Semantic Data Management

Facultat d'Informatica de Barcelona

LAB Assignment - Knowledge Graphs

May 29, 2024

Authors:
Dilbar Isakova *email*: dilbar.isakova@estudiantat.upc.edu
Narmina Mahmudova *email*: narmina.mahmudova@estudiantat.upc.edu

Professor:
Anna Queralt
Oscar Romero

Contents

1 A. Pre-processing

Following the recommendation to use the same data set from SDM Lab 1, we leveraged the Scopus database for Tsinghua University's computer field spanning from 2018 to 2023 dataset to compare the advantages and disadvantages of knowledge graphs and property graphs. We exported our property graph data into CSV files, including both nodes and relations. To ensure the data's integrity and prepare it for further analysis, we performed some cleaning and pre-precessing. This involved removing unused fields and enriching the dataset with additional relevant fields. Finally, we established connections between the nodes and relations, resulting in refined data files that align with our data model.

After cleaning and preprocessing, we added new fields, appending "ID" to each field name to ensure unique identifiers. For instance, we added 'authorID' to the authors dataset, giving each author a unique identifier. This same process was applied to all other fields, including paper, topic, publication, conference, journal, proceedings, and all remaining fields.

Furthermore, to define the relationships between datasets, we created additional fields. For example, to link Conferences to Proceedings, we created the 'conProIds' column. We applied a similar approach for connecting Journal-Volumes,Conference-Journals.

All data modifications and the correspondiong pre-processing code we documented that can be found here. All the related files and codes of the project is available on this GitHub repository

2 B.1 TBOX Definition

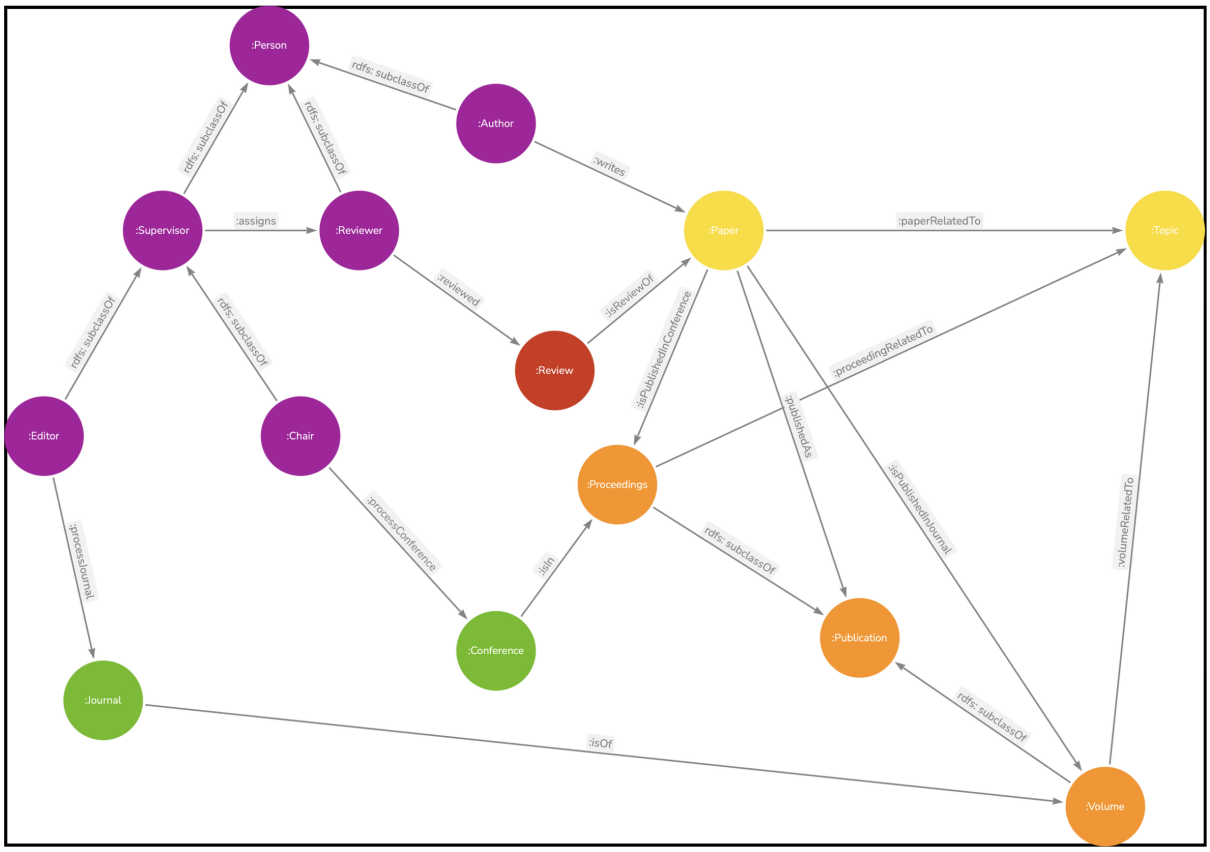

Figure 1: TBOX

To provide a better understanding of our graph structure, you can find a visual representation at this link. Our TBOX was built using Python's RDFS library.

These assumptions help in understanding the structure and logic behind the TBOX modeling in the knowledge graph:

- All `Papers` are considered `Publications`. However, the number of instances for both classes is not the same because the number of `Publication` is smaller, since not all of the papers are accepted. It's important to note that `Papers` that are rejected do not move forward to be included in conference proceedings or journal volumes.
- We created properties like `'paperRelatedTo'`, `'publicationRelatedTo'`, and `'volumeRelatedTo'` to connect conferences, journals, papers, and publications directly to relevant Topics.
- Each edition of a `Conference` or `Journal` where a paper is published is associated with a specific `Topic`. To illustrate these relationships, we link conference proceedings to topics using the `'ProceedingRelatedTo'` connection. Similarly, we connect journal volumes to topics through the `'VolumeRelatedTo'` connection."
- Each conference has its own unique set of proceedings, and similarly, every journal has distinct volumes.
- `Editors` are responsible for managing conferences, and `Chairs` oversee journals. Both roles involve assigning reviewers to papers. To reflect their supervisory duties, both editors and chairs can be modeled as subclasses of `'Supervisor'` which itself is a subclass of `'Person'`. This structure allows supervisors, whether they are editors or chairs, to assign `Reviewers` to papers effectively.
- In RDF schemas, each property must have a unique name. That is why, the relationship between conferences and their proceedings is represented by ':isIn', while the link between journals and their volumes is captured by ':isOf'. This naming convention shows the way how papers from conferences are grouped into proceedings and those from journals are collected into volumes.
- Moreover, all the publications are submitted to only conferences and journals, nowhere other than that.

3 B.2 ABOX Definition

Some assumptions and details related to the ABOX modeling are mentioned below:

- Our ABox creation process involved populating a graph with data from CSV files containing detailed records about authors, papers, conferences, and more. Using RDFLib, this graph is constructed by representing each entity, such as a paper or an author, as a URIRef. Relationships between these entities, like an author writing a paper or a paper being part of a conference proceeding, are depicted as predicates that link these URIRefs. Additionally, literal values including names, titles, and dates are attached to these entities using RDF's literal representation, ensuring that attributes like 'lab:name' or 'lab:title' are accurately reflected in the ABox.
- The class hierarchy, as depicted in the attached visual representation, showcases the structured categorization of our main concepts. The lab:Person class acts as a superclass for entities such as authors, editors, and reviewers, indicating the diverse roles an individual

may assume in the academic domain.

- At the core of our ABox, the lab:Person class ties together various activities within the scholarly domain, demonstrating the multifaceted roles individuals play. For instance, an author might also be a reviewer or a conference chair, reflecting the real-life overlaps in academic responsibilities.

- The classes lab:Paper, lab:Publication, lab:Journal, and lab:Conference are essential to model the publication process. Papers go through reviews (lab:Review) and are associated with specific topics (lab:Topic), which are crucial for categorizing research areas.

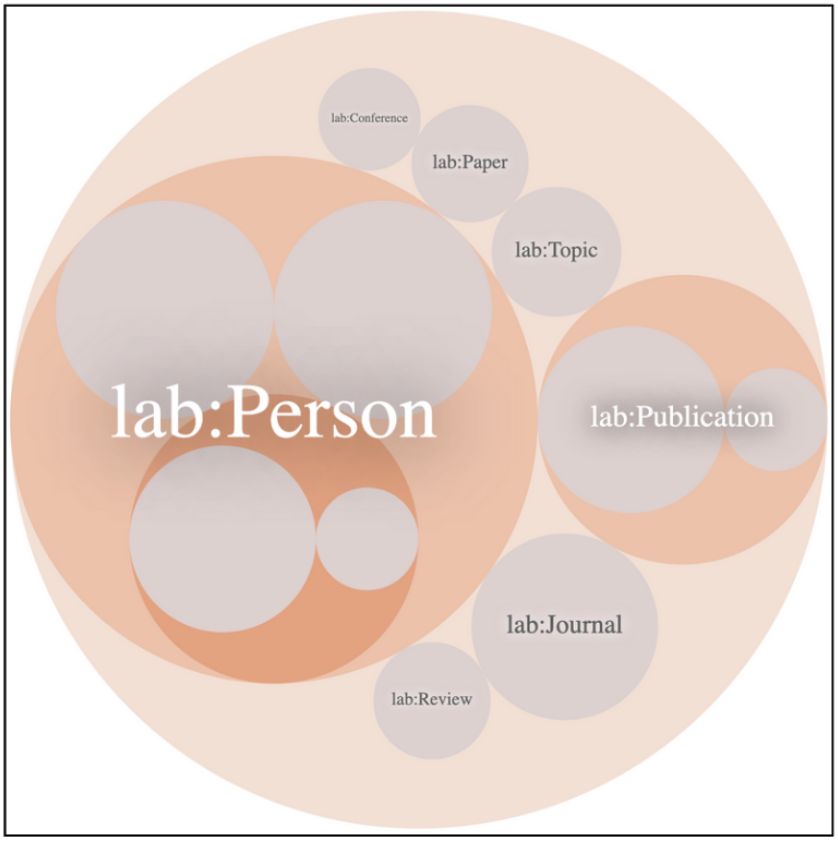

Figure 2: Class hierarchy

4 B.3 Create the final ontology

Triples are added to the graph based on data from CSV files, establishing various relationships:

- Authors are linked to papers they have written, papers to publications and topics, conferences to proceedings, and journals to volumes. Specific links between papers and either conferences or journals, as well as topics to proceedings or volumes, are determined by identifiers in the data. Additionally, chairs are associated with conferences, editors with journals, supervisors with reviewers, and reviewers with reviews, which in turn are linked to papers.

- The TBOX contains terminologies to describe the domain of interest, consisting of classes and properties established initially. The ABOX, on the other hand, contains specific instances and their relationships, loaded with data. Connections in the TBOX involve defining classes (such as LAB.Author and LAB.Paper) and properties (such as LAB.writes

and LAB.publishedAs), while ABOX connections are made by adding individuals and their relationships to the graph from the CSV files, including authors, papers, publications, and conferences.

4.1 Inference regime entailment

- The inference regime employed in our RDF Schema (RDFS) entailment framework uses inference rules to deduce implicit relationships from explicitly stated facts, focusing primarily on class membership represented by rdf:type links. These explicit links are crucial for categorizing entities into specific classes and leveraging the RDFS inference rules to derive further associations.

- To define the ontology and establish connections between the ABOX and TBOX, we employ the same RDFLib tools as used during ABOX creation. Using the `graph.add()` function, we iterate over the datasets, adding instances where each iteration results in the insertion of a triple. `Type` from the RDF namespace indicates that an added instance is a type of a class from namespace `LAB`. For instance, when adding supervisors, the process is as follows:

```
for k in range(len(supervisors['supervisorsIDs'])):
g.add((URIRef(LAB + supervisors['authorID'][k]), RDF.type, LAB.Reviewer))
```

Here, each supervisor is added as an instance of the LAB.Reviewer class, effectively linking the supervisor data from our dataset to the corresponding class in our ontology.

Note that for this laboratory, we have used RDFS-Plus (Optimized) entailment regiment with inference as on

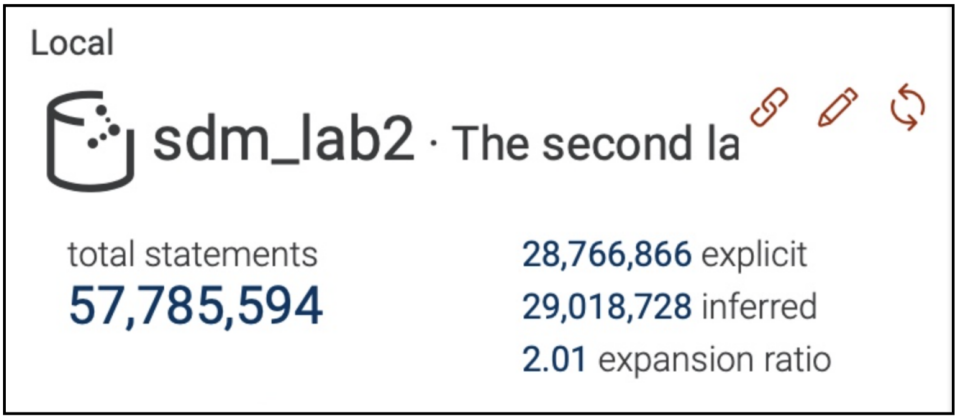

Figure 3: Class hierarchy

4.2 Summary table of instances

Here can be found the queries used to obtain statistics below.

The below table shows the statistics of the graph obtained after linking TBOX and ABOX:

Description	Value
Number of Instances per Class	12
Triples for Each Property	12
Total Number of Triples	1

Table 1: General Information

Main Class	Number of Instances
Author	105,871
Paper	500
Publication	20,729
Topic	487
Proceedings	233
Volume	20,111

Table 2: Instances for Each Main Class

Main Property	Number of Triples
writes	105,872
publishedAs	501
paperRelatedTo	501
isIn	230
isOf	20,001

Table 3: Number of Triples Using Main Properties

5 B.4 Querying the ontology

Query 1: Find all Authors.

```
PREFIX lab: <http://SDM_LAB2.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?author ?name
WHERE {
    ?author rdf:type lab:Author .
    ?author lab:Name ?name .
}
```

As outlined during the creation of the TBox and ABox, the "Author" rdfs:Class makes identifying all authors straightforward; simply search for all data instances classified under the rdf:type "Author." In this specific query, we also retrieve the authorName. The query and its results.

Query 2: Find all properties whose domain is Author.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX lab: <http://SDM_LAB2.org/>

SELECT ?property
WHERE
{
    ?property rdfs:domain lab:Author .
}
```

As specified during the TBox creation, the "Author" rdfs:Class encompasses properties such as "authorName" and "writes." This query targets these properties since "Author" is designated as the domain. The query and its results.

Query 3: Find all properties whose domain is either Conference or Journal.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX lab: <http://SDM_LAB2.org/>

SELECT ?property
WHERE {
    {
        ?property rdfs:domain lab:Conference.
    }
    UNION
    {
        ?property rdfs:domain lab:Journal.
    }
}
```

As similarly to the Query 2, the Conference and Journal classes do not possess any properties where they act as a superclass in the domain. Therefore, only the direct properties' domains are identified. The query and its results.

Query 4: Find all the papers written by a given author that where published in database conferences.

```
PREFIX lab: <http://SDM_LAB2.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?paper ?title
WHERE {
    ?author lab:Name "Chen Y." .
    ?author lab:writes ?paper .
    ?paper lab:title ?title .
    ?paper lab:isPublishedInConference ?proceedings .
    ?proceedings lab:proceedingRelatedTo ?topic .
    ?topic lab:topicName "Chlorine Compounds" .
}
```

In this query, we select "Chen Y." as the author in focus. Initially, we identify the papers authored by Chen Y., and then we narrow the search to those papers accepted at conference proceedings connected to the topic of "Chlorine compounds." The assumption for this query was to find papers presented at a database conference. However, since our dataset does not include the topic of Database, we substituted it with "Chlorine compounds" instead. The query and its results.

Query 5: Retrieve the average length of papers published in each journal.

```
PREFIX lab: <http://SDM_LAB2.org/>
SELECT ?journal (AVG(STRLEN(?abstract)) AS ?avgAbstractLength)
WHERE {
  ?paper a lab:Paper ;
         lab:publishedAs ?publication ;
         lab:isPublishedInJournal ?journal .
  OPTIONAL { ?paper lab:abstract ?abstract }
}
GROUP BY ?journal
ORDER BY DESC(?avgAbstractLength)
```

This query calculates the average length of paper abstracts published in each journal. It selects journals and computes the average length of the 'abstract' field for papers that are categorized as belonging to these journals. If a paper does not have an abstract, it is still included in the average calculation but treated as having a length of zero. The results are grouped by journal and ordered in descending order of average abstract length, allowing us to identify which journals feature longer abstracts on average. The query and its results.

Query 6: List the conferences with the highest number of papers published.

```
PREFIX lab: <http://SDM_LAB2.org/>
SELECT ?conference (COUNT(?paper) AS ?numPapers)
WHERE {
  ?paper a lab:Paper ;
         lab:isPublishedInConference ?conference .
}
GROUP BY ?conference
ORDER BY DESC(?numPapers)
LIMIT 10
```

This query retrieves the top ten conferences based on the number of papers published at each event. It counts the total papers tagged as being presented at each conference and groups the results by conference. The outcome is ordered in descending order to display the conferences with the highest paper counts first, providing insight into which conferences are the most prolific in terms of paper publications. The query and its results.