Práctica 2: Linked Data, Big Data y Web Social

Sistemas de Información Distribuidos

Pedro Allué Tamargo (758267) Juan José Tambo Tambo (755742)

10 de marzo de 2021

Índice

1. Esfuerzos

- **Pedro**: Realización del ejercicio 1, ejercicio 2 y redacción de la memoria.
  - **Tiempo invertido:** 5:30 horas.
- **Juan José**: Puesta a punto del proyecto *Java*, realización del ejercicio 1, ejercicio 2 y redacción de la memoria.
  - **Tiempo invertido:** 6:00 horas.

2. Ejercicio 1

Para el primer ejercicio se ha creado un proyecto Java y se han añadido las dependencias de *Apache Jena* y *Twitter4J* al proyecto. El siguiente paso a partir de la plantilla proporcionada junto con el enunciado de la práctica es crear la *Query* para conocer si una cadena de caracteres se corresponde con el nombre de algún jugador de fútbol. Para ello se ha decidido utilizar una *Query* del tipo *ASK* ya que la respuesta a este tipo de consultas es un valor *booleano*. La consulta realizada se basa en la interrogación de distintas tripletas (utilizando *UNION*) a la fuente de datos de *live.dbpedia.org* para obtener la respuesta de si una cadena se corresponde con el nombre de un jugador.

Para ello, en una primera versión del programa (Listing 4.1) se utilizan dos propiedades (`rdt:type` y `dct:subject`) cuyo nodo objeto son recursos pertenecientes a jugadores de fútbol (es decir, cualquier futbolista), poniendo como condición en la consulta que el nombre de este objeto (propiedad `rdfs:label`) sea el de la tendencia. A la hora de enlazar con el nombre, se ha añadido "@en" ya que el campo de nombre está indicado con un lenguaje, en caso contrario la consulta no obtiene ningún resultado.

Para evitar que el lenguaje sea exclusivamente inglés (@en), se ha desarrollado una segunda versión del programa (Listing 4.2) en la que se ha usado la lógica de *LIKE* de *SQL*. Para ello, se ha usando una *regrex* en *SPARQL* que acepta cualquier valor en el campo `rdfs:label` del tipo ¡CADENA¿seguido de cualquier carácter 0 o más veces.

De esta manera nos evitamos el problema del idioma (@en) pero aparece otro notable, ya que puede admitir como futbolistas algunas cadenas que realmente no lo son, como por ejemplo la cadena vacía.

La estructura de las tripletas de estas versiones es la siguiente (Figura 1):

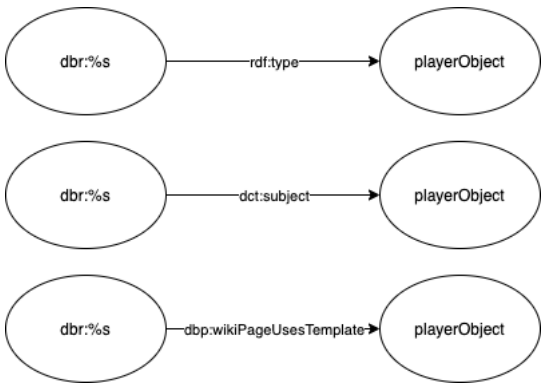

Figura 1: Grafo RDF de la consulta a realizar

Para evitar los errores que generaban las versiones anteriores, se ha desarrollado una tercera y última versión de la consulta *SPARQL* en la que no se hace uso del campo `rdfs:label` y se accede al objeto directamente utilizando "`dbr:<cadena>rdf:type <tipo-futbolista>`" (Listing 4.3). Además, se consigue que la consulta sea menos costosa en tiempo de ejecución.

Para permitir que también se tengan en cuenta las redirecciones entre distintas páginas de la *dbpedia*, se ha utilizado una *selectQuery* (Figura 2) (Listing 4.4). Esta consulta se realiza en caso de que la consulta *ASK* anterior no haya obtenido ningún resultado.

A diferencia de la consulta *ASK* anterior, la fuente de datos usada en esta es *dbpedia.org*, ya que con la versión *live* de la misma, aparecían problemas con las redirecciones. Por ello, se han tenido que cambiar las direcciones usadas en algunos prefijos.

En la consulta se obtienen todos los recursos que a los que la tendencia se relacione mediante `dbo:wikiPageRedirects` y que además sean futbolistas (indicado mediante `rdf:type` como en la consulta *ASK*).

De esta manera, si la consulta ha obtenido algún resultado, esa tendencia es un jugador de fútbol.

Figura 2: Grafo de la imagen de la *Select Query*

3. Ejercicio 2

Para la realización de este ejercicio se parte de que ambos miembros del equipo ya tenían una cuenta de *Twitter* para desarrolladores creada por una asignatura anterior. Se han utilizado los códigos de acceso regenerándolos desde el panel de control de la aplicación de *Twitter* (Figura 3).

Figura 3: Generación de *tokens* desde panel de control de *Twitter Developers*

Para introducir los códigos de acceso en la aplicación, ha sido necesario crear el archivo `twitter4j.properties` el cual posee 4 variables a las que se les debe asignar las claves de acceso. Para `oauth.consumerKey` y `oauth.consumerSecret`, se usa lo obtenido en *API Key & secret*; Para `oauth.accessToken` y `oauth.accessTokenSecret`, se obtiene de *Access token & secret*.

La metodología seguida ha sido: Mirar la API de *Twitter* para buscar los *endpoints* necesarios y posteriormente traducir esto a la librería *Twitter4J*. La traducción era directa y se ha utilizado la documentación *Javadoc* de la misma.

Para obtener las tendencias a nivel nacional se necesita el *endpoint* `/trends/place` y recibe como parámetro de la consulta un identificador `id` con un valor *WOEID (Where On Earth IDentifier)* para referenciar la ubicación donde buscar las tendencias. Este identificador hay que obtenerlo utilizando otro endpoint de la API (`/trends/available`)

que devuelve una lista con las ubicaciones y su identificador *WOEID*.

La traducción a *Twitter4J* fue directa y se puede encontrar en el Listing 4.5.

Para obtener el *WOEID* de españa, se ha llamado una vez a la función `getAvailableTrends()` de *twitter4j* y se ha obtenido el corresponidente.

Como algunas tendencias van precedidas de "#", se ha eliminado de las tendencias obtenidas para evitar problemas realizando las consultas de *SPARQL*.

4. Urls de interés

- Documentación de la *API* de *Twitter* para obtener las tendencias por lugar: (enlace)
- Documentación de la *API* de *Twitter* para obtener los lugares donde obtener las tendencias: (enlace)
- Documentación *Javadoc* de la librería de *Twitter4J*: (enlace)

Anexo 1: Códigos

4.1. Primera versión de la consulta askQuery

```
String.format("PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
    "PREFIX yago: <http://dbpedia.org/class/yago/> " +
    "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
    "PREFIX dct: <http://purl.org/dc/terms/>" +
    "PREFIX dbc: <http://dbpedia.org/page/Category:>" +
    "PREFIX dbp: <http://dbpedia.org/property/wikiPageUsesTemplate>" +
    "PREFIX dbt: <http://dbpedia.org/resource/Template:>" +
    "ASK WHERE {" +
    "{ ?x rdf:type yago:FootballPlayer110101634 . ?x rdfs:label \"%s\"@en }" +
    "UNION" +
    "{ ?x rdf:type yago:LaLigaFootballers . ?x rdfs:label \"%s\"@en }" +
    "UNION" +
    "{ ?x dct:subject dbc:LaLigaFootballers . ?x rdfs:label \"%s\"@en }" +
    "UNION" +
    "{ ?x dct:subject dbc:Spanish_footballers . ?x rdfs:label \"%s\"@en }" +
    "UNION" +
    "{ ?x dct:subject dbc:Spain_international_footballers . ?x rdfs:label \"%s\"@en
        }" +
    "UNION" +
    "{ ?x dbp:wikiPageUsesTemplate dbt:FIFA_player . ?x rdfs:label \"%s\"@en }" +
    "UNION" +
    "{ ?x dbp:wikiPageUsesTemplate dbt:UEFA_player . ?x rdfs:label \"%s\"@en } }"
    , tendenciaWithSpaces, tendenciaWithSpaces,
    tendenciaWithSpaces, tendenciaWithSpaces,
    tendenciaWithSpaces, tendenciaWithSpaces,
    tendenciaWithSpaces, tendenciaWithSpaces);
```

4.2. Segunda version de la consulta askQuery

```
String.format("PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
    "PREFIX yago: <http://dbpedia.org/class/yago/> " +
    "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
    "PREFIX dct: <http://purl.org/dc/terms/>" +
    "PREFIX dbc: <http://dbpedia.org/page/Category:>" +
    "PREFIX dbp: <http://dbpedia.org/property/wikiPageUsesTemplate>" +
    "PREFIX dbt: <http://dbpedia.org/resource/Template:>" +
    "ASK WHERE {" +
    "{ ?x rdf:type yago:FootballPlayer110101634 . ?x rdfs:label ?y . FILTER regex(?
        y, \"%s.*\") }" +
    "UNION" +
    "{ ?x rdf:type yago:LaLigaFootballers . ?x rdfs:label ?y . FILTER regex(?y, \"%
        s.*\") }" +
    "UNION" +
    "{ ?x dct:subject dbc:LaLigaFootballers . ?x rdfs:label ?y . FILTER regex(?y,
        \"%s.*\") }" +
    "UNION" +
    "{ ?x dct:subject dbc:Spanish_footballers . ?x rdfs:label ?y . FILTER regex(?y,
        \"%s.*\") }" +
    "UNION" +
    "{ ?x dct:subject dbc:Spain_international_footballers . ?x rdfs:label ?y .
        FILTER regex(?y, \"%s.*\") }" +
```

```
    "UNION" +
    "{ ?x dbp:wikiPageUsesTemplate dbt:FIFA_player . ?x rdfs:label ?y . FILTER
        regex(?y, \"%s.*\") }" +
    "UNION" +
    "{ ?x dbp:wikiPageUsesTemplate dbt:UEFA_player . ?x rdfs:label ?y . FILTER
        regex(?y, \"%s.*\") }}",
    tendenciaWithSpaces, tendenciaWithSpaces,
    tendenciaWithSpaces, tendenciaWithSpaces,
    tendenciaWithSpaces, tendenciaWithSpaces,
    tendenciaWithSpaces, tendenciaWithSpaces);
```

4.3. Tercera version de la consulta askQuery

```
String.format("PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
    "PREFIX yago: <http://dbpedia.org/class/yago/> " +
    "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
    "PREFIX dct: <http://purl.org/dc/terms/>" +
    "PREFIX dbc: <http://dbpedia.org/page/Category:>" +
    "PREFIX dbp: <http://dbpedia.org/property/wikiPageUsesTemplate>" +
    "PREFIX dbt: <http://dbpedia.org/resource/Template:>" +
    "PREFIX dbo: <http://live.dbpedia.org/ontology/wikiPageRedirects>" +
    "PREFIX dbr: <http://dbpedia.org/resource/>" +
    "ASK {" +
    "{ dbr:%s rdf:type yago:FootballPlayer110101634}" +
    "UNION" +
    "{ dbr:%s rdf:type yago:LaLigaFootballers}" +
    "UNION" +
    "{ dbr:%s dct:subject dbc:LaLigaFootballers}" +
    "UNION" +
    "{ dbr:%s dct:subject dbc:Spanish_footballers}" +
    "UNION" +
    "{ dbr:%s dct:subject dbc:Spain_international_footballers}" +
    "UNION" +
    "{ dbr:%s dbp:wikiPageUsesTemplate dbt:FIFA_player}" +
    "UNION" +
    "{ dbr:%s dbp:wikiPageUsesTemplate dbt:UEFA_player}}"
    , tendenciaWithoutSpaces, tendenciaWithoutSpaces,
    tendenciaWithoutSpaces, tendenciaWithoutSpaces,
    tendenciaWithoutSpaces, tendenciaWithoutSpaces);
```

4.4. Consulta selectQuery para redirecciones

```
String.format("PREFIX␣dbo:␣<http://dbpedia.org/ontology/>" +
    "PREFIX␣dbr:␣<http://dbpedia.org/resource/>" +
    "PREFIX␣rdfs:␣<http://www.w3.org/2000/01/rdf-schema#>" +
    "PREFIX␣rdf:␣<http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
    "PREFIX␣yago:␣<http://dbpedia.org/class/yago/>" +
    "PREFIX␣umbel-rc:␣<http://umbel.org/umbel/rc/>" +
    "SELECT␣DISTINCT␣?redirects␣" +
    "WHERE␣" +
    "{␣{dbr:%s␣dbo:wikiPageRedirects␣?redirects␣.␣?redirects␣rdf:type␣dbo:
        SoccerPlayer␣}" +
    "UNION" +
    "{␣dbr:%s␣dbo:wikiPageRedirects␣?redirects␣.␣?redirects␣rdf:type␣yago:
        FootballPlayer110101634}" +
    "UNION" +
    "{␣dbr:%s␣dbo:wikiPageRedirects␣?redirects␣.␣?redirects␣rdf:type␣umbel-rc:
        SoccerPlayer}}",
    tendenciaWithoutSpaces, tendenciaWithoutSpaces, tendenciaWithoutSpaces,
        tendenciaWithoutSpaces);
```

4.5. Función para obtener las tendencias de *Twitter*

```
private static List<String> getTwitterTrends() throws TwitterException {
    Twitter twitter = TwitterFactory.getSingleton();
    Trends dailyTrends;
    List<String> list = new ArrayList<>();
    // woeid de espa a se ha obtenido usando la funci n que se encuentra
        comentada encima
    dailyTrends = twitter.getPlaceTrends(23424950);

    // Recorremos las tendencias obtenidas
    for (Trend tren : dailyTrends.getTrends()) {
        // Se eliminan los posibles '#'
        list.add(tren.getName().replace("#", ""));
    }
    return list;
}
```