Assignment 1 - RDF and Sparql

Zefan Liang

School of Electrical Engineering and Computer Science

University of Ottawa

27th January, 2019

CSI 5180 Topics in AI - Ontologies and Semantic Web

Caroline Barrière

Table of content

1 Introduction

This report contains the whole content of the requirements, including 6 steps to learn about RDF model, Jena framework, Sparql and DBpedia.

2 Creating my small RDF model

The first step is to create a small RDF of myself and two of my friends and this is the content of my turtle file.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix vCard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix ex: <http://csi5180-example.org/> .

ex:LZF
        vCard:FN "Zefan Liang" ;
        vCard:N [ vCard:Family "Liang" ;
        vCard:Given "Zefan"] ;
        vCard:Email "583190078@qq.com";
        vCard:locality      dbr:Ottawa ;
        ex:studies    ex:CSI5137 ;
        ex:studies    ex:CSI5175 ;
        ex:studies    ex:CSI5180 ;
        foaf:knows    ex:DXX ;
        foaf:knows    ex:WZX .
ex:CSI5137     vCard:FN "CSI5137";  ex:offeredAt dbr:University_of_Ottawa .
ex:CSI5175     vCard:FN "CSI5175";  ex:offeredAt dbr:University_of_Ottawa .
ex:CSI5180     vCard:FN "CSI5180";   ex:offeredAt dbr:University_of_Ottawa .
ex:DXX
        vCard:FN "Xiaoxiang Dong" ;
        vCard:N [ vCard:Family "Dong" ;
        vCard:Given "Xiaoxiang";] ;
        vCard:Email "821440183@qq.com";
        vCard:locality      dbr:Xi’an ;
        ex:studies    ex:P12P182912 ;
        ex:studies    ex:P12P182919 ;
        ex:studies    ex:G14G185919 .
ex:P12P182912    vCard:FN "P12P182912";  ex:offeredAt dbr:Northwest_University .
```

```
ex:P12P182919    vCard:FN "P12P182919";  ex:offeredAt dbr:Northwest_University .
ex:WZX
        vCard:FN "Zhixiang Wang" ;
        vCard:N [ vCard:Family "Wang" ;
        vCard:Given "Zhixiang";] ;
        vCard:Email "2856193308@qq.com";
        vCard:locality      dbr:Xi'an ;
        ex:studies    ex:M14M12005 ;
        ex:studies    ex:M14M12007 ;
        ex:studies    ex:U10G11001 .
ex:M14M12005
    vCard:FN "M14M12005 ";  ex:offeredAt dbr:Northwestern_Polytechnical_University .
ex:M14M12007
    vCard:FN "M14M12007";  ex:offeredAt dbr:Northwestern_Polytechnical_University .
```

Figure 1 - content of file LZF.ttl (turtle file)

The data model includes my full name, family name, given name, email, location and three courses that I have registered in University of Ottawa in this semester. It also includes my two friends with the same properties and they are not studying in Ottawa now.

3 Testing validators and converters

In this section, by using converter, http://rdfvalidator.mybluemix.net/ , to convert my turtle file into an RDF file. The result is as follows : (For the convenience of viewing, I Screenshot and put it together).

Output:

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#"
    xmlns:dbr="http://dbpedia.org/resource/"
    xmlns:ex="http://csi5180-example.org/"
    xmlns:foaf="http://xmlns.com/foaf/0.1/" >
  <rdf:Description rdf:nodeID="A0">
    <vCard:Given>Xiaoxiang</vCard:Given>
    <vCard:Family>Dong</vCard:Family>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/M14M12007">
    <ex:offeredAt rdf:resource="http://dbpedia.org/resource/Northwestern_Polytechnical_University"/>
    <vCard:FN>M14M12007</vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/M14M12005">
    <ex:offeredAt rdf:resource="http://dbpedia.org/resource/Northwestern_Polytechnical_University"/>
    <vCard:FN>M14M12005 </vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/CSI5175">
    <ex:offeredAt rdf:resource="http://dbpedia.org/resource/University_of_Ottawa"/>
    <vCard:FN>CSI5175</vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/LZF">
    <foaf:knows rdf:resource="http://csi5180-example.org/WZX"/>
    <foaf:knows rdf:resource="http://csi5180-example.org/DXX"/>
    <ex:studies rdf:resource="http://csi5180-example.org/CSI5180"/>
    <ex:studies rdf:resource="http://csi5180-example.org/CSI5175"/>
    <ex:studies rdf:resource="http://csi5180-example.org/CSI5137"/>
    <vCard:locality rdf:resource="http://dbpedia.org/resource/Ottawa"/>
    <vCard:Email1>583190078@qq.com</vCard:Email1>
    <vCard:N rdf:nodeID="A1"/>
    <vCard:FN>Zefan Liang</vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/CSI5137">
    <ex:offeredAt rdf:resource="http://dbpedia.org/resource/University_of_Ottawa"/>
    <vCard:FN>CSI5137</vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:nodeID="A2">
    <vCard:Given>Zhixiang</vCard:Given>
    <vCard:Family>Wang</vCard:Family>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/P12P182919">
    <ex:offeredAt rdf:resource="http://dbpedia.org/resource/Northwest_University"/>
    <vCard:FN>P12P182919</vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/DXX">
    <ex:studies rdf:resource="http://csi5180-example.org/G14G185919"/>
    <ex:studies rdf:resource="http://csi5180-example.org/P12P182919"/>
    <ex:studies rdf:resource="http://csi5180-example.org/P12P182912"/>
    <vCard:locality rdf:resource="http://dbpedia.org/resource/Xian"/>
    <vCard:Email1>821440183@qq.com</vCard:Email1>
    <vCard:N rdf:nodeID="A0"/>
    <vCard:FN>Xiaoxiang Dong</vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/CSI5180">
    <ex:offeredAt rdf:resource="http://dbpedia.org/resource/University_of_Ottawa"/>
    <vCard:FN>CSI5180</vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/P12P182912">
    <ex:offeredAt rdf:resource="http://dbpedia.org/resource/Northwest_University"/>
    <vCard:FN>P12P182912</vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:about="http://csi5180-example.org/WZX">
    <ex:studies rdf:resource="http://csi5180-example.org/U10G11001"/>
    <ex:studies rdf:resource="http://csi5180-example.org/M14M12007"/>
    <ex:studies rdf:resource="http://csi5180-example.org/M14M12005"/>
    <vCard:locality rdf:resource="http://dbpedia.org/resource/Xian"/>
    <vCard:Email1>2856193308@qq.com</vCard:Email1>
    <vCard:N rdf:nodeID="A2"/>
    <vCard:FN>Zhixiang Wang</vCard:FN>
  </rdf:Description>
  <rdf:Description rdf:nodeID="A1">
    <vCard:Given>Zefan</vCard:Given>
    <vCard:Family>Liang</vCard:Family>
  </rdf:Description>
</rdf:RDF>
```


Figure 2 - LZF.ttl (turtle file) converted to RDF format

4 Visualization

By using online visualization program (http://visgraph3.org/), I transformed the RDF into a graph to visualize it easily.

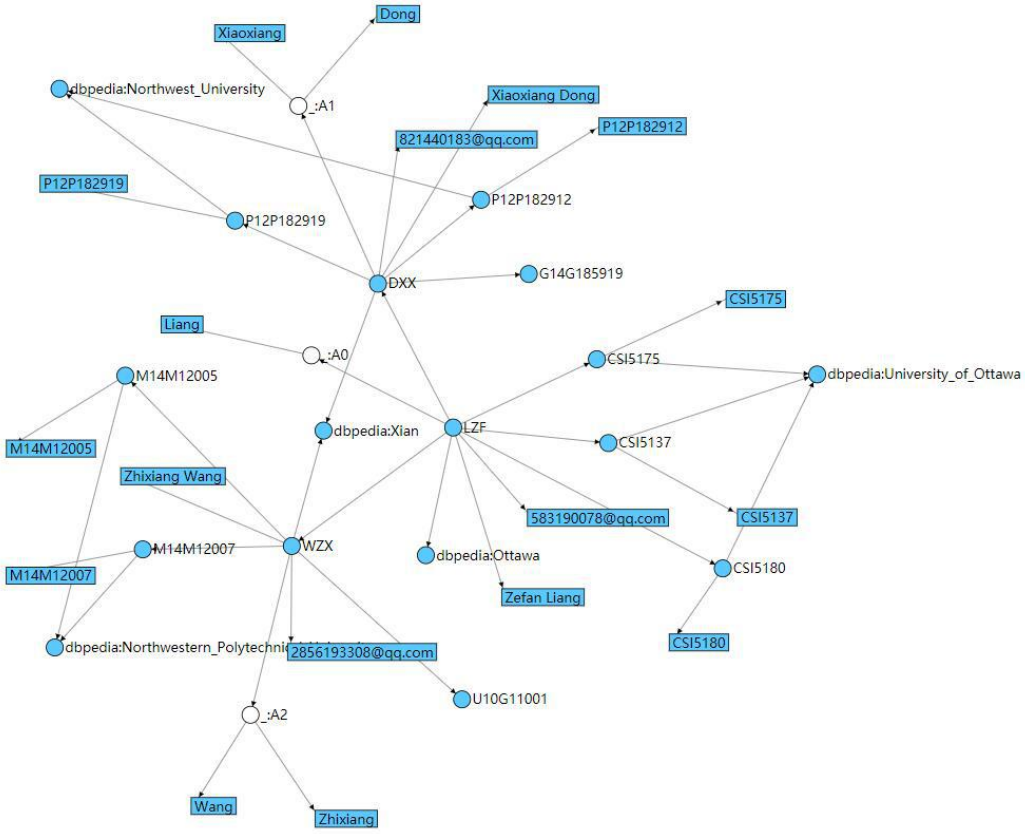

Figure 3 - LZF.rdf uploaded to the visualizer

5 Querying your model within Jena

First, I downloaded Jena and included it in my Java Project.

The prefix strings are as follow, they are all the prefix strings in this section.

And the development environment is eclipse 4.5.1.

```
public static String PREFIX_STRING=
    "PREFIX vCard: <http://www.w3.org/2001/vcard-rdf/3.0#>"+
    "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>"+
    "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
    "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"+
    "PREFIX foaf:  <http://xmlns.com/foaf/0.1/>"+
    "PREFIX dc: <http://purl.org/dc/elements/1.1/>"+
    "PREFIX dbr: <http://dbpedia.org/page/>"+
    "PREFIX ex: <http://csi5180-example.org/>"+
    "PREFIX dbo: <http://dbpedia.org/ontology/>"+
    "PREFIX dbp: <http://dbpedia.org/property/>";
```

Figure 4 - prefix strings of section4

Q1:List courses that I take at the University

```
public static String FIND_MY_COURSES =
        " SELECT  ?Course"+
        " WHERE "+
        "{ ex:LZF ex:studies ?o ."+
        " ?o  vCard:FN ?Course . }";
```

Figure 5 - The Sparql queries in Q1.

```
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileM
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more
Course  CSI5137

Course  CSI5175

Course  CSI5180
```

Figure 6 - The result of Q1.

Q2:List the given names of my friends.

```
public static String FIND_MY_FRIENDS =
        " SELECT DISTINCT ?Givenname"+
        " WHERE "+
        "{ ex:LZF foaf:knows ?o ."+
        " ?o  vCard:N ?nodeID . "+
        "?nodeID vCard:Given ?Givenname . }";
```

Figure 7 - The Sparql queries in Q2.

```
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileManager
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Givenname       Xiaoxiang

Givenname       Zhixiang
```

Figure 8 - The result of Q2.

Q3:List the organizations (schools) where my friends work or study.

```
public static String FIND_MY_FRIENDS_University =
        " SELECT DISTINCT ?School"+
        " WHERE "+
        "{ ex:LZF foaf:knows ?o ."+
        " ?o  ex:studies ?Course . "+
        "?Course ex:offeredAt ?School . }";
```

Figure 9 - The Sparql queries in Q3.

```
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileManag
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more inf
School  http://dbpedia.org/resource/Northwest_University

School  http://dbpedia.org/resource/Northwestern_Polytechnical_University
```

Figure 10 - The result of Q3.

Q4:List the emails of my friends.

```
public static String FIND_MY_EMAIL =
        " SELECT DISTINCT ?Fullname ?Email "+
        " WHERE "+
        "{ ex:LZF foaf:knows ?o ."+
        " ?o vCard:FN ?Fullname . "+
         " ?o vCard:Email ?Email . }";
```

Figure 11 - The Sparql queries in Q4.

```
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileManager
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info
Fullname        Xiaoxiang Dong

Email   821440183@qq.com

Fullname        Zhixiang Wang

Email   2856193308@qq.com
```

Figure 12 - The result in Q4.

Q5:List the courses my friend registered in their universities

```
public static String FIND_MY_FRIENDS_Courses =
        " SELECT DISTINCT ?Fullname ?Coursename "+
        " WHERE "+
        "{ ex:LZF foaf:knows ?o ."+
        " ?o vCard:FN ?Fullname . "+
        " ?o ex:studies ?Course . "+
        "?Course vCard:FN ?Coursename}";
```

Figure 13 - The Sparql queries in Q5.

```
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Fullname        Xiaoxiang Dong

Coursename      P12P182912

Fullname        Xiaoxiang Dong

Coursename      P12P182919

Fullname        Xiaoxiang Dong

Coursename      G14G185919

Fullname        Zhixiang Wang

Coursename      M14M12005

Fullname        Zhixiang Wang

Coursename      M14M12007

Fullname        Zhixiang Wang

Coursename      U10G11001
```

Figure 14 - The result in Q5.

6 Explore DBpedia Sparql endpoint

By going to http://dbpedia.org/sparql to access the SPARQL endpoint, the questions, Sparql queries and results are as follow:

Q1. Show all the universities in Ottawa.

The Sparql queries in Q1:

```
select distinct ?result
where
{ ?university dbo:city dbr:Ottawa .
  FILTER regex(?university , "University")
  ?university dbp:uniname ?result
}
LIMIT 30
```

UniversityinOttawa
"University of Ottawa"^^<http://www.w3.org/1999/02/22-rdf-syntax-ns#langString>
"Carleton University"^^<http://www.w3.org/1999/02/22-rdf-syntax-ns#langString>

Figure 15 - The result of Q1.

Q2. Show cities in Canada more populated than Ottawa.

The Sparql queries in Q2:

```
select DISTINCT MIN(?name) ?popu
where
{ ?city dbo:country dbr:Canada .
  ?city rdf:type dbo:City .
  ?city dbo:populationTotal ?popu .
  dbr:Ottawa dbo:populationTotal ?Ottawapo .
  FILTER (?popu > ?Ottawapo) .
  ?city  foaf:name ?name .
}
 LIMIT 1000
```

callret-0	popu
"Montreal"@en	"1649519"^^<http://www.w3.org/2001/XMLSchema#nonNegativeInteger>
"Toronto"@en	"2615060"^^<http://www.w3.org/2001/XMLSchema#nonNegativeInteger>

Figure 16 - The result of Q2.

Q3. Show musicians born (use dbo:MusicalArtist) in Ottawa who are older than 40 years old.

The Sparql queries in Q3:

```
select DISTINCT ?name  MIN(year(xsd:dateTime(?birthDate)) as ?birthYear)
where
{
  ?person dbo:birthPlace dbr:Ottawa .
  ?person rdf:type dbo:MusicalArtist .
  ?person foaf:name ?name .
  ?person dbo:birthDate ?birthDate .
  filter($birthDate < "1979-01-01"^^xsd:date).
  bind(replace(str($birthDate),"(\\d+)-\\d*-\\d*", "$1") as ?year)
  filter(xsd:integer(?year) < 1979)
}
 LIMIT 1000
```

name	callret-1
"Saukrates"@en	1978
"Marjorie Elliott de Oduber"@en	1925
"Paul Anka"@en	1941
"Chris Thompson"@en	1971
"Valdy"@en	1945
"Kathleen Edwards"@en	1978
"Harvey Glatt"@en	1934
"Brad Sucks"@en	1976
"Edwin Orion Browne11"@en	1964
"Socalled"@en	1976
"Stanley Sheldon"@en	1950
"Marjorie Elliott Sypher"@en	1925
"Sneezy Waters"@en	1945
"Bernie LaBarge"@en	1953
"Gene Cornish"@en	1944
"Bruce Cockburn"@en	1945
"Terry Carisse"@en	1942
"Richard Parry"@en	1977
"Jeremy Gara"@en	1978
"Chris McKhool"@en	1968
"Brad Turcotte"@en	1976
"David Murray Brockie"@en	1963
"Glen Drover"@en	1969
"William Hawkins"@en	1940
"Jordan O'Connor"@en	1972
"Andrew Scott"@en	1967
"Oderus Urungus"@en	1963
"Jeff Waters"@en	1966
"Winifred Bambrick"@en	1892
"Richard Reed Parry"@en	1977
"Alanis Morissette"@en	1974
"Les Emmerson"@en	1944

Figure 17 - The result of Q3 (For the convenience of viewing, I Screenshot and put it together).

Q4.Find the “C9 Alliance” Universities of China

The Sparql queries in Q4:

```
select DISTINCT ?university
where
{
  ?university dbo:type   ?p .
  FILTER (?p IN (dbr:Public_university , dbr:Public_University) ) .
  ?university dbo:affiliation dbr:C9_League .
}
 LIMIT 1000
```

university
http://dbpedia.org/resource/Xi'an_Jiaotong_University
http://dbpedia.org/resource/Fudan_University
http://dbpedia.org/resource/Peking_University
http://dbpedia.org/resource/Tsinghua_University
http://dbpedia.org/resource/Harbin_Institute_of_Technology
http://dbpedia.org/resource/Nanjing_University
http://dbpedia.org/resource/Zhejiang_University
http://dbpedia.org/resource/University_of_Science_and_Technology_of_China
http://dbpedia.org/resource/Shanghai_Jiao_Tong_University

Figure 18 - The result of Q4 .

Q5.Select the team-members of Borussia_Dortmund whose position is midfielder, whose height is taller than 1.80 meters and who is German footballer.

The Sparql queries in Q5:

```
select DISTINCT ?teammember
where
{
  ?teammember dbo:team dbr:Borussia_Dortmund .
  ?teammember dbo:position  dbr:Midfielder .
  ?teammember dbo:height ?height .
  FILTER (?height >= 1.8) .
  ?teammember dct:subject dbc:German_footballers .
}
 LIMIT 1000
```

teammember	http://dbpedia.org/resource/Thomas_Helmer
http://dbpedia.org/resource/Sven_Bender	http://dbpedia.org/resource/Dominik_Nothnagel
http://dbpedia.org/resource/Kevin_Großkreutz	http://dbpedia.org/resource/Giovanni_Federico
http://dbpedia.org/resource/Marco_Reus	http://dbpedia.org/resource/Thomas_Franck_(footballer)
http://dbpedia.org/resource/Michael_Zorc	http://dbpedia.org/resource/Pascal_Stenzel
http://dbpedia.org/resource/Frank_Riethmann	http://dbpedia.org/resource/Andreas_Möller
http://dbpedia.org/resource/Tim_Gutberlet	http://dbpedia.org/resource/Steffen_Freund
http://dbpedia.org/resource/Peter_Fraßmann	http://dbpedia.org/resource/İlkay_Gündoğan
http://dbpedia.org/resource/Dennis_Brinkmann	http://dbpedia.org/resource/Christian_Nerlinger
http://dbpedia.org/resource/Enis_Alushi	http://dbpedia.org/resource/Rico_Benatelli
http://dbpedia.org/resource/Ingo_Anderbrügge	http://dbpedia.org/resource/Wolfgang_Schüler

Figure 19 - The result of Q5 (For the convenience of viewing, I Screenshot and put it together).

7 Mix local and endpoint queries within Jena

In this section, by modifying the java code, I mixed local and endpoint queries with Jena.

The development environment is eclipse 4.5.1.

The local model queries, endpoint queries and results are as follow:

Q1. Show the populations of the cities where my friends live.

```
public static String FIND_MY_FRIENDS_LOCATION =
        " SELECT DISTINCT ?location   "+
        " WHERE "+
        "{ ex:LZF foaf:knows ?o . "+
        " ?o vCard:locality ?location . }";
```

Figure 20 - The local queries in Q1.

```
String sparqlQueryString1 =
"PREFIX dbo: <http://dbpedia.org/ontology/> " +
"PREFIX dbp: <http://dbpedia.org/property/>" +
"PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>" +
"PREFIX dbr: <http://dbpedia.org/page/>"+
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
"PREFIX foaf:  <http://xmlns.com/foaf/0.1/>"+
"   SELECT ?popu " +
"   WHERE { "+
"<"+value + ">  "+" dbo:populationTotal ?popu . "+
" }";
```

Figure 21 - The endpoint queries in Q1.

```
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileManager)
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
-----------------------------------------------------------------------
| popu                                                                |
=======================================================================
| "8705600"^^<http://www.w3.org/2001/XMLSchema#nonNegativeInteger> |
-----------------------------------------------------------------------
```

Figure 22 - The result of Q1 (My two friends live in the same city).

Q2. Show the list of cities where you friends live which are smaller (in area) than Ottawa.

```
public static String FIND_MY_FRIENDS_LOCATION =
        " SELECT DISTINCT ?location   "+
        " WHERE "+
        "{ ex:LZF foaf:knows ?o . "+
        " ?o vCard:locality ?location . }";
```

Figure 23 - The local queries in Q2.

```
String sparqlQueryString1 =
"PREFIX dbo: <http://dbpedia.org/ontology/> " +
"PREFIX dbp: <http://dbpedia.org/property/>" +
"PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>" +
"PREFIX dbr: <http://dbpedia.org/resource/>"+
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
"PREFIX foaf:  <http://xmlns.com/foaf/0.1/>"+
"   SELECT ?loca ?area " +
"   WHERE { "+
"<"+value + ">  "+" dbo:areaMetro ?area . "+
"<"+value + ">  "+" dbp:location ?loca . "+
" dbr:Ottawa dbo:areaMetro ?Ottawaarea .  "+
"FILTER (?area<?Ottawaarea) . "+
" }";
```

Figure 24 - The endpoint queries in Q2.

```
<terminated> Jenapro [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (2019年1月31日 上午11:53:11)
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
-----------------------------------------------------------------------------------------------
| loca                                                                    | area        | Ottawaarea |
===============================================================================================
| "Xi'an"^^<http://www.w3.org/1999/02/22-rdf-syntax-ns#langString> | 3.86625e+09 | 5.716e+09  |
-----------------------------------------------------------------------------------------------
```

Figure 25 - The result of Q2 .

Q3:Show the friends who work in the same city where they live.

```
SELECT DISTINCT ?city1
 WHERE
{ ex:LZF foaf:knows ?o .
ex:LZF foaf:knows ?o1 .
?o ex:studies ?course1.
?o1 ex:studies ?course2.
?course1 ex:offeredAt ?University1.
?course2 ex:offeredAt ?University2.
SERVICE <http://sparql.org/sparql>
{?University1 dbo:city ?city1.
?University2 dbo:city ?city2.
FILTER (?city1 = ?city2} .
}
}
```

```
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
city1  http://dbpedia.org/resource/Xi'an
```

Figure 26 - The result of Q3 .

Q4:Show the nickname of my friends' university.

```
public static String FIND_MY_FRIENDS_LOCATION =
        " SELECT DISTINCT ?University   "+
        " WHERE "+
        "{ ex:LZF foaf:knows ?o . "+
        " ?o ex:studies ?course . "+
        " ?course ex:offeredAt ?University}";
```

Figure 27 - The local queries in Q4.

```
String sparqlQueryString1 =
"PREFIX dbo: <http://dbpedia.org/ontology/> " +
"PREFIX dbp: <http://dbpedia.org/property/>" +
"PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>" +
"PREFIX dbr: <http://dbpedia.org/resource/>"+
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
"PREFIX foaf:  <http://xmlns.com/foaf/0.1/>"+
"   SELECT ?nickname" +
"   WHERE { "+
"<"+value + ">  "+" foaf:nick ?nickname . "+
" }";
```

Figure 28 - The endpoint queries in Q4.

```
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
-----------
| nickname |
============
------------
------------------
| nickname       |
==================
| "NWPU, NPU"@en |
------------------
```

Figure 29 - The result of Q4 .

Q5:Show the English name of provinces where my friends live in.

```
public static String FIND_MY_FRIENDS_LOCATION =
        " SELECT DISTINCT ?location   "+
        " WHERE "+
        "{ ex:LZF foaf:knows ?o . "+
        " ?o vCard:locality ?location . }";
```

Figure 30 - The local queries in Q5.

```
String sparqlQueryString1 =
"PREFIX dbo: <http://dbpedia.org/ontology/> " +
"PREFIX dbp: <http://dbpedia.org/property/>" +
"PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>" +
"PREFIX dbr: <http://dbpedia.org/page/>"+
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
"PREFIX foaf:  <http://xmlns.com/foaf/0.1/>"+
"   SELECT ?pn " +
"   WHERE { "+
"?Province dbp:capital "+ "<"+value + ">  ."+
"?Province dbp:englishname ?pn  ."
+" }";
```

Figure 31 - The endpoint queries in Q5.

```
log4j:WARN No appenders could be found for logger (org.apache.jena.util.FileManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
-------------------------------------------------------------------------------
| pn                                                                          |
===============================================================================
| "Shaanxi Province"^^<http://www.w3.org/1999/02/22-rdf-syntax-ns#langString> |
-------------------------------------------------------------------------------
```

Figure 32 - The result of Q5.

The java code of combining the part of the query from the model, and the part from DBpedia (Using Q1 as the example):

```
public class Jenapro {
        public static void main(String args[]) {
                String inputFileName= "F:\\Canada\\SemanticWeb\\homework1\\LZF.rdf";
                Model model = ModelFactory.createDefaultModel();
                InputStream in =FileManager.get().open(inputFileName);
                if(in==null) {
```

```
                    throw new IllegalArgumentException("File:"+inputFileName+"not found");
                }
                model.read(in,null);
                performQuery(FIND_MY_FRIENDS_LOCATION, model);
        }
        public static String PREFIX_STRING=
                        "PREFIX vCard: <http://www.w3.org/2001/vcard-rdf/3.0#>"+
                        "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>"+
                        "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
                        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"+
                        "PREFIX foaf:  <http://xmlns.com/foaf/0.1/>"+
                        "PREFIX dc: <http://purl.org/dc/elements/1.1/>"+
                        "PREFIX dbr: <http://dbpedia.org/page/>"+
                        "PREFIX ex: <http://csi5180-example.org/>"+
                        "PREFIX dbo: <http://dbpedia.org/ontology/>"+
                        "PREFIX dbp: <http://dbpedia.org/property/>";
        public static String FIND_MY_FRIENDS_LOCATION =
                        " SELECT DISTINCT ?location   "+
                        " WHERE "+
                        "{ ex:LZF foaf:knows ?o . "+
                        " ?o vCard:locality ?location . }";
        public static void performQuery(String queryString, Model model) {
        StringBuffer sBuffer = new StringBuffer();
        Query query = QueryFactory.create(PREFIX_STRING +queryString);
        QueryExecution qexec = QueryExecutionFactory.create(query,model);
        try {
                ResultSet results = qexec.execSelect();
                while(results.hasNext()) {
                        QuerySolution soln = results.nextSolution();
                        Iterator<String> names = soln.varNames();
                        while(names.hasNext()) {
                                String varName =names.next();
                                String value =soln.get(varName).toString();
                                String sparqlQueryString1 =
                                "PREFIX dbo: <http://dbpedia.org/ontology/> " +
                    "PREFIX dbp: <http://dbpedia.org/property/>" +
                    "PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>" +
                    "PREFIX dbr: <http://dbpedia.org/resource/>"+
                    "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
                    "PREFIX foaf:  <http://xmlns.com/foaf/0.1/>"+
                    "   SELECT ?popu" +
                    "   WHERE { "+
                    "<"+value + ">  "+" dbo:populationTotal ?popu . "+
                    " }";
                Query query1 = QueryFactory.create(sparqlQueryString1);
                QueryExecution qexec1 =
QueryExecutionFactory.sparqlService("http://dbpedia.org/sparql", query1);
                ResultSet results1 = qexec1.execSelect();
```

```
                ResultSetFormatter.out(System.out, results1, query1);
                sBuffer.append(results1);
                qexec1.close() ;
                        }
                }
        }
        finally {
                qexec.close();
        }
        }
}
```

8 References

[1] CSI 5180 Topics in AI - Ontologies and Semantic Web. Retrieved from https://uottawa.brightspace.com/d2l/le/content/102243/viewContent/2140883/View/

[2] An Introduction to RDF and the Jena RDF API. Retrieved from http://jena.apache.org/tutorials/rdf_api.html/

[3] RDF Validator and Converter. Retrieved from http://rdfvalidator.mybluemix.net/

[4] Online visualization program. Retrieved from http://visgraph3.org/

[5] Virtuoso SPARQL Query Editor. Retrieved from http://dbpedia.org/sparql/