SPARQL

Martín Alejandro Castro Álvarez

martincastro.10.5@gmail.com

Universidad Internacional de Valencia (VIU)

Calle Pintor sorolla, 21 46002, Valencia (España)

Abril 2024

Abstract. Este estudio demuestra la aplicación de SPARQL para consultar y extraer datos estructurados de DBpedia sobre la ciudad de Valencia, España. Utilizando una serie de consultas SPARQ, se obtienen datos multidimensionales que incluyen aspectos geográficos, demográficos y culturales. Esta investigación no solo refleja la capacidad de SPARQL como herramienta poderosa para la Semántica Web sino que también destaca su eficacia en la integración y visualización de datos complejos.

Keywords: SparkQL. DBPedia. Python. Canva.

1. Introducción

1.1. Objetivo

Realiza y explica una consulta SPARQL para obtener información sobre la ciudad de Valencia en España, utilizando DBpedia. Selecciona propiedades relevantes como nombre, población total, área total en kilómetros cuadrados, coordenadas geográficas (latitud y longitud), comunidad autónoma a la que pertenece, y otras características que consideres interesantes. Utiliza la siguiente URI: http://dbpedia.org/resource/Valencia como punto de partida para la consulta. Además, puedes incluir detalles sobre historia y fundación, fiestas y tradiciones, etc. Asegúrate de ordenar la información de manera clara y presentar los resultados de manera legible.

1.2. Solución Propuesta

La solución propuesta implica el uso de SPARQL para realizar consultas complejas a la base de datos semántica de DBpedia con el fin de recopilar información detallada sobre Valencia. A través de un script de Python, se ejecutan múltiples consultas que permiten la búsqueda de datos específicos como el nombre de la ciudad, la población total, el área total, las coordenadas geográficas, etc.

2. Antecedentes Teóricos

2.1. SparkQL

Spark SQL es un módulo de Apache Spark diseñado para el manejo de datos estructurados, integrándose de manera simple con los programas Spark para permitir consultas SQL junto con la API DataFrame, que es accesible desde Java, Scala, Python y R [1].

2.2. DBPedia

DBpedia es una plataforma de datos abiertos, que ofrece una interfaz sobre los datos de Wikipedia (y otros recursos colaborativos como Wikidata) que sirve como herramienta esencial para investigadores, académicos y profesionales, porque permite el acceso y la manipulación de datos estructurados de forma semántica.

3. Metodología

3.1. Python

Para ejecutar las consultas y recolectar los datos relacionados a Valencia, se ha utilizado el programa de Python (Fig. 1), que consiste en la ejecución de cuatro consultas SPARQL secuenciales:

Primero, se realiza una consulta principal para obtener un conjunto de datos básicos. Segundo, se ejecuta una segunda consulta enfocada en recopilar referencias visuales, que retorna varios resultados. Tercero, se hace una consulta por las locaciones relacionadas, y finalmente una última consulta obtiene datos relacionados con esas locaciones. En esa última consulta, se utilizan los resultados de la consulta anterior.

El programa se utiliza con el comando (1).

$$poetry\ run\ python3\ spark.py \tag{1}$$

3.2 Consulta de Detalles

La consulta (1) está diseñada para obtener información general y esencial sobre Valencia. Utiliza varios predicados para extraer datos como el resumen de la ciudad (dbo:abstract), la página web oficial (dbp:website), el país al que pertenece (dbo:country), código postal (dbo:postalCode), el área total y urbana en kilómetros cuadrados (dbp:areaTotalKm y dbp:areaUrbanKm), la población total (dbo:populationTotal), y las coordenadas geográficas (geo:lat y geo:long).

```
PREFIX dbo: <http://dbpedia.org/ontology/>                    (2)
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX geo:
<http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX dbr: <http://dbpedia.org/resource/>

SELECT
  ?abstract
  ?website
  ?country
  ?zip
  ?areaTotal
  ?areaUrban
  ?population
  ?lat
  ?long
WHERE {
  dbr:Valencia
  dbo:abstract            ?abstract       ;
  dbp:website             ?website        ;
  dbo:country             ?country        ;
  dbo:postalCode          ?zip            ;
  dbp:areaTotalKm         ?areaTotal      ;
  dbp:areaUrbanKm         ?areaUrban      ;
  dbo:populationTotal     ?population     ;
  geo:lat                 ?lat            ;
  geo:long                ?long           .
  FILTER (lang(?abstract) = 'es')
}
```

3.3 Consulta de Imágenes

La consulta (3) se centra en la recopilación de imágenes representativas de Valencia. Busca enlaces directos a imágenes utilizando el predicado foaf:depiction, que se relaciona con representaciones visuales de la ciudad.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbr: <http://dbpedia.org/resource/>

SELECT ?ref
WHERE {
  dbr:Valencia foaf:depiction ?ref
}
```
(3)

3.4 Consulta de Locaciones Importantes

La consulta (4) explora entidades y eventos significativos ubicados en Valencia, utilizando el predicado dbo:location para identificar aquellos elementos que están geográficamente asociados con la ciudad.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>

SELECT ?event WHERE {
  ?event dbo:location dbr:Valencia .
}
```
(4)

3.5 Consulta de Detalles de Locaciones Importantes

La consulta (5) toma los resultados de la consulta (4) y busca la información de cada locación o evento identificado. Busca detalles adicionales como etiquetas descriptivas (rdfs:label), resúmenes en español (dbo:abstract), páginas web oficiales (foaf:homepage), fechas de inicio (dbo:startDate), y coordenadas específicas (dbo:lat y dbo:long). Para evitar obtener múltiples resultados, se agrupan utilizando GROUP BY, de la misma manera que se puede hacer en cualquier base de datos SQL. Además, el hecho de que no todas las locaciones y eventos tienen los mismos atributos, es necesario utilizar la cláusula OPTIONAL.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX res: <http://dbpedia.org/resource/Valencia>
PREFIX geo:
<http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```
(5)

```
SELECT
  (SAMPLE(?label) AS ?label)
  ?entity
  (SAMPLE(?abstract) AS ?abstract)
  (SAMPLE(?website) AS ?website)
  (SAMPLE(?date) AS ?date)
  (SAMPLE(?lat) AS ?lat)
  (SAMPLE(?long) AS ?log)
WHERE {
  VALUES ?entity { dbr:Miguelete_Tower
dbr:Valencia_History_Museum dbr:Aqua_Multiespacio }
  OPTIONAL {?entity dbo:abstract ?abstract . FILTER
(lang(?abstract) = 'es')}
  OPTIONAL {?entity rdfs:label ?label}
  OPTIONAL {?entity foaf:homepage ?website}
  OPTIONAL {?entity dbo:startDate ?date}
  OPTIONAL {?entity geo:lat ?lat}
  OPTIONAL {?entity geo:long ?long}
}
GROUP BY ?entity
```

4. Resultados

4.1. Salida

La salida del programa de Python (Fig. 1) da como resultado un archivo (Fig. 2) que contiene la siguiente información:

En primer lugar, se obtiene la descripción de Valencia, su web, la referencia en DBPedia del país en el que se encuentra, el rango de códigos postales, el área total, el área urbana total, la población total, y sus coordenadas geográficas.

En segundo lugar, se obtiene una lista de 9 URLs de imágenes representativas de Valencia.

En tercer lugar, se obtiene información acerca de 3 ubicaciones importantes de Valencia: “Aqua Multiespacio”, “Micalet”, y “Museu d’história de Valéncia”.

4.2 Infographic

Como es habitual en el ámbito profesional, los resultados del análisis no tienen ningún valor, a menos que logren convencer a la audiencia sobre la utilidad de los datos presentados.

Los resultados de este análisis han sido presentados en formato infographic utilizando la herramienta canva.com (Fig. 3).

5. Referencias

[1] Apache Spark. (s.f.). “Spark SQL”. https://spark.apache.org/sql/

[2] DBpedia Association. (s.f.). “Acerca de DBpedia”. https://www.dbpedia.org/

6. Anexos

```
import os
import json
import random
import urllib.parse
from SPARQLWrapper import SPARQLWrapper, JSON
PREFIXES: dict[str, str] = {
    "dbo": "<http://dbpedia.org/ontology/>",
    "dbp": "<http://dbpedia.org/property/>",
    "res": "<http://dbpedia.org/resource/Valencia>",
    "geo": "<http://www.w3.org/2003/01/geo/wgs84_pos#>",
    "foaf": "<http://xmlns.com/foaf/0.1/>",
    "dbr": "<http://dbpedia.org/resource/>",
    "rdfs": "<http://www.w3.org/2000/01/rdf-schema#>",
}
def dedent(s: str) -> str:
    return "\n".join([
        line.strip()
        for line in s.split("\n")
    ])
def send(query: str) -> list[dict]:
    sparql: SPARQLWrapper = SPARQLWrapper("http://dbpedia.org/sparql")
    sparql.setReturnFormat(JSON)
    for prefix in reversed(PREFIXES):
        query = f"PREFIX {prefix}: {PREFIXES[prefix]}\n{query}"
    query = dedent(query)
    sparql.setQuery(query)
    print(f"Query:\n{query}")
    response = sparql.query().convert()
    print(f"Response:\n{json.dumps(response, indent=2, sort_keys=True)}")
    return [
        result
        for result in response.get("results", {}).get("bindings", [])
    ]
def test():
    output = "valencia.txt"
    if os.path.isfile(output):
        os.remove(output)
    query = """
        SELECT
            ?abstract
            ?website
            ?country
            ?zip
            ?areaTotal
            ?areaUrban
            ?population
            ?lat
            ?long
```

```
    WHERE {
        dbr:Valencia
            dbo:abstract            ?abstract       ;
            dbp:website             ?website        ;
            dbo:country             ?country        ;
            dbo:postalCode          ?zip            ;
            dbp:areaTotalKm         ?areaTotal      ;
            dbp:areaUrbanKm         ?areaUrban      ;
            dbo:populationTotal     ?population     ;
            geo:lat                 ?lat            ;
            geo:long                ?long           .
        FILTER (lang(?abstract) = 'es')
    }
"""
details = send(query)
assert details
assert len(details) == 1
details = details[0]
query = """
    SELECT
        ?ref
    WHERE {
        dbr:Valencia foaf:depiction ?ref
    }
"""
references = send(query)
assert references
query = """
    SELECT ?event WHERE {
        ?event dbo:location dbr:Valencia .
    }
"""
locations = send(query)
assert locations
names = []
location_samples = 3
for location in random.sample(locations, location_samples):
    uri = location['event']['value']
    name = urllib.parse.quote(uri.split('resource/')[1], safe='')
    names.append(f"dbr:{name}")
query = f"""
    SELECT
        (SAMPLE(?label) AS ?label)
        ?entity
        (SAMPLE(?abstract) AS ?abstract)
        (SAMPLE(?website) AS ?website)
        (SAMPLE(?date) AS ?date)
        (SAMPLE(?lat) AS ?lat)
        (SAMPLE(?long) AS ?log)
    WHERE {{
        VALUES ?entity {{ {' '.join(names)} }}
```

```
        OPTIONAL {{?entity dbo:abstract ?abstract . FILTER (lang(?abstract) = 'es')}}
        OPTIONAL {{?entity rdfs:label ?label}}
        OPTIONAL {{?entity foaf:homepage ?website}}
        OPTIONAL {{?entity dbo:startDate ?date}}
        OPTIONAL {{?entity dbo:lat ?lat}}
        OPTIONAL {{?entity dbo:long ?long}}
      }}
      GROUP BY ?entity
    """
    location_details = send(query)
    assert location_details
    assert len(location_details) == location_samples, len(location_details)
    with open(output, "w") as f:
      f.write("DETAILS:\n")
      for key, value in details.items():
        f.write(f"{key}: {value['value']}"[:300] + "\n")
      f.write("\n\n")
      f.write("IMAGES:\n")
      for reference in random.sample(references, 9):
        f.write(f" - {reference['ref']['value']}\n")
      f.write("\n\n")
      f.write("LOCATIONS:\n")
      for location in location_details:
        f.write("\n")
        for key, value in location.items():
          f.write(f"{key}: {value['value']}"[:300] + "\n")
if __name__ == "__main__":
    test()
```

Fig. 1: Script de Python responsable de ejecutar las consultas con SparkQL

DETAILS:
abstract: Valencia (oficialmente en valenciano: València, AFI: [vaˈɬensia])es un municipio y una ciudad de España, capital de la provincia homónima y de la Comunidad Valenciana. Con una población de 801 545 habitantes (2020),que sube a 1 581 057 habitantes (2020) si se incluye su espacio urbano,
website: http://www.valencia.es
country: http://dbpedia.org/resource/Spain
zip: 46000-46080
areaTotal: 134.65
areaUrban: 628.8099999999999
population: 789744
lat: 39.46666717529297
long: -0.375
IMAGES:
- http://commons.wikimedia.org/wiki/Special:FilePath/Escut_de_València.svg
- http://commons.wikimedia.org/wiki/Special:FilePath/Valencia,_Spain.jpg
- http://commons.wikimedia.org/wiki/Special:FilePath/Crema_falla_2015.jpg
-
http://commons.wikimedia.org/wiki/Special:FilePath/Collage_de_la_ciudad_de_Valencia,_capital_de_la_Comunidad_Valenciana,_España.png
- http://commons.wikimedia.org/wiki/Special:FilePath/El_Museu_de_les_Ciències_Príncipe_Felipe_–_Bilim_ve_Uzay_Müzesi.jpg
- http://commons.wikimedia.org/wiki/Special:FilePath/Sentiment_Valencianista.jpg
- http://commons.wikimedia.org/wiki/Special:FilePath/Convento_de_Santo_Domingo,_Valencia,_España,_2014-06-29,_DD_13.jpg
- http://commons.wikimedia.org/wiki/Special:FilePath/Baldomer_Gili_Roig._El_Palmar_(La_Albufera_de_València),_c._1915_(2).jpg
- http://commons.wikimedia.org/wiki/Special:FilePath/Plat_àrab_València.jpg
LOCATIONS:
label: Aqua Multiespacio
entity: http://dbpedia.org/resource/Aqua_Multiespacio
abstract: Aqua Multiespacio es un edificio de Valencia en España. Tiene una altura de 95 metros convirtiéndolo en el tercero más alto de la ciudad tras la Torre Hilton y la Torre de Francia, el complejo consta de dos torres, la más alta con planta ovalada y la más baja con planta rectangular, Su con
label: Micalet
entity: http://dbpedia.org/resource/Miguelete_Tower
abstract: La torre del Miguelete (en valenciano, Torre del Micalet) es la torre campanario de la catedral de Valencia, España. La construcción de la torre se inicia en 1381 y finaliza en 1429. Por su complejidad y largos años de construcción, fue dirigida sucesivamente por varios maestros de obra; s
website: https://catedraldevalencia.es/el-miguelete/
label: Museu d'història de València
entity: http://dbpedia.org/resource/Valencia_History_Museum
abstract: El Museo de historia de Valencia (en valenciano y oficialmente, Museu d'història de València), conocido como MhV, inaugurado el 7 de mayo de 2003, es un museo dedicado al desarrollo de la historia de la ciudad de Valencia (España). El objetivo de su exposición permanente es presentar los p
website: http://www.valencia.es/mhv

Fig. 2: Resultado generado por el script de Python que genera las consultas SparkQL

Fig. 3: Presentación amigable de los resultados obtenidos de DBPedia