSTARTS(STR(?property), STR(sdm:)))
}
ORDER BY ?property
```

Number of instances per class

```
PREFIX sdm: <http://sdm_upc.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?class (COUNT(?instance) AS ?instanceCount)
WHERE {
  ?instance a ?class .
  ?class a rdfs:Class .
  FILTER(STRSTARTS(STR(?class), STR(sdm:)))
}
GROUP BY ?class
```

Number of triplets per property

```
PREFIX sdm: <http://sdm_upc.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?property (COUNT(*) AS ?tripleCount)
WHERE {
  ?s ?property ?o .
  ?property a rdf:Property .
  FILTER(STRSTARTS(STR(?property), STR(sdm:)))
}
GROUP BY ?property
ORDER BY ?property
```