Technical report

Group: Team O

Patrick Hughes, Caoimhe Mooney, Caleb Teo, Arne Philipeit, Alexandra Silva

Description of Competency Questions that ontology answers

For our application, we investigated different statistics for the Country of Ireland across different counties and years. Our application can handle queries that can return multiple statistics from crime, housing prices, average salaries, weather and homelessness for different Counties and years. This application can then help provide information on statistics and make correlations or connections between the different statistics. We have 10 Competency Questions in which our ontology can answer. These 10 questions are spanned across multiple databases and return multiple statistics for county and year.

Question 1: *"What is the number of homeless people in the County with the highest housing prices in 2014?"*

Description: This question will return the number of homeless people with the county that has the highest housing prices. This question can show if there is a high or low number of homelessness depending on the housing price of a County. The result will be the number of homeless, the corresponding county and the housing price of that county.

Question 2: *"What was the average precipitation of Dublin in the year with the highest number of thefts?"*

Description: This query will return the average precipitation and the number of crimes related to 'Theft and related offences. From this query we can investigate into the connection between theft crimes and the rainfall in Dublin. This could also be extended to any other county.

Question 3: "*What was the average cost of housing when salaries were at their highest?"*

Description: This query will return the average cost of housing when salaries are at its highest. The query will return the average salary and the year

Question 4: *"Given that the average rainfall in a county is 89.58mm what counties has values below this and from these counties which has the lowest rate of homelessness?"*

Description: This query will return the number of homeless in a county and corresponding county with average rainfall less than the given rainfall. This is to see if there is less homelessness for low average rainfall counties.

Question 5: *"What were the average salary and housing prices in Dublin in 2010 vs 2016?"*

Description: This question returns the average mean salary and housing price for a county in 2 different years. This can display the change in the statistic values between years.

Question 6: *"In 2015, did the county with the highest salary have more kidnappings than the county with the lowest salary?"*

Description: This is a comparison query in which shows the difference in number of kidnappings between the counties of the highest salary and lowest salary. The will return the county with the highest salary, the corresponding salary, corresponding number of kidnappings, the county with the lowest salary, the corresponding salary, and the corresponding number of kidnappings.

Question 7: *"In the year with the highest average salary how many homeless people were there across the county?"*

Description: This query will return the number of homeless in a given county with the highest salary and the associated year.

Question 8: *"What is the homelessness in Dublin and Crime rate of theft in 2014 vs 2019"*

Description: This question is to see if there is an increase in homelessness related to crimes overtime. The query will return the number of homelessness and the number of thefts in both years.

Question 9: ***"What is the housing prices vs crime over years and counties?"***

Description: This query will return the housing price and crime rate of 2 different years in a county. In the case where there are multiple divisions per county this query will also aggregate the number of crimes.

Question 10: "Which Counties saw a year over year increase in housing prices while not seeing an increase in homelessness?"

Description: This query will return each county that had an increase in housing prices but no increase in homelessness. This will return the county, the year, the corresponding housing price for that year, the increased housing price of the next year, the homeless count for that year and the decreased/equal homeless count for the next year. This query compares one year to the next.

Description of datasets selected for application

The datasets we selected for the application include datasets on crime, housing prices, salaries, weather and homelessness. The crime dataset outlines the number of different types of crimes carried out in different Garda divisions around Ireland in a given year. The crime has the number of occurrences that has been reported for each division. A division is usually related to a County. In some Counties there is a shared Garda division for the 2 counties i.e. Cavan and Monaghan has the same Garada Division. There are also counties that have multiple Garda Divisions i.e. Cork has Cork City, North and West Garda Divisions. The housing prices dataset outlines the volume of housing sales, the mean price of housing sales and the value of housing sales in different counties in a given year. The salaries dataset outlines the average salary of people in different counties in a given year. The weather dataset outlines the average monthly precipitation, average monthly greatest daily rainfall, average monthly number of rain days(days with greater than 0.2mm) and average number of wet days(greater than 1.0mm) in mm per year. The homelessness dataset outlines the number of homeless people in different counties in a given year. It also provides information on the gender of these people and their age range.

Assumptions made

Initially the homelessness dataset was divided by regions such as south-east, south-west etc. instead of counties. In order to divide the regions by county instead we made the assumption that the number of homeless people was broken up between the counties in each region by the ratio of homeless people per county for 2014 and 2015. For example, the number of homeless in the mid-east region was spilt over Kildare, Meath and Wicklow by the ratio of homeless people in those counties in 2016. From 2016 on the numbers of people were provided per county and could be averaged over the 12 months of the year. These numbers however were not available for gender and age range so the number of people was again divided by the ratio, this time of male to female or each age range.

When choosing datasets for weather we made a compromise between datasets that went back a long time and datasets that had a large number of weather categories e.g. sunshine duration, mean maximum temperature.We assumed that prioritizing the time range of queries was more important than the number of weather categories. It turns out that having a broader range of weather categories would have made for more interesting queries. Especially considering that none of the other datasets spanned back nearly as far as the 1940s so this data was largely useless for cross dataset queries. This was a bad assumption we made.

In the crime dataset, there were some assumptions made for the creation of our ontology. The first assumption was that there was a category for every crime. Twelve crime categories were created to fit each crime. The twelve categories are referred below. These were created as subclasses of the Crime class. In our approach to the ontology we assumed that each subcategory would have additional attributes to each crime. This is the second assumption in which each subcategory crimes have additional attributes.

Subcategories	**Additional Attributes**	**Subcategories**	**Additional Attributes**
MurderDeath	ModeGenderVictim	Transport	ModeVehiclesReported
Sexual Offences	NumOfCaseRelatingToDemesticHarm	Kidnapping	AverageAgeOfKidnapped
AttemptedMurder	NumOfAssaultsRelatingToDrugs	Theft	AverageValueStolen
Driving	DamagesToPublicProperty	Drugs	AmountOfDrugsConfiscated
Harm	ModeReportedWeapon	Weapon	ModeWeaponReported
Offences	NumOfReportedArrests	PropertyDamage	AverageCostOfDamages

Each of the additional attributes are fabricated and have no real value. In our application the data is either 'none' or 0 depending on the data type. This was designed to mimic realistic application with these subcategories of crime.

References to vocabulary sources

We used dbpedia for Year, http://dbpedia.org/ontology/Year. Unfortunately we couldn't find any other ontologies to reuse.

Discussion of your data mapping process

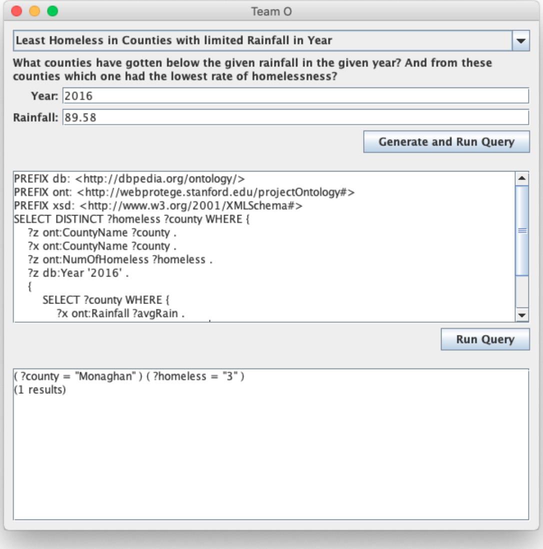
The first step in our data mapping process was ensuring that all our datasets followed the same domain in that the values were all yearly over each county. This ensured that the datasets could all be queried via years and counties. We created a parent class called Stats which had each dataset as a subclass. Stats was then related to both county and year so each dataset would inherently also be related to both county and year. While weather, homelessness, salaries and housing prices are all related to counties, crime was a little more complex in that it was related to divisions and these divisions were related to counties. We created a parent class called Area with subclasses county and division in order to deal with this.

Explanation of use of inverse, symmetric and transitive properties

The inverse properties we implemented in our ontology were *'nextYear'* and *'previousYear'*. An inverse property is one that can be obtained by getting another property and just changing its direction. *'nextYear'* is an inverse property of *'previousYear'* because if we know the value of *'nextYear'* then we also know the value of *'previousYear'* and vice versa.
The symmetric property we implemented in our ontology was *'adjacentTo'*. A symmetric property is one whose direction does not matter, for instance the property will relate A to B in the same way it relates B to A. *'adjacentTo'* is a symmetric property because Dublin is *'adjacentTo'* Wicklow in the same way that Wicklow is *'adjacentTo'* Dublin.
The transitive property we implemented in our ontology was *'has'*. If a transitive property holds from A to B and from B to C, it holds from A to C. In our case, a geographical area can contain (*'has'*) smaller geographical areas. This works over several levels. The country Ireland *'has'* county Dublin which *'has'* a division Dublin South. Therefore, because *'has'* is transitive, Ireland *'has'* Dublin South. This simplifies queries that work with geographical areas, as the same requirements can also be applied to and aggregated for smaller contained geographical areas.

Overview of Design

Description of Application Query Interface



What counties have gotten below the given rainfall in the given year? And from these counties which one had the lowest rate of homelessness?

Year: 2016

Rainfall: 89.58

```
PREFIX db: <http://dbpedia.org/ontology/>
PREFIX ont: <http://webprotege.stanford.edu/projectOntology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT DISTINCT ?homeless ?county WHERE {
   ?z ont:CountyName ?county .
   ?x ont:CountyName ?county .
   ?z ont:NumOfHomeless ?homeless .
   ?z db:Year '2016' .
   {
      SELECT ?county WHERE {
         ?x ont:Rainfall ?avgRain .
```

```
( ?county = "Monaghan" ) ( ?homeless = "3" )
(1 results)
```

The Application Query Interface is kept simple. The results of the last SPARQL query are listed at the bottom. In the middle are a text field and button ("Run Query") to edit and run queries manually.

The combo box at the top has a selection of our prepared queries. Directly below is a longer description of the selected query, followed by zero or more text fields to fill specific parts of the query. In the screenshot above, the user can enter a year and rainfall amount (in mm) that will be included in the query. The button "Generate and Run Query" then generates a query with the given values, updates the text area below with the generated query, runs the query, and displays the results.

The Application uses Apache Jena to load our model (ttl file) and run SPARQL queries. The User Interface was built with the Java Swing library.

Description of Queries

Query 1: What was the number of homeless people in the county with the highest housing prices in the given year?

In the above query we run a nested subquery that allows us to isolate the highest housing price for the given year, in this case 2014. We match triples according to year, price, county name and housing price type. The price values returned from matching triples are ordered in descending order using "ODER BY DESC" and the first element is chosen (The highest value for price). This price variable is within the scope of the outer SELECT statement and is used in triple matching to allow us to link ?z and ?x by county. This results in 3 variables with one value each for number of homeless(?num) in county(?county) with highest housing price(?price).

Query 2: What was the average precipitation of the given County in the year with the highest number of thefts?

Nested subquery queries crime instance(?x) according to year, county, number of crime and type of crime committed. We cast the number of crime committed to a float, order the returned values in descending order and chose the first value(the highest value). The value for number of crime committed and year they were committed in are inside the scope of the outer query. We use the year variable to query for rainfall in Dublin on that year and return the values for year(which was filtered according to year with most crime) and rain.

Query 3: What was the average cost of housing when salaries were at their highest?

Nested subquery queries salary instance (?x) using salary value, year and given county(in this case Wicklow). Again we order by descending and pick the first value to return the highest value for salary. The highest value for salary and year it occurred in is passed on to the outer query. We use the year that the highest salary occurred in to find the housing prices in the same county and return the cost of housing in Wicklow in the year salaries were at their highest.

Query 4: What counties have gotten below the given rainfall in the given year? And from these counties which one had the lowest rate of homelessness?

Nested subquery queries weather instance (?x) using rainfall, county name and year. We filter these values to only include weather instances with rainfall less than the given average (89.58). We return the counties that have below the given rainfall to the outermost query. We link the counties that have below average rainfall with homelessness instances and order these homelessness instances in ascending order to find lowest number of homeless. We return the lowest number of homeless and the county it occurred in.

Query 5: What were the average salary and housing prices in county in 2010 vs 2016?

Nested subquery queries salaries in Dublin in 2010 and 2016 using salary instances (?a, ?b). These values are returned to the outer query. We then query the price of housing in Dublin 2010 and 2016 and return all four values. The inner variables did not link the variables in the outer query, but the result of the query is still interesting and involves more than 1 dataset.

Query 6: Does the county with the highest salary have more kidnappings than county lowest salary?

For this query we first get the county with the highest salary in a specific year. We then use this county value and the same year value to find the corresponding kidnapping rate. The same is done for the county with the highest salary. The salaries and kidnappings for both counties are outputted. The year can be chosen by the user.

Query 7: Number of Homeless in year with the highest Salary

Firstly, the year with the highest salary was found. The highest salary was found in a specific county. The county can be chosen by the user. The year and county are then used to find the number of homeless people in that year. This query can have some issues as homelessness data only goes back as far as 2014 while Salary goes back much further.

Query 8: What is the homelessness in Dublin and Crime rate of theft in 2014 vs 2019

The crime rate in the selected county is found for the two years selected first. Then the homelessness numbers are found using the same parameters. The county and years can be determined by user input.

Query 9: Housing prices vs Crime

Firstly the number of crimes for burglary related offences is found for the selected county and years. The sum is taken of the crime numbers as counties with multiple divisions print out multiple numbers, one per division. The housing prices are then found for the same parameters as crime, specifying executions. The county and years can be determined by user input.

Query 10: Counties with Year over Year Housing Price increase without Homelessness increase

This query uses filters to find counties where the difference between the housing prices in two years increases but the difference between homelessness numbers stay the same or decrease. The housing price type is determined by user input.

Challenges faced while modelling ontology or creating queries and mappings

Designing ontology

We encountered a few issues in the initial stages of designing our ontology. Our main issues stemmed from each team member having a different understanding of how the ontology should be laid out and what layout of information makes the most logical sense. This resulted in our initial ontology design taking longer than it should have to finalise.

Uplifting

In relation to the uplifting, most datasets were completed without many complications. There were a number of challenges to overcome in cleaning the data in which everyone was able to overcome by writing scripts or manually aggregating the datasets in a logical manor. There were some issues with uplifting the homeless data with the age ranges columns. This issue was due to the input of the column name and we had to adjust the csv so that JUMA could read in the column name correctly. There were challenges in uplifting the crime dataset as on first attempts. In the JUMA for crime there were incorrectly label classes and the instances of the data was grouped together. Once the incorrect labels were spotted then the correction was simple to make. For the grouped instances, we figured out that this caused issues with our sparql queries and to fix the issue in the JUMA tool we created a different instance ID. This gave us separated data instances for the crime dataset. This created the data instances to match our designed ontology.

WIDOCO

Using WIDOCO to document our project presented many issues. One very prominent issue was the length of time that it took to parse our rdf into the ttl file we needed for our app. As mentioned in our issues with uplifting the crime dataset too a few tries to get right and by the end was a very large file. When run on a windows laptop after over an hour it still hadn't finished running and it eventually worked after approximately 30 minutes on a mac. After many issues we eventually sought some help and discovered the main deliverable was a pdf rather than using WIDOCO which cleared up many of our issues.

SPARQL

One major challenge we faced while writing our SPARQL queries was becoming familiar with the syntax. Apart from the self-directed exercises in class, this was the first time any of us had to write in SPARQL. To overcome this challenge we all took the time to study the lecture notes and began practicing writing simple queries that related to only one dataset before moving onto the more complex queries wrote for the assignment.

Conclusion:

One weakness we all found from out ontology model was the lack of reused ontologies. Unfortunately, we were only able to reuse one, http://dbpedia.org/ontology/Year. If we were to continue working on this ontology we would find a way to include more existing ontologies. Our queries are applied over the many datasets and derived classes in our final turtle file. The span of our queries is wide and allows us to make interesting hypothesis about potential correlations in our data. For example, our data might reinforce the believe that higher salaries will always result in higher housing prices or that an increase in housing prices directly results in an increase in the number of homeless people. The potential for our queries is limited by the range of our data and our syntactic knowledge of SPARQL.

This leads me to the weaknesses of our queries. Ask I mentioned already our queries produce meaningful results but, in some cases, brute force query methods are used to achieve these results. If we were more familiar with SPARQL we might have used UNION or OPTIONAL to identify graph patterns. Or we could have used ASK to identify the presence of triples in a graph. Further use of aggregates such as MAX, MIN and AVG (which were introduced in SPARQL 1.1) instead of ordering by and limiting would have cleaned up our queries.

Overall, I feel our queries display the usefulness of our final linked data product very well.