Knowledge Graph Lab - SDM3

Spring Semester, 2021

Mohana Fathollahi, Hattem Dayekh

May 26, 2021

1 B Ontology creation

1.1 B.1 TBOX definition

In the figure 1, the main structure of TBOX has been provided. In this structure, all relation for conference papers and journal papers that are accepted or rejected have been considered. To make it easier to follow structure of graph, we mentioned basic structures in below:

- Paper considered as a class and journal paper and conference paper considered as subclass of it.
- Journal paper considered as a class and has journal publication, full paper, short paper and demo paper as sub classes. Journal publications are articles that accepted and will be published in journal.
- Conference paper has five sub classes, conference publication, short paper, demo and poster paper. Like journal publication, conference publication are those paper that accepted and will be published in conference.
- When a paper submit to journal or conference, editor or chair respectively should assign reviewer to this paper. Therefore, we can say that reviewer is not only related to journal paper or editor, but also it related to pair of them; $< JournalPaper, Editor >$. Here we do not have binary relation and we should use Reification concept to solve problem of n-binary relations. As a result, we created blank node with name of eventAssign node for each article.
- Additionally, each reviewer should reject or accept a paper and give his or her opinion about paper. opinion of a reviewer is a concept that relate to both of reviewer and article, like previous situation we should create blank node to connect these three concepts.Therefore, eventreviwe has been created.

The language that has been considered to build our knowledge graph is RDFS, even though OWL is richer language with respect to using cardinality constrains. For example in this project we should consider that every paper can be reviewed by at least two reviewers. We could not implement this constraint in RDFS, while we could do that in OWL. However, the inference in OWL is more difficult and expensive than RDFS.

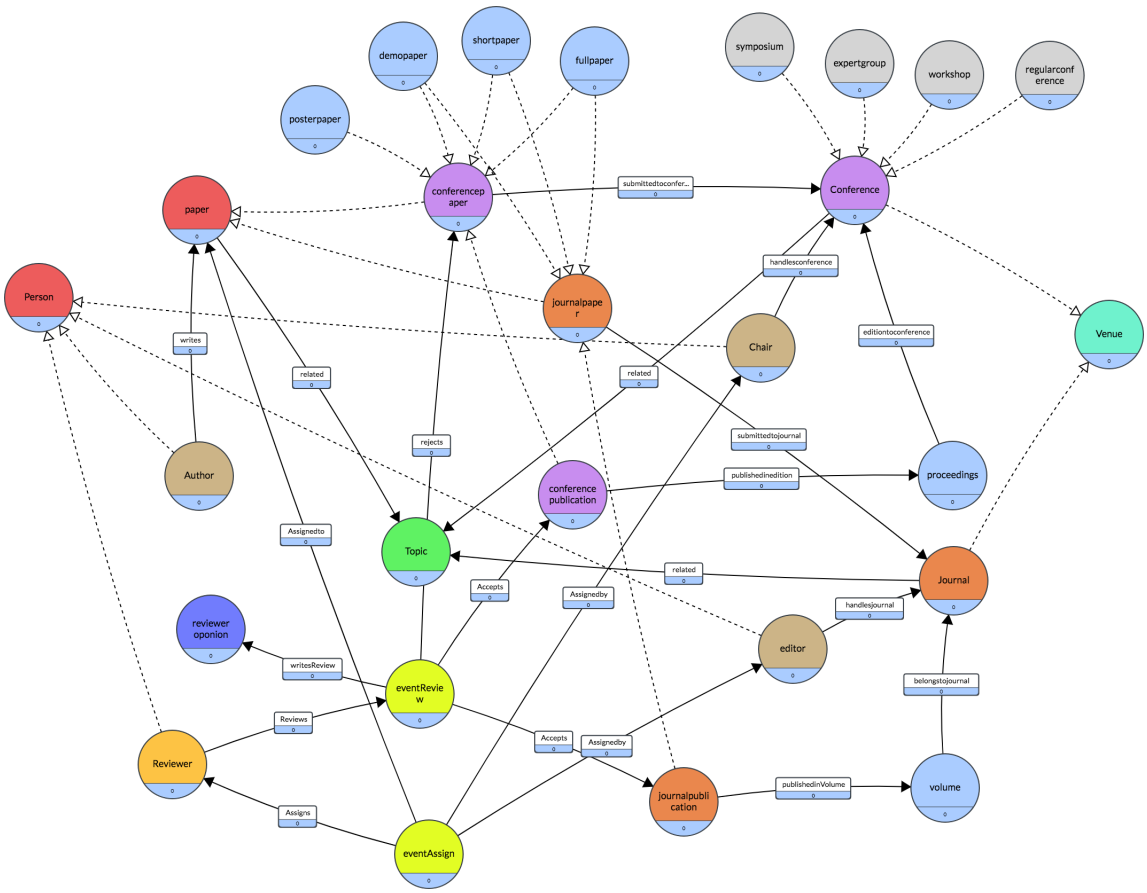


Figure 1: Graph structure

To generate the TBOX, execute the code in the Java project. The execution of the script *main.java* will generate a TBOX Turtle file in the folder ontologies, and the output file is placed in the folder, *TBOX.ttl*.

2 B.2 ABOX definition

For the ABOX definition, we have reused the data from the lab 1.
We did some preprocessing on data to make it ready for converting to rdf and link them to TBOX. This preprossesing has been done in python with name of $Group-B2-DayekhFathollahi.py$. After running[1] this scrip we could get information of conference and journals in 4 different csvs that provided in below:

- Conference publications accepted
- Journal publications accepted
- Conferences rejected
- Journals rejected

The next step will be converting these csvs to rdf files. We used Open Refine to create these files. The easier way to build connections in Open Refine is importing vocabulary terms from a file. We used *TBOX.ttl* that we created in the part B.1 and imported in to Open Refine. Therefore, we get vocabulary terms that we want.

[1]Because some part of code randomly assigned features to variables you may get different results, after running the script.

After constructing those synthetic rdf files, using Apache Jena, we loaded the ABOX in the same Java script (main.java) that generates the TBOX since the properties and classes were already defined in the script.
We had a limitation on creating all triples in Open Refine. For example, it does not have rdf: property that is used to state that the values of a property are instances of one or more classes. To be more clear about cases like this in our graph, we should mention that; conference paper is an instance of class of conference paper and it can be instance of fullpaper or short paper class too. Therefore we should link conference paper to two classes. It happens for journal articles and different kind of conferences like; workshops, symposiums and so on.
The best solution that we could find is randomly assign conference papers, journal papers and conferences to each of classes that have been defined in *main.java*.

3 B.3 Create the final ontology

3.1 Inference regime entailment

The regime entailment chosen in GraphDB is RDFS Plus. With Consideration of inference we could save to explicitly generate the rdf:type that previously defined in the TBOX. For example when a full paper of conference has property rdf:type with conference paper and in TBOX, conference paper is sub class of paper we can save rdf:type property between full conference paper and paper.

3.2 Summary of statistic

Therfore, after running[2] main file in java, we get $ABOX\_TBOX\_RDFS\_2.ttl$ file that has all information that we want and in this step, we will import it into GraphDB. The final result provided in image 3.2 and we can see classes in this picture. In the following statistical detail of most classes have been provided.

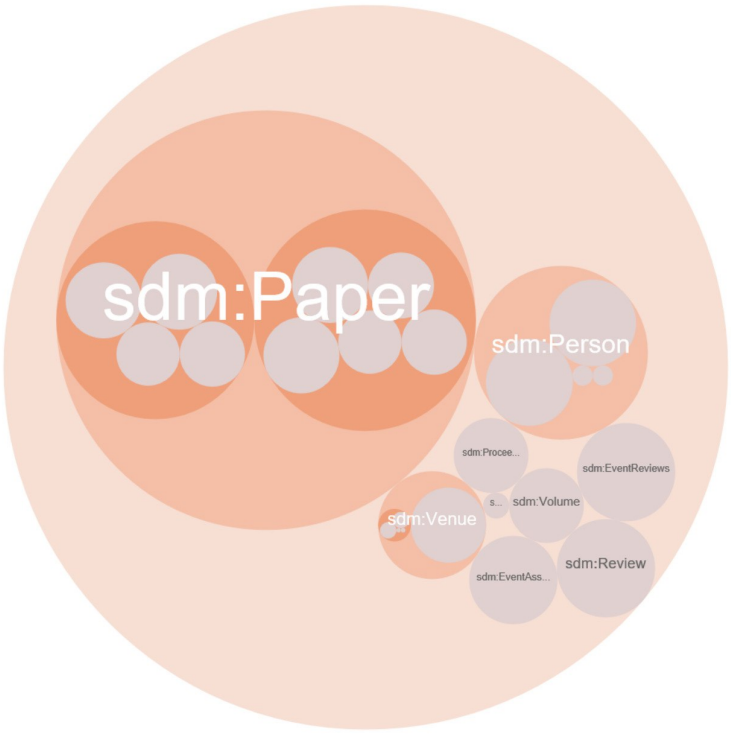


Figure 2: Output of GraphDB

[2]Because last part of code randomly assigned features to variables you may get different results, after running the script.

- Paper considered as a class that has 477 members and 2 sub classes based on table below:

Conference Paper	Journal Paper
239	238

Table 1: Members of Classes of Conference and Journal Papers

- For conference Papers, we used queries in GraphDB to extract how many papers that published belong to each subclass of conference papers. In the table 2 detail of each class for rejected and accepted conference papers have been provided.

	Full paper	Short Paper	Demo Paper	Poster Paper	Total
Published (Accepted)	41	42	37	80	200
Submitted (Rejected)	3	4	13	19	39
Total	44	46	50	99	

Table 2: Conference Papers

- Like what we did for conference papers we did for journal papers and the result of this part provided in table 4

	Full paper	Short Paper	Demo Paper	Total
Published (Accepted)	34	40	126	200
Submitted (Rejected)	4	8	26	38
Total	38	48	152	

Table 3: Journal Papers

- Person as an another class has 496 members with 4 sub classes:

Subclass	Member Count
Reviewer	428
Author	435
Chair	3
Editor	3

Table 4: Sub classes of Pesron

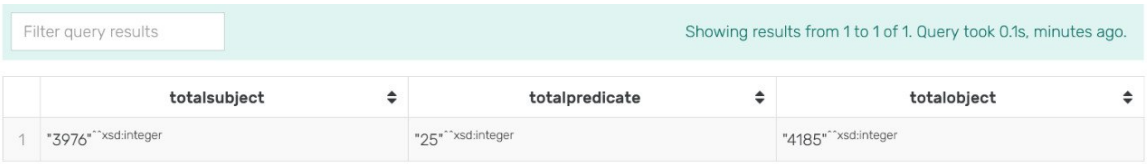
- For other classes number of instances provided in table below:

Class	Count
Review	954
Venue	376
Topic	5

Table 5: Other classes in graph

- Based on query in below, we can get total number of properties, object and subjects;

```
SELECT (COUNT(DISTINCT ?subject) AS ?totalsubject) (COUNT(DISTINCT ?predicate) AS
    ?totalpredicate) (COUNT(DISTINCT ?object) AS ?totalobject)
WHERE
  { ?subject ?predicate ?object}
```



4 B.4 Querying the ontology

4.1 1. Find all Authors.

```
PREFIX sdm: <http://www.sdm.com/>
PREFIX rdfs: <http://www/w3/org/2000/01/rdf-schema#>
select distinct ?Authors
where {
 ?Authors a sdm:Author
}
```

4.2 2. Find all properties whose domain is Author.

The result of this query is sdm:Writes.

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

select distinct ?property
where {
    ?property rdfs:domain sdm:Author
}
```

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

This Query did not return any property, while we know that conference and journal are subclass of venue and venue is domain of *relatedto* property and has a range equal to topic. We can conclude that inference is done through the classes, not the properties.

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

select distinct ?property
where {
    {?property rdfs:domain sdm:Journal}
    union
    {?property rdfs:domain sdm:Conference}
}
```

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

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX sdm: <http://www.sdm.com/>
select ?Author ?Paper ?topic ?ConferenceX
where {
    ?AuthorX sdm:Writes ?PaperX.
    ?AuthorX rdfs:label ?Author.
    ?PaperX rdfs:label ?Paper.
    ?PaperX sdm:PublishedInEdition ?ProceedingX.
    ?ProceedingX sdm:EditionOfConference ?ConferenceX.
    ?ProceedingX a sdm:Proceedings.
 ?ConferenceX sdm:RelatedTo ?topicX.
    ?topicX rdfs:label ?topic.
    filter(?topic = "databases")
    filter(?Author = "Liusheng Huang")
}
```

Filter query results

Showing results from 1 to 1 of 1. Query took 0.1s, today at 05:17.

	Author	Paper	topic	ConferenceX
1	"Liusheng Huang"	"Cryptanalysis of a certificateless signature scheme without pairings."	"databases"	http://www.SDM.com/%20Gartner%20Data%20&%20Analyt