Andrew Plum

Professor Ma

CS 489

11/5/2024

SPARQL Query Report

1. “State the topic that you select and the purpose of your query.”
   1. My topic is figureheads of the executive branch in American history. My query returns a list of US presidents and vice presidents, the office they held (either president or vice president), the political parties they were affiliated with, and the start and end of their term in the specific office they held.
2. “Describe the process in the design of your query: e.g. How you find relevant classes and properties in the SPARQL endpoint of DBpedia? How you select the classes and properties that can be used to compose your query? How you design different parts of your query?”
   1. Because of the goal I wanted to achieve with my query, I knew I wanted to return people in my query, so I started by looking at the web page “<https://dbpedia.org/ontology/Person>” for relevant information. On the person page, I found the two relevant classes “dbo:president” or “dbo:vicePresident”; I then returned a query result of subjects whose rdf:type were these objects. I was then able to look at some specific presidents and vice presidents because these were the subjects which returned however the list seemed to be too inclusive as it contained subjects who were not leaders of the US, so I tried to narrow it down. By looking at the dbpedia pages of the subjects whose rdf:type “dbo:president” or “dbo:vicePresident” (example: <https://dbpedia.org/page/Millard_Fillmore>), in the rdf:type section I found better classes “yago:WikicatPresidentsOfTheUnitedStates” and “yago:WikicatVicePresidentsOfTheUnitedStates” where these subjects were a better comprehensive list of all the US presidents and vice presidents. From there, to finish building the query, it was just continuing the iterative process I was already doing of looking for and testing the relevant pieces of information I could find on dbpedia to solve the subproblems I was trying to do to complete my overall goal of the query.
3. “Explain the meaning of the variables you use in the SPARQL code. Describe how you organize the sequence of your SPARQL code to achieve the purpose of the query.”

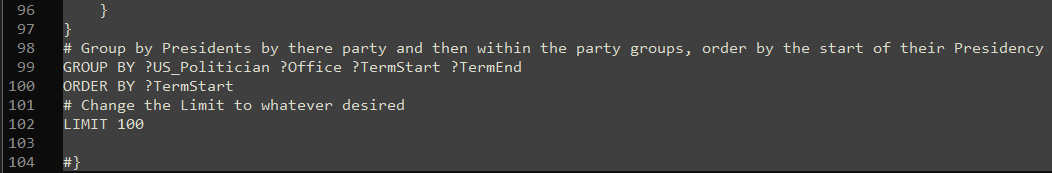
\*Note: Might need to zoom in on some of the pictures\*

* 1. A computer screen with white text

     Description automatically generated
     1. This is the beginning of the query where prefixes which are referenced are established and the final column variables of the results are listed in the SELECT clause; the variable columns that will show up in the final results are ?US\_Politician which is the name of the US politician of the row, ?TotalPartyAffiliations which is the number of party affiliations the US politician in the row had over their lifetime, ?Office which is the specific office this politician held (should only be president or vice president in this query; if they held more than one there should be more than one rows with the same US\_politician value), ?PartyAffiliations which is a comma separated list of the specific parties the politician was a part of (used GROUP\_CONCAT to create the list), ?TermStart which is the start date of the term the politician had for holding the office in the specific row, and ?TermEnd which is the end date of the term the politician had for holding the office in the specific row; all of these variables are selected from the subquery in the next code section.
  2. A screen shot of a computer

     Description automatically generated
     1. This subquery selects many of the same variables mentioned in the first code section in its SELECT clause, although several of the variables are named differently (?US\_politician is a renamed variable which is the variable ?instance in this subquery) and ?party is not yet the concatenated list ?Parties; in the WHERE clause, using rdf: and dbo: ?instance is narrowed to being a person, a politician, and more specifically has rdf:type of “yago:WikicatPresidentsOfTheUnitedStates” or (done with the UNION clause) “yago:WikicatVicePresidentsOfTheUnitedStates” which when all done together, ?instance should be a list of US presidents and vice presidents, however from my querying of only this, ?instance items are not only presidents and vice presidents but also items like <http://dbpedia.org/resource/Timeline_of_Presidents_of_the_United_States> which will need to be filtered out later; at the end of the code section is where I create a variable ?party which is the object in the dbo:party relation with the subject ?instance.
  3. A screenshot of a computer program

     Description automatically generated
     1. This code section shows a big UNION clause which entails two different, although somewhat similar, blocks of next statements based on whether the ?instance value is a US politician was a US president or a US vice president; at the top of both of each subsections in the UNION clause in a similar way to how I created ?party previously, I get the values of ?instance’s dbp:office and rdfs:comment and store each respectively in ?office and ?comment; I then filter out the rows so that I only have rows with values in these variables that are English and have the presence of strings which ensure the ?instance was actually a US president or vice president; I then use several optional clauses to get the specific start date (stored in variable ?TermStart) and end date (stored in variable ?TermEnd) of the term of office that the ?instance value of the row has, this subsection of statements are set up in this way with several optional clauses in the case of a date values not existing to be stored in either ?TermStart or ?TermEnd or both, a relevant example of this is shown on this page <https://dbpedia.org/page/Joe_Biden__Tenure__1> as Joe Biden only has a recorded start date to his term of office but not yet an end date because he is a current president – I want him to be listed in the results with the relevant start date value of his term shown and the optional clauses set up this way does that; this concludes the two subsections of the UNION clause.
  4. A screen shot of a computer code

     Description automatically generated
     1. Here more filtering is done where rows kept have values in the variable ?office only with the string “of the United States” at the end of their string value; values in ?instance do not contain the substrings “administration”, “reforms”, “policy”, or “campaign” at the end of the string; and the start of the string values in ?instance contain only the string “http://dbpedia.org/resource/”.
  5. 
     1. This is near the end of the query where I use a GROUP BY clause to group by the variables ?US\_Politician ?Office ?TermStart ?TermEnd; I use an ORDER BY clause to order the rows by their start date in ascending order; I use the LIMIT clause to limit the results to just the first 100 rows (this should show all of the US presidents or vice presidents but it can easily be changed to show only the first x amount of rows in the results).

SPARQL Code:

PREFIX dbo: <http://dbpedia.org/ontology/>

PREFIX dbr: <http://dbpedia.org/resource/>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

# For totaling the number of rows - need to comment out inner SELECT and uncomment the bottom }

#SELECT (SUM(?US\_President\_Count) AS ?US\_President\_Total)

#{

#SELECT (COUNT(DISTINCT \*) AS ?US\_President\_Count)

# Main query

SELECT ?US\_Politician

(COUNT(\*) AS ?TotalPartyAffiliations)

?Office

(GROUP\_CONCAT(DISTINCT ?Party; separator = ", ") AS ?PartyAffiliations)

?TermStart

?TermEnd

WHERE

{

SELECT DISTINCT ?instance AS ?US\_Politician, COUNT(DISTINCT ?instance) AS ?InstancesInResult, ?office AS ?Office, ?party AS ?Party, ?TermStart, ?TermEnd

WHERE

{

# Narrow entries to US presidents

?instance rdf:type dbo:Person ;

rdf:type dbo:Politician .

# Instance is the union of US Presidents and Vice Presidents

{

?instance rdf:type yago:WikicatPresidentsOfTheUnitedStates . # Common type of presidents in this list dbr:List\_of\_presidents\_of\_the\_United\_States but returns to many items

}

UNION

{

?instance rdf:type yago:WikicatVicePresidentsOfTheUnitedStates . # Common type of vice presidents in this list dbr:List\_of\_vice\_presidents\_of\_the\_United\_States but returns to many items

}

# Selected variable

?instance dbo:party ?party .

# Beginning of large union structure for getting the correct US presidents and vice presidents

# Without it, some items that are the yago type but not US presidents and vice presidents will be in the results

{

?instance dbp:office ?office ;

rdfs:comment ?comment .

# Filter office for US President

FILTER (LANG(?office) = "en") # only English

FILTER (REGEX(STR(?office), "^President of the United States$", "i")) # "i" is a flag which makes the search case insensitive

# Filter comment for US President

FILTER (LANG(?comment) = "en") # only English

FILTER (REGEX(STR(?comment), "President of the United States", "i")) # "i" is a flag which makes the search case insensitive

# Selected variable

OPTIONAL # Optional to allow for the inclusion of presidents in the results even if they do not have a recorded start date associated with them

{

?instance dbo:termPeriod ?tenurePeriod .

?tenurePeriod dbo:office ?tenureOffice .

FILTER (REGEX(STR(?tenureOffice), "^President of the United States$", "i")) # "i" is a flag which makes the search case insensitive

}

OPTIONAL { ?tenurePeriod dbo:start ?TermStart . }

OPTIONAL { ?tenurePeriod dbo:end ?TermEnd .}

}

UNION

{

?instance dbp:office ?office ;

rdfs:comment ?comment .

# Filter office for US President

FILTER (LANG(?office) = "en") # only English

FILTER (REGEX(STR(?office), "^Vice President of the United States$", "i")) # "i" is a flag which makes the search case insensitive

# Filter comment for US President

FILTER (LANG(?comment) = "en") # only English

FILTER (REGEX(STR(?comment), "Vice President of the United States", "i")) # "i" is a flag which makes the search case insensitive

# Selected variable

OPTIONAL # Optional to allow for the inclusion of presidents in the results even if they do not have a recorded start date associated with them

{

?instance dbo:termPeriod ?tenurePeriod .

?tenurePeriod dbo:office ?tenureOffice .

FILTER (REGEX(STR(?tenureOffice), "^Vice President of the United States$", "i")) # "i" is a flag which makes the search case insensitive

}

OPTIONAL { ?tenurePeriod dbo:start ?TermStart . }

OPTIONAL { ?tenurePeriod dbo:end ?TermEnd . }

}

# End of large union structure for getting the correct US presidents and vice presidents

# Without it, some items that are the yago type but not US presidents and vice presidents will be in the results

# Check to make sure end of string of ?instance is "of the United States"

FILTER (STRENDS(STR(?office), "of the United States"))

# Filter variable that is being SELECTED for keywords we don't want in query results

FILTER (!REGEX(STR(?instance), ".\*(administration|reforms|policy|campaign)$", "i")) # Results with these keywords anywhere in them are excluded

# Make sure all results returned have a dbpedia reasource in the instance column

FILTER (STRSTARTS(STR(?instance), "http://dbpedia.org/resource/"))

}

}

# Group by Presidents by there party and then within the party groups, order by the start of their Presidency

GROUP BY ?US\_Politician ?Office ?TermStart ?TermEnd

ORDER BY ?TermStart

# Change the Limit to whatever desired

LIMIT 100

#}