Group B

Knowledge & Data Engineering Report

Introduction

In this assignment we aimed to combine two different datasets to produce a set of queries that make use of data from both datasets when outputting their results. The first dataset we made use of is the GeoHive dataset; a linked data dataset containing Irish geospatial information such as boundary data per-county, city and even by parishes.

The second dataset is the "Crimes at Garda Stations Level 2010-2016" dataset, a CSV formatted dataset specifying a list of different Garda-Stations, their Divisions and a set of different crimes that occurred within that station's jurisdiction over the past year.

https://data.gov.ie/dataset/crimes-at-garda-stations-level-2010-2016

Approach to Ontology Modelling

The chosen dataset contains information about all the different Garda-Stations in Ireland, the name in both English and Gaelic, the Division that the station falls under, a list of different crimes committed within that stations jurisdiction within a time range and even the geographical Cartesian coordinates for each Station. This location information per station will then be used to link our dataset to GeoHive, by determining in which county each station is.

i. Assumptions Made

Due to the large volume of data within our dataset, we decided to cut down the statistics across the years, focusing instead on just the 2015 data – the most recent year with complete statistics across the whole year.

Ontology modelling assumptions;

a. A County can have many stations
b. A County contains geospatial data
c. A Station can have multiple different crime counts
d. A Station has a location – X & Y coordinates

ii. References to sources used/reused

When designing the ontology, the documentation for both Protégé (https://protegewiki.stanford.edu/wiki/ProtegeDesktopUserDocs) and Jena (https://jena.apache.org/documentation/) were used.

Before implementation, Stanford University Protégé tutorials were also used to fully understand the theory behind ontology modelling. (https://protege.stanford.edu/publications/ontology_development/ontology101.pdf)

When picking the ontologies, we were inspired by the works done by the Maynooth University, providing a GUI that allows users to interact with a map of the Republic of Ireland, showing multiple different statistics based on the users choices. (http://airo.maynoothuniversity.ie/external-content/recorded-crime-monitoring-tool)

iii. Data conversion process

In this project, 2 different datasets are combined to get the expected outputs, GeoHive dataset and the Irish Crime dataset. The GeoHive dataset will provide the information about a county, such as polygonal coordinates. Through this information, an area can be calculated from this polygonal data. The GeoHive dataset is provided in RDF format while the Crime dataset is provided as a CSV document. This dataset is to be uplifted to RDF or equivalent format for combining and querying the data.

In this project, we selected a single year (2015) crime data to avoid the ambiguity and complexity of having multiple similar values for different years. Another reason for this was that feeding the entire corpus for conversion threw an error due to incorrect values or invalid cells in the CSV File.

For converting the CSV document to TTL/RDF format Apache Jena is being used. The CSV file is read, and each row is parsed removing the extra columns in the process. The empty values are replaced with an empty string and saved to an intermediate CSV file. This Intermediate file is then converted to a turtle file (.ttl). Prefixes such as foaf, owl, RDF etc. are added to the converted model using setNsPrefix() function. The resulting model is written to a TTL file so that it can be used in protégé and Jena for ontology modelling.

Overview of Design

i. Description of query interface

Once the ontology and corresponding owl file was created, the next task was to query for results. The queries are again executed using the Apache Jena library.

The created ontology is loaded using the OntModel class. Once the ontology is loaded the query string is created and the querying is performed using the QueryExecutionFactory class – loading the Result into ResultSet.

A stand-alone Java Swing app was created, displaying the possible queries in the question panel. When the user selects a specific query and clicks on the submit button the query is automatically triggered and the output is displayed on the output pane. This user interface was designed with simplicity in mind, acting as an intermediate between user and ontology; allowing any user to run the queries in a very simple manner.

ii. Description of Queries

We have used two datasets Geohive which specifically gives information about the various counties with their co-ordinates.The second dataset we used was the Crimnology dataset which provides information about the various crimes reported to a particular station having co-ordinates X and Y.

In the process of writing the queries we went through the entire ontology and analyzed it's structure in depth. We came across some issues where some properties which were defined in the class defination were not used while defining individuals or some properties which were incorrectly used. So, in the due course of writing queries we refined our ontology. We also faced challenges while running the SPARQL queries through Protege. Hence, we opted for running the queries through JAVA using Apache Jena framework. Finally, we came up with the following list of queries:

1) The station with it's county having highest number of burglaries reported.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select (?a AS ?Station) (?x AS ?County) (?b AS ?NumberOfBurglaryReported)  where {
?x a ns0:County.
?x ns0:hasStations ?a.
?a ns0:hasBurglary ?b.
FILTER(?b!=0)
} ORDER BY DESC(xsd:integer(?b)) LIMIT 1
```

2) The station with it's county having highest number of murders reported.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select (?a AS ?Station) (?x AS ?County) (?b AS ?NumberOfMurdersReported)  where {
?x a ns0:County.
?x ns0:hasStations ?a.
?a ns0:hasMurder ?b.
FILTER(?b!=0)
}ORDER BY DESC(xsd:integer(?b)) LIMIT 1
```

3)The station with it's county having highest number of thefts reported.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select (?a AS ?Station) (?x AS ?County) (?b AS ?NumberOfTheftsReported)  where {
```

```
?x a ns0:County.
?x ns0:hasStations ?a.
?a ns0:hasTheft ?b.
FILTER(?b!=0)
}ORDER BY DESC(xsd:integer(?b)) LIMIT 1
```

4)The station with it's county highest number of Dangerous crimes reported

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select (?a AS ?Station) (?x AS ?County) (?b AS ?NumberOfDangerousReported) where {
?x a ns0:County.
?x ns0:hasStations ?a.
?a ns0:hasDangerous ?b.
FILTER(?b!=0)
}ORDER BY DESC(xsd:integer(?b)) LIMIT 1
```

5) The number of stations in a county.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select (?x AS ?County) (COUNT(?a) AS ?TotalNumberOfStations) WHERE{
?x a ns0:County.
?x ns0:hasStations ?a
} GROUP BY ?x
```

6) The list of all stations with their county and total number of crimes reported in descending order.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select (?a AS ?Station) (?x AS ?County) (?b AS ?TotalCrimes)  where {
?x a ns0:County.
?x ns0:hasStations ?a.
?a ns0:totalCrime ?b
} ORDER BY DESC(xsd:int(?b))
```

7) The list of station their county and the type of crimes reported.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select  (?a AS ?Station) (?x AS ?County) (?b AS ?TypesOfCrimes)  where {
?x a ns0:County.
?x ns0:hasStations ?a.
?a ns0:hasCrime ?b}
```

8) The list of county and the number of counties adjacent to them.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select  (?x AS ?County) (COUNt(?a) AS ?AdjacentCounty)   where {
?x a ns0:County.
?x ns0:adjacentTo ?a
} GROUP BY ?x
```

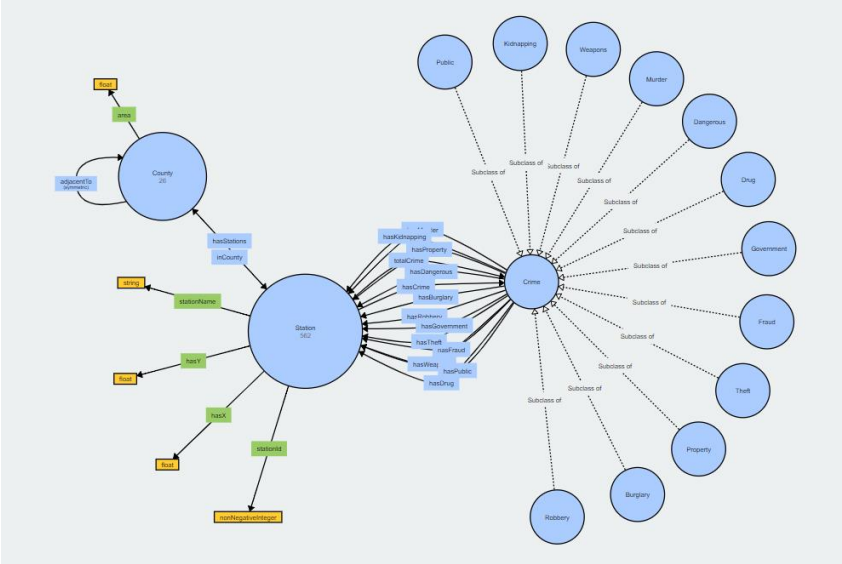
9) The name of the county in English and Gaelic

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select  (?x AS ?County) (?a AS ?Gaelic) (?b AS ?EngName) where {
?x a ns0:County.
?x rdfs:label ?a.
?x rdfs:label ?b.
FILTER(langMatches(lang(?a), \"GA\"))
FILTER(langMatches(lang(?b), \"EN\"))}
```

10) The station and county with second most total crimes reported.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns0: <http://lab.Jena.Kdeg.ie/CrimeOntology.owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT *  WHERE {
select (?a AS ?Station) (?x AS ?County) (?b AS ?NumberOfDangerousReported) where {
?x a ns0:County.
?x ns0:hasStations ?a.
?a ns0:totalCrime ?b.
FILTER(?b!=0)
}ORDER BY DESC(xsd:integer(?b)) LIMIT 2}
ORDER BY (?NumberOfDangerousReported) LIMIT 1
```

iii. VOWL Representation



Challenges faces while ontology modelling or creating queries

- Deciding whether a class/property was necessary
- Deciding whether the types of crimes should be listed as classes or individuals
- Testing properties like symmetricity and transitivity of properties
- Finalizing which tool to use when developing the ontology (Protégé or Jena)
- Getting the reasoner to work properly and execute in reasonable time
- Deciphering the errors thrown by the reasoner
- Unable to find a relationship between division and county. They were initially assumed to be the same
- Finding the link between the two datasets
- Introducing transitive and symmetric properties

Conclusion

Our ontology model could successfully link the two different chosen datasets through the use of station coordinates. Using these coordinates, we could then identify in which County each station was and extrapolate that data in our queries to determine which county was the most dangerous in 2015.While inverse and symmetric properties were defined and used in our ontology, no transitive properties were used. Our initial proposal was to link counties to the divisions they each contain and then link divisions to all the stations within them. Using this approach, the transitive property could have easily been used to infer stations within each county. However, the dataset only provides location information per station, making all the provided Division information would have been essentially useless for us. The dataset explains that Divisions are the region in which different stations are, but since only stations have location information, we had to use stations to link directly to counties. For this reason, we also decided to remove Divisions from our ontology since there was no real benefit of having this extra information.