Lab assignment 3: knowledge graphs

Ricardo Rojas Ruíz
Gabriela Martínez

Date: May 8th, 2019

A. Exploring Dbpedia

1. Find the class representing an Actor in the dataset (using filters).

```
SELECT ?class WHERE {
    ?class rdf:type owl:Class FILTER regex(?class, "/Actor$")
}
```

2. Find the super class for the class Actor.

```
SELECT ?superclass WHERE {
    <http://dbpedia.org/ontology/Actor> rdfs:subClassOf ?superclass
}
```

3. Find all the actors in the dataset.

```
SELECT ?actors WHERE {
    ?actors rdf:type <http://dbpedia.org/ontology/Actor>
}
```

4. Get different classes that are defined as range of the properties that have the class Actor defined as their domain.

```
SELECT DISTINCT ?classes WHERE {
    ?property rdfs:domain <http://dbpedia.org/ontology/Actor>.
    ?property rdfs:range ?classes
}
```

5. Find the super property of the goldenRaspberryAward property.

```
PREFIX : <http://dbpedia.org/ontology/>
SELECT ?superProperty WHERE {
   ?property rdf:type rdf:Property.
   ?property rdfs:subPropertyOf ?superProperty
   FILTER (?property=:goldenRaspberryAward)
}
```

6. Return all the properties that have the class Actor as either their range or domain.

```
SELECT ?property WHERE {
?class rdf:type owl:Class FILTER regex(?class, "/Actor$")
 {
   ?property rdfs:domain ?class
 }
  UNION
 {
   ?property rdfs:range ?class
 }
}
```

7. Return all persons that are not actors.

```
SELECT ?notActors WHERE {
   ?notActors rdf:type <http://dbpedia.org/ontology/Person>.
   MINUS {?notActors rdf:type <http://dbpedia.org/ontology/Actor>}
}
```

B. Analytical queries on top of QBAirbase

For the following tasks, we considered the examples and documentation provided at http://qweb.cs.aau.dk/qboairbase/queries/q1.txt, http://qweb.cs.aau.dk/qboairbase/

1. List the country, station type, latitude, and longitude details of each station. *Note*: *Limit the query to 25 results, and extract only the string values of the required object and not the whole IRIs.*

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT
(str(?countryName) as ?Country), (str(?stationType) as ?stationType), ?longitude, ?latitude WHERE {
?obs schema:station ?station.
?station property:type ?stationType.
?station schema:inCountry ?country.
?country property:country ?countryName.
?station property:latitudeDegree ?latitude.
?station property:longitudeDegree ?longitude.
}
LIMIT 25
```

2. List the 10 highest averages of C6H6 emission and the country and the year on which they were recorded. *Note: A sensor has a property (defined through the prex: <http://qweb.cs.aau.dk/airbase/property/>) stastisticShortName, and it can be Mean, Max, etc.*

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT
str(?countryName) as ?Country, ?yearNum, (avg(?C6H6) as ?avgC6H6) WHERE {
?obs schema:station ?station.
?station schema:inCountry ?country.
?country property:country ?countryName.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?obs schema:C6H6 ?C6H6.
?obs schema:sensor ?sensor .
?sensor property:statisticShortName "Mean"^^xsd:string.
}
ORDER BY DESC (?avgC6H6)
LIMIT 10
```

3. For each city and property type, give the yearly average emission for NO2, SO2, PB, and PM10.

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT * {

{
SELECT DISTINCT
str(?cityName) as ?City, str(?stationType) as ?stationType, ?yearNum as ?Year, ?metric, (avg(?var) as ?avgMetric)
WHERE {
?obs schema:station ?station. BIND ("NO2" as ?metric).
?station schema:inCity ?city.
?city property:city ?cityName.
?station property:type ?stationType.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?obs schema:NO2 ?var.
?obs schema:sensor ?sensor.
?sensor property:statisticShortName "Mean"^^xsd:string.
}
ORDER BY ?City, ?stationType, ?Year
}
UNION
{
SELECT DISTINCT
str(?cityName) as ?City, str(?stationType) as ?stationType, ?yearNum as ?Year, ?metric, (avg(?var) as ?avgMetric)
WHERE {
?obs schema:station ?station. BIND ("SO2" as ?metric).
```

```
?station schema:inCity ?city.
?city property:city ?cityName.
?station property:type ?stationType.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?obs schema:SO2 ?var.
?obs schema:sensor ?sensor.
?sensor property:statisticShortName "Mean"^^xsd:string.
}
ORDER BY ?City, ?stationType, ?Year
}
UNION
{
SELECT DISTINCT
str(?cityName) as ?City, str(?stationType) as ?stationType, ?yearNum as ?Year, ?metric, (avg(?var) as ?avgMetric)
WHERE {
?obs schema:station ?station. BIND ("PB" as ?metric).
?station schema:inCity ?city.
?city property:city ?cityName.
?station property:type ?stationType.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?obs schema:PB ?var.
?obs schema:sensor ?sensor.
?sensor property:statisticShortName "Mean"^^xsd:string.
}
ORDER BY ?City, ?stationType, ?Year
}
UNION
{
SELECT DISTINCT
str(?cityName) as ?City, str(?stationType) as ?stationType, ?yearNum as ?Year, ?metric, (avg(?var) as ?avgMetric)
WHERE {
?obs schema:station ?station. BIND ("PM10" as ?metric).
?station schema:inCity ?city.
?city property:city ?cityName.
?station property:type ?stationType.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?obs schema:PM10 ?var.
?obs schema:sensor ?sensor.
?sensor property:statisticShortName "Mean"^^xsd:string.
}
ORDER BY ?City, ?stationType, ?Year
}
}
```

4. Define 3 additional SPARQL queries (and their corresponding interpretation) that you think could be interesting for the domain of analyzing air quality/pollution.

- Get the maximum value of PM2.5 AND PM10 registered for each country, city and year.

Specifically, *"Particulate Matter, also known as soot or PM, consists of microscopically small solid particles or liquid droplets that can either be emitted directly into the air, or formed from secondary reactions involving gaseous pollutants that combine in the atmosphere."* [1]

PM is normally measured in terms of two ranges, according to the size of the particulates being present in the air: PM10 and PM2.5. Measuring both of them is extremely important for monitoring air quality, even though, efforts should be focused on monitoring and reducing PM2.5, as this material, also called "fine particulates", is a serious concern for human health because they can stay within human lungs for days or even weeks, while PM10 is removed from the body within a couple of hours. Therefore, one could be interested in getting to know what has been the maximum concentration of PM in every city per year, so actions can be more focused on the places where higher particulates are present and follow a growing trend over time.

In particular, in India, it has been found that *"it is clear that the pollutants that are most responsible for the poor air quality index in India right now are PM2.5 and PM10 with all other pollutants within good or satisfactory ranges"* [2].

The following table shows the impact of PM particulates on human health per range of measurement:

AQI Category	PM2.5 (ug/m3)	PM10 (ug/m3)	Health Impact
Good (0-50)	0-30	0-50	Minimal
Satisfactory (51-100)	31-60	51-100	Minor Breathing discomfort to sensitive people.
Moderately polluted (101-200)	61-90	101-250	Breathing discomfort to asthma patents, elderly and children.
Poor (201-300)	91-120	251-350	Breathing discomfort to all
Very poor (301-400)	121-250	351-430	Respiratory illness on prolonged exposure.
Severe (401-500)	250+	430+	Health impact even on light physical work. Serious impact on people with heart/lung disease.

Figure 1. Categories of particulate matter according to their concentration. Source: taken from [2].

The query proposed is as follows:

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT * {
{
SELECT DISTINCT
str(?countryName) as ?Country, str(?cityName) as ?City, ?yearNum as ?Year, ?metric, (max(?var) as ?MaxValue)  WHERE
{
?obs schema:station ?station. BIND ("PM2.5" as ?metric).
?station schema:inCountry ?country.
?country property:country ?countryName.
?station schema:inCity ?city.
?city property:city ?cityName.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?obs schema:PM2.5 ?var.
?obs schema:sensor ?sensor.
?sensor property:statisticShortName "Max"^^xsd:string.
}
ORDER BY ?Country, ?City, ?Year, ?MaxValue
}
UNION
{
SELECT DISTINCT
str(?countryName) as ?Country, str(?cityName) as ?City, ?yearNum as ?Year, ?metric, (max(?var) as ?MaxValue)  WHERE
{
?obs schema:station ?station. BIND ("PM10" as ?metric).
?station schema:inCountry ?country.
?country property:country ?countryName.
?station schema:inCity ?city.
?city property:city ?cityName.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?obs schema:PM10 ?var.
?obs schema:sensor ?sensor.
?sensor property:statisticShortName "Max"^^xsd:string.
}
ORDER BY ?Country, ?City, ?Year, ?MaxValue
}
}
```

From the query execution we extracted the results for the city of Barcelona, which are shown as follows:

Country	*City*	*Year*	*Metric*	*MaxValue*
Spain	BARCELONA	2006	PM2.5	104
Spain	BARCELONA	2008	PM2.5	55.6
Spain	BARCELONA	2009	PM2.5	68.2
Spain	BARCELONA	2010	PM2.5	55.2
Spain	BARCELONA	2011	PM2.5	68.7
Spain	BARCELONA	2012	PM2.5	59.1

Table 1. PM2.5 concentration for Barcelona over the years.

Country	*City*	*Year*	*Metric*	*MaxValue*
Spain	BARCELONA	2001	PM10	1608
Spain	BARCELONA	2002	PM10	194
Spain	BARCELONA	2003	PM10	114
Spain	BARCELONA	2004	PM10	224
Spain	BARCELONA	2005	PM10	105
Spain	BARCELONA	2006	PM10	178
Spain	BARCELONA	2007	PM10	220
Spain	BARCELONA	2008	PM10	149
Spain	BARCELONA	2009	PM10	151.9
Spain	BARCELONA	2010	PM10	74.4
Spain	BARCELONA	2011	PM10	121.9
Spain	BARCELONA	2012	PM10	89.7

Table 2. PM10 concentration for Barcelona over the years.

From the previous tables, it can be seen that in general Barcelona hosts a moderately polluted air, which only takes breathing discomfort for asthma patients, elderly and children, with a few exceptions in 2006, when the PM2.5 reached a maximum poor concentration of 104, and in 2001 when the PM10 particulates reached an extremely severe maximum of 1608. Take into consideration that this last figure could not be necessarily true, as it seems to be an outlier when compared with the rest of the numbers and further exploration on this metric is required.

- Average PM2.5 and PM10 per city and its corresponding country.

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT * {
{
SELECT DISTINCT
str(?countryName) as ?Country, str(?cityName) as ?City, ?metric, (avg(?var) as ?AvgValue)  WHERE {
?obs schema:station ?station. BIND ("PM2.5" as ?metric).
?station schema:inCountry ?country.
?country property:country ?countryName.
?station schema:inCity ?city.
?city property:city ?cityName.
?obs schema:PM2.5 ?var.
?obs schema:sensor ?sensor.
?sensor property:statisticShortName "Mean"^^xsd:string.
}
ORDER BY ?Country, ?City, ?AvgValue
}
UNION
{
SELECT DISTINCT
str(?countryName) as ?Country, str(?cityName) as ?City, ?metric, (avg(?var) as ?AvgValue)  WHERE {
?obs schema:station ?station. BIND ("PM10" as ?metric).
?station schema:inCountry ?country.
?country property:country ?countryName.
?station schema:inCity ?city.
?city property:city ?cityName.
?obs schema:PM10 ?var.
?obs schema:sensor ?sensor.
?sensor property:statisticShortName "Mean"^^xsd:string.
}
ORDER BY ?Country, ?City, ?AvgValue
}
}
```

Country	*City*	*Metric*	*AvgValue*
Spain	MADRID	PM2.5	14.823918367346939
Spain	MADRID	PM10	32.032523668639053
United Kingdom	LONDON	PM2.5	17.059273333333333
United Kingdom	LONDON	PM10	28.445099514563107
Belgium	BRUSSELS	PM2.5	20.096732394366197
Belgium	BRUSSELS	PM10	32.450827272727273
Netherlands	AMSTERDAM	PM2.5	16.512928571428571
Netherlands	AMSTERDAM	PM10	29.189045112781955
Germany	BERLIN	PM2.5	19.950125
Germany	BERLIN	PM10	28.702853333333333
Bosnia and Herzegovina	TUZLA	PM2.5	64.019888888888889
Turkey	BATMAN	PM10	119.2627

Table 3. Average PM2.5 and PM10 metrics for some cities.

From Table 3, it can be observed that in general, most of the biggest capitals in Europa have average levels of PM2.5 and PM10 within the best classification possible, even though some of them have had years and months out of this scope. Note that still, there are some cities such as Tuzla and Batman that are moderately polluted according to one of the two metrics, as not both PM2.5 and PM10 are always available for every city.

- Maximum levels of O3 and CO per year

Similar to the previous queries, the next one will retrieve the maximum levels of two other important pollutants: ozone (O3) and carbon monoxide (CO) per year, considering all the territories,

```
PREFIX schema: <http://qweb.cs.aau.dk/airbase/schema/>
PREFIX property: <http://qweb.cs.aau.dk/airbase/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT * {
{
SELECT DISTINCT
?yearNum as ?Year ?metric (max(?o3) as ?MaxMetric) WHERE {
?station property:type ?stationType. BIND ("O3" as ?metric).
?station property:ozoneClassification ?class.
?obs schema:station ?station.
?obs schema:O3 ?o3.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?obs schema:sensor ?sensor .
?sensor property:statisticShortName "Max"^^xsd:string.
} GROUP BY ?yearNum ?metric
ORDER BY ?yearNum
}
UNION
{
SELECT DISTINCT
?yearNum as ?Year ?metric (max(?co) as ?MaxMetric) WHERE {
?station property:type ?stationType. BIND ("CO" as ?metric).
?station property:ozoneClassification ?class.
?obs schema:station ?station.
?obs schema:CO ?co.
?obs schema:year ?year.
?year property:yearNum ?yearNum.
?obs schema:sensor ?sensor .
?sensor property:statisticShortName "Max"^^xsd:string.
} GROUP BY ?yearNum ?metric
ORDER BY ?yearNum
}
}
```

A preview of the results is shown in the following table:

Year	*Metric*	*MaxValue*
2009	O3	377.72
2010	O3	332.29
2011	O3	512.32
2009	CO	25.3
2010	CO	30.6
2011	CO	17.5

Table 4. Max O3 and CO metrics per year for the European territory.

It is important to highlight that carbon monoxide only represents a serious risk for human health only under permanent exposure in the long-term when its levels are around 50 PPM [3]. As seen in the previous table, the European territories are not experiencing a serious issue in this regard because even these maximum values are not considered hazardous.

On the other hand, note that some of the maximum levels of ozone in the last years have been around 300 and even more. I this case, those levels do represent a serious risk for human health because and outdoor activities should be avoided when those peaks are reached [4].

C. Ontology creation

C.1. TBOX definition

For the TBOX definition, we used Apache Jena and the full code can be found in the following repository: https://github.com/mgmartinezl/SDM_Lab3/blob/master/Main.java

The classes created were:

- Person: imported from *http://dbpedia.org/ontology/Person*
- Author: imported from *http://dbpedia.org/ontology/Writer*
- Paper: imported from *http://dbpedia.org/ontology/Article*
    - Short paper
    - Full paper
    - Demo paper
    - Survey paper
- Journal: imported from *http://dbpedia.org/ontology/AcademicJournal*
    - Open access journal
- Conference: imported from *http://dbpedia.org/ontology/AcademicConference*
    - Database conference
- Organization: imported from *http://dbpedia.org/ontology/Organisation*
- Reviewer

- Review

Moreover, we also defined some data properties for some classes, such as:

- *Paper*
    - hasKeywords
    - hasTitle
    - hasAbstract: imported from *http://dbpedia.org/ontology/abstract*
- *Conference*
    - hasMonth
    - hasYear
    - hasCity: imported from *http://dbpedia.org/ontology/cityLink*
    - hasEdition
    - hasName
- *Journal*
    - hasVolume
    - hasYear
    - hasName
- *Organization*
    - hasName
- *Author*
    - hasBirthName: imported from *http://dbpedia.org/ontology/birthName*
    - hasGender: imported from *http://dbpedia.org/ontology/sex*
- *Review*
    - hasDecision
    - hasScore

Also, some object properties were set in order to establish relationships between classes:

- *isInOrganization*
    - Domain: person
    - Range: organization
- *isCitedBy*
    - Domain: paper
    - Range: paper
- *coAuthors*
    - Domain: paper
    - Range: author
- *correspondingAuthor* (as only one author can be the principal one for each paper)
    - Domain: paper
    - Range: author
- *hasPapers*
    - Domain: conference and journal
    - Range: paper
- *hasReviewer*
    - Domain: review
    - Range: reviewer

- *ofPaper*
    - Domain: review
    - Range: paper

Finally, some restrictions were done on the ontology, to specify that:

- If a review exists, it must have a reviewer and paper associated.
- There exists a cardinality one to one restriction between every paper and a principal or corresponding author.
- Corresponding authors are disjoint with the co-authors of a paper.

Note, however, that this graph may contain other constraints or restrictions not covered for this exercise.

From Apache Jena, the ontology created in OWL can be previewed here:
https://github.com/mgmartinezl/SDM_Lab3/blob/master/myOntology.owl.
Also, a graphical representation of the ontology was made in Protègè, and is as follows:

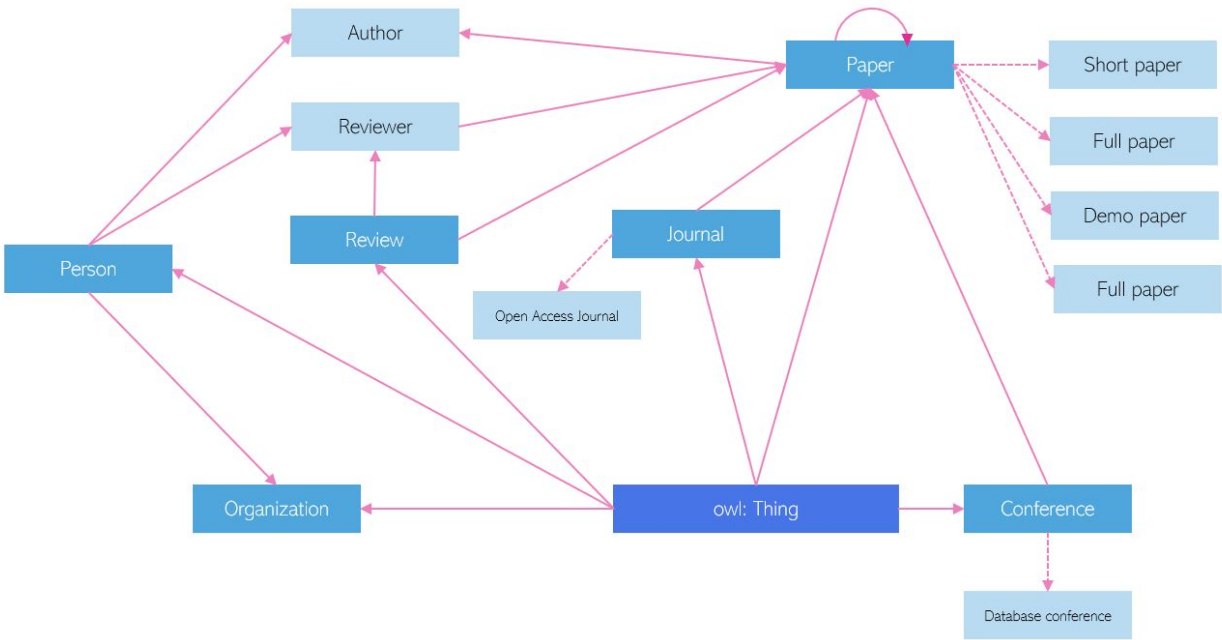

Figure 2. Visual representation of the ontology created for the DBLP database.

C.2. ABOX definition

For the ABOX definition, we also used Apache Jena. Through this API, we read the *txt* files from the previous assignment and converted them into statements of triplets that contain subjects, predicates, and objects. Subjects were extracted directly from the ontology, as well as the predicates. On the other hand, the objects were defined by the content of the *txt* files.

The data used for building the instances and statements can be found here:
https://github.com/mgmartinezl/SDM_Lab3/tree/master/data

The code for building the ABOX in Apache Jena is available at:
https://github.com/mgmartinezl/SDM_Lab3/blob/master/Main.java

C.3. Linking ABOX to TBOX

Apache Jena allowed us to directly link our ABOX together with our TBOX definition. Simple statistics about the ontology created can be obtained executing the following query:

```
SELECT ?tag (count(distinct ?instance) as ?count)
WHERE {?instance rdf:type ?tag}
```

Table 5. Simple statistics about the ontology.

Number of classes	Number of properties	Number of instances
14 classes	4 Property +14 DatatypeProperty +7ObjectProperty **= 25 properties**	15 Company +150 Writer +150 Reviewer +500 SurveyPaper +500 FullPaper +500 DemoPaper +500 ShortPaper +6000 Review +25 AcademicConference +25 AcademicJournal **= 8365 instances**

Note: The query also counts three restrictions that have not been included in the table

C.4. Queries on top of the ontology

1. Find all the authors.

Taking advantage of our TBOX definition:

```
SELECT * WHERE {
?author rdf:type <http://dbpedia.org/ontology/Writer>
}
```

Assuming our TBOX does not exist:

```
SELECT DISTINCT ?author WHERE {
?author ?b ?c FILTER regex(?a, "^http://www.sdmlab3.com/my_ontology#Author")
}
```

2. Find all the properties whose domain is Author

```
SELECT * WHERE {
?a rdfs:domain <http://dbpedia.org/ontology/Writer>
}
```

Assuming our TBOX does not exist:

```
SELECT DISTINCT ?b WHERE {
?a ?b ?c FILTER regex(?a, "^http://www.sdmlab3.com/my_ontology#Author")
}
```

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

```
SELECT * WHERE {
{?a rdfs:domain <http://dbpedia.org/ontology/AcademicConference>}
UNION
{?a rdfs:domain <http://dbpedia.org/ontology/AcademicJournal>}
}
```

Assuming our TBOX does not exist:

```
SELECT DISTINCT ?b WHERE {
{?a ?b ?c FILTER regex(?a, "^http://www.sdmlab3.com/my_ontology#Conference")}
UNION
{?d ?b ?f FILTER regex(?d, "^http://www.sdmlab3.com/my_ontology#Journal")}
}
```

4. Find all the things that Authors have created (either Reviews or Papers)

```
SELECT * WHERE {
{?a rdf:type <http://dbpedia.org/ontology/Article>}
UNION
{?a rdf:type <http://www.sdmlab3.com/my_ontology#ShortPaper>}
UNION
{?a rdf:type <http://www.sdmlab3.com/my_ontology#DemoPaper>}
UNION
{?a rdf:type <http://www.sdmlab3.com/my_ontology#SurveyPaper>}
UNION
{?a rdf:type <http://www.sdmlab3.com/my_ontology#FullPaper>}
UNION
{?a rdf:type <http://www.sdmlab3.com/my_ontology#Review>}
}
```

Assuming our TBOX does not exist:

```
SELECT ?thing WHERE {
{?thing <http://www.sdmlab3.com/my_ontology#hasCorrespondingAuthor> ?author}
UNION
{?thing <http://www.sdmlab3.com/my_ontology#hasReviewer> ?reviewer}
}
```

References

[1] Spare the air. *Particulate matter.* Available at: http://www.sparetheair.org/stay-informed/particulate-matter [Accessed on April 29th, 2019]

[2] Airveda. *Measuring Data - Understanding Particulate Matter and How It Impacts Our Health.* Available at: https://www.airveda.com/blog/Understanding-Particulate-Matter-and-Its-Associated-Health-Impact [Accessed on April 29th, 2019]

[3] CO Knowledge Center. *What is CO And Who's At Risk?.* Available at: https://www.detectcarbonmonoxide.com/co-health-risks/ [Accessed on April 29th, 2019]

[4] AirNow. *Air Quality Guide for Ozone.* Available at: https://www.airnow.gov/index.cfm?action=pubs.aqiguideozone [Accessed on April 29th, 2019]