Semantic Music Genre Data from DBpedia: Geo-plotting in Google Maps

Kushal Kokje, Vinoth Aryan Nagabooshanam, Udit Patel
Department of Data Science
School of Library and Information Science
Indiana University
Bloomington, USA
{kkokje,vinaryan,udipatel}@umail.iu.edu

Abstract

Web 3.0, the Linked Data Web, the Web of Data or mainly referred as the Semantic Web represents the next major evolution in connecting information. It enables data to be linked from a source to any other source and to be understood by computers so that they can perform increasingly sophisticated tasks on our behalf. The Web becomes a platform for integrating data and services. This paper discusses the use of linked data in DBpedia for discovering Artists in mainly four genres of music trance, techno, metal and dubstep. This paper demo the power of semantic web and linked data.

Keywords: linked data, DBpedia, SPARQL, Geo-plotting in Google maps

1. Introduction:

With the decade-round endeavor from the Semantic Web believers, researchers and practitioners, Semantic Web has made remarkable progress recently. It has raised significant attentions from US, UK governments, and European Commission who are willing to deploy Semantic Web technologies to enhance the transparency of egovernment. The Linked Open Data (LOD) initiative is on their way to convert current document web into data web and to further enable various data and service mashups. The fast adoption of Semantic Web technologies in medical and life science has created impressive showcases to the world. All these efforts march a crucial step to enable the takeoff and the success of the Semantic Web.

DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data. This work will make it easier for the huge amount of information in Wikipedia to be used in some new interesting ways.
There is a public SPARQL endpoint over the DBpedia data set at http://dbpedia.org/sparql. The endpoint is provided using OpenLink Virtuoso as both the back-end database engine and the HTTP/SPARQL server.

We studied the OWL classes required for extracting the data about Music Artist's Name, Birthplace and Date of Birth and created a SPARQL query which is executed against the DBpedia SPARQL endpoint.
The results set which is a set of triples is exported into a CSV file and later plotted into a world map using
Google Maps API. Users can scroll over a geographic area to look for artists and find artists by name, their webpage link and their date of birth.

2. Related work

People have used DBpedia data previously to create state-of-the-art visualization across several domains. The data can also be used to prepare models in medial domain for educational purposes. The projects and use cases of DBpedia can be viewed at http://wiki.dbpedia.org/projects

3. Extracting Data from DBpedia

Knowledge bases are playing an increasingly important role in enhancing the intelligence of Web and enterprise search and in supporting information integration. Today, most knowledge bases cover only specific domains, are created by relatively small groups of knowledge engineers, and are very cost intensive to keep up-to-date as domains change. At the same time, Wikipedia has grown into one of the central knowledge sources of mankind, maintained by thousands of contributors. The DBpedia project leverages this gigantic source of knowledge by extracting structured information from Wikipedia and by making this information accessible on the Web.
The English version of the DBpedia knowledge base describes 4.58 million things, out of which 4.22 million are classified in a consistent ontology, including 1,445,000 persons, 735,000 places (including 478,000 populated places), 411,000 creative works (including 123,000 music albums, 87,000 films and 19,000 video games), 241,000 organizations (including 58,000 companies and 49,000 educational institutions), 251,000 species and 6,000 diseases.
We'll be using DBpedia SPARQL endpoint called snorql to retrieve music artist's data only for Genre Trane, Music, Dubstep and metal.

For almost every Wikipedia page there exists a similar DBpedia page with the class information. For example, wiki page Wikipedia.org/page/Trance can be viewed in Dbpedia as Dbpedia.org/page/Trance. The page contains basic information semantically arranged on the wiki page. Below is a snapshot of Dbpedia page for "Trance".

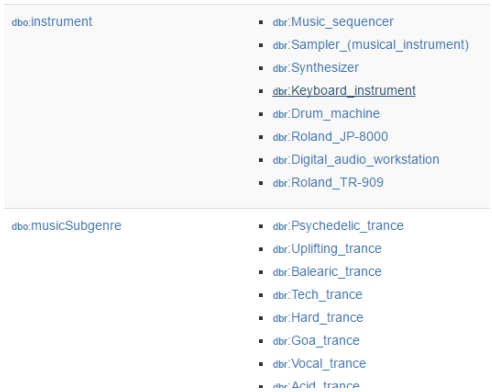

Figure 1: DBpedia OWL classes

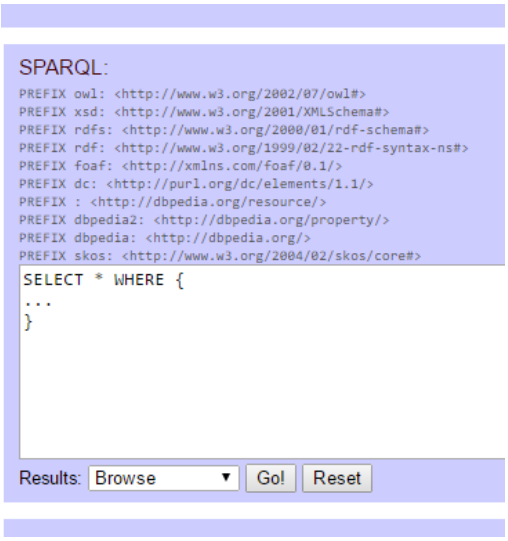

Figure 2: DBpedia SPARQL endpoint SNORQL

```
PREFIX dbpediao: <http://dbpedia.org/ontology/>

SELECT STRAFTER(STR(?name),"http://dbpedia.org/resource/") as ?Name
       STR(?name) as ?DBpedia_link
       COALESCE(STRAFTER(STR(?birthplace),"http://dbpedia.org/resource/"),"Bloomington") as ?birthplace
       ?bdate
WHERE {
        SELECT ?name as ?name
               max(?birthplace) as ?birthplace
               max(STR(?bdate)) as ?bdate
        WHERE  {

               {
               ?name a dbpediao:MusicalArtist.
               ?name foaf:givenName ?givenname.
               OPTIONAL{?name dbo:birthDate ?bdate.}
               OPTIONAL{?name dbp:birthPlace ?birthplace.
                        ?birthplace a dbo:Place }.
               OPTIONAL{?name dbo:hometown ?birthplace .
                        ?birthplace a dbo:Place}.
               OPTIONAL{?name dbo:birthPlace ?birthplace .
                        ?birthplace a dbo:Location}.
               ?name dbpediao:genre ?genre.
               ?genre dbp:name ?t_name.
               FILTER regex(?t_name,"^trance","i").
               }
```

Figure 3: SPARQL query for Genre "Trance"

As shown in Figure [2] we create the graph pattern by looking resources which are instance of owl DBpedia class dbpediao:MusicalArtist. We link the name of the artist with Genre class to FILTER OUT Musical artists having Genre as "Trance". Then we use dbo:birthdate class to get the artists birthdate and a dbo:birthplace to extract the birthplace. We also link the data for dbo:hometown where dbo:birthplace value is not available.
Executing the query on DBpedia Sparql endpoint with FILTER condition for genre as "Trance" we get following results (only samples shown here, actual result contains about 200-250 triplets).

SPARQL results:

Name	DBpedia_link	birthplace	bdate
"Mark_Reeder"	"http://dbpedia.org/resource/Mark_Reeder"	"Bloomington"	"1958-01-05"
"ATB"	"http://dbpedia.org/resource/ATB"	"Freiberg"	"1973-02-26"
"Alston_Koch"	"http://dbpedia.org/resource/Alston_Koch"	"Sri_Lanka"	-
"Randy_Katana"	"http://dbpedia.org/resource/Randy_Katana"	"Saint_Martin"	"1965-03-14"
"Pulstate"	"http://dbpedia.org/resource/Pulstate"	"Ireland"	"1992-05-10"
"William_Orbit"	"http://dbpedia.org/resource/William_Orbit"	"Shoreditch"	"1956-12-15"

Figure 4: Query results for Genre "Trance"

We also made sure that no duplicates are present in the data. Duplicates in the data would further lead to extra pins on the Google map plotting confusing the viewers.
We created four different sparql queries each one for Trance,metal,techno and dubstep and executed the queries against DBpedia sparql endpoint.
We exported the data and performed data quality checks to avoid errors while Geo-plotting of the data.

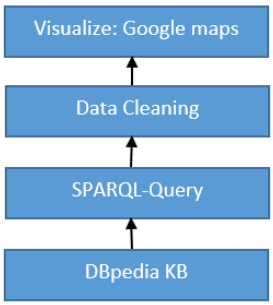

Figure 5: Data Flow diagram

4. Geo-plotting in Google Maps

The Google Map Chart displays a map using the Google Maps API. Data values are displayed as markers on the map. Data values can be coordinates (lat-long pairs) or addresses. The map will be scaled so that it includes all the identified points. We can identify the places to put markers by name on the map. When the user selects one of the markers, a tooltip with the name and population of the country is displayed, because we used the showInfoWindow option. Also, when the user hovers over one of the markers for a short while, a 'title' tip will show up with the same info, because we used the showTooltip option.

```
var data = google.visualization.arrayToDataTable([
  ['Lat', 'Long', 'Name'],
  [37.4232, -122.0853, 'Work'],
  [37.4289, -122.1697, 'University'],
  [37.6153, -122.3900, 'Airport'],
  [37.4422, -122.1731, 'Shopping']
]);
```

Figure 6: Input for ArrayToDatatable got G-maps

The major part of the visualization which is a arrayToDataTable which maps locations on the map is shown in the above figure. Latitude-Longitude pair or Location name can be used by the API to plot data on map.

5. Results of the Geo-plotting

Figure 7: “Trance” artists over European area

Figure 8: “Trance” artists over USA area

Figure 9: “Trance” artist sample info box

6. Publishing the results

We have published the results of our findings on Indiana University Mercury server. The links to the published webpages are as follows:

Trance: http://pages.iu.edu/~kkokje/Trance-map.html

Techno: http://pages.iu.edu/~kkokje/techno-map.html

Dubstep: http://pages.iu.edu/~kkokje/dubstep-map.html

Metal: http://pages.iu.edu/~kkokje/metal-map.html

It usually takes like 2-3 minutes to load the data into the map. Users can scroll over the Google map for a specific location and click the location pin to get more details about the artist. Additional details about the artists can be found by visiting the DBpedia link mentioned in the INFO box.

7. Conclusion

This paper expresses the power of semantic web and how linked data in the DBpedia can be used to explore data from Wikipedia. We extracted the data using the sparql endpoint and then visualized using Google Maps API. The project can be further improved by providing a link in the info box of the location pin so that users can directly link to corresponding wiki page. The model can also be deployed over web in way that users can query geographic related data and visualize the same without extra modifications.

REFREENCES

[1] DBpedia.org

[2] DBpedia.org/snorql

[3] GoogleMaps API

[4] http://www.cambridgesemantics.com/semantic-university/introduction-semantic-web