Recomendación de contenido en Twitter

Samuel Rocha, Alain Cabrera, Mauricio González

2 de abril de 2016

Resumen

En este documento se propone un sistema de recomendación de contenido en Twitter basado en la actividad de la red del usuario. En primera instancia, se obtuvieron los los tweets compartidos de las personas que se siguen y con una librería de python se obtuvieron los temas principales de dichos tweets; en segunda instancia, se colocaron en la base de datos neo4j; finalmente, con Cypher se hicieron recomendaciones con respecto a los link compartidos por los seguidores del usuario.

Introducción y Motivación

En la actualidad, el problema de recomendación -películas, música, productos y contenido- ha adquirido gran importancia. Las empresas invierten cada vez más fondos en la investigación y desarrollo de un sistema de recomendación para optimizar sus utilidades.
En nuestro caso, buscamos crear un sistema de recomendación de artículos de lectura para los usuarios de Twitter. En especial los artículos compartidos por el círculo inmediato en la red social.
Para el desarrollo del sistema de recomendación se utilizó software libre: Python y Neo4j.

Metodología

Obtención de datos

Los datos usados para crear el sistema de recomendación de contenido fueron obtenidos de Twitter por medio de la librería Tweepy de Python y de la API de la red social. Nuestro interés se centra en links compartidos por nuestros *followers*. Lo que se hizo fue obtener una lista de *followeds*, para cada uno de ellos se obtuvieron 200 twitts y se encontraron todos los URLs compartidos en ellos. Los links obtenidos se almacenan en un archivo csv junto con el nombre del usuario que lo compartió.

Análisis de contenido

Una vez obtenidos los urls, el siguiente paso es tener una idea del contenido de dichas direcciones (supondremos que cada dirreción alberga un artículo de noticia). Hicimos este análisis utilizando la libreria newspaper3k de Python. Como resultado obtenemos el autor de cada artículo y un conjunto de keywords del mismo.

Modelo en Neo4j

El modelo de datos utlizado consta de cuatro tipos de nodos (user, url, author, keyword) y tres tipos de aristas que describen la relación entre los nodos. Se utilizó la librería py2neo de Python para conectar nuestra instancia de Neo4j y se agregaron los datos por medio de consultas de Cypher.

En la siguiente figura se puede observar los nodos conectados por 3 aristas: Cada usuario que comparte alguna url está escrito por cierto autor y habla sobre algún tema.

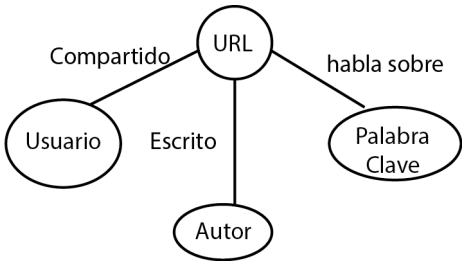

Figura 1: Diagrama general del grafo

Después de determinar el modelo que se iba a utilizar para poder determinar las url compartidas por los usuarios, se importaron a la base de datos de Neo4j, pero en una prueba que se realizó para un usuario que tenía seguía a 2000 personas se alcanzaba el rate limite, ya que se checaban 200 twitts de cada usuario y se checaba si compartía un artículo. Decidimos usar una cuenta que seguía a 80 personas para realizar un grafo de menor tamaño, pero la idea es escalable para todos los usuarios. A continuación se muestra el grafo ejemplo de 10 usuarios que han compartido artículos.

Sistema de Recomendación

En la figura 3 se puede observar el diagrama del usuario jgomezjunco(en color amarillo) que representa el grafo de los últimos articulos compartidos(en color rojo) al igual que las palabras más importantes del artículo(en color rosa). EL objetivo está en determinar, dados los artículos que compartió, extraer

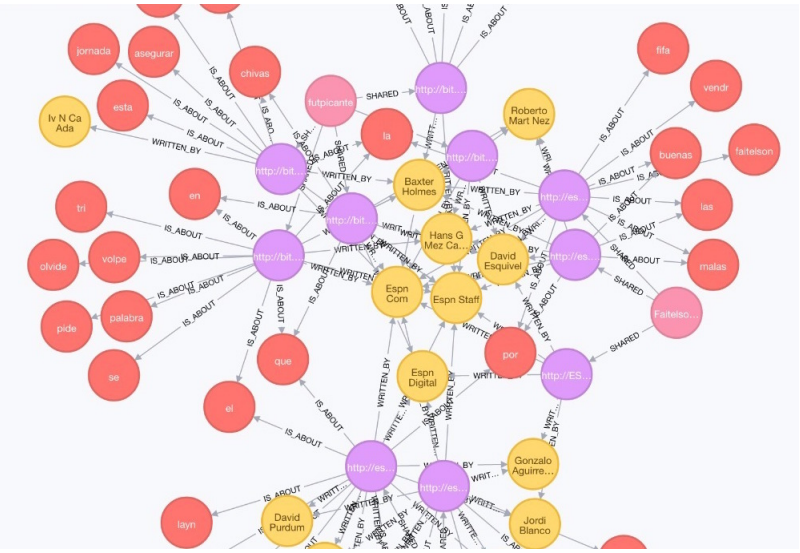

Figura 2: Diagrama general del grafo

las palabras clave para determinar cuales son los artículos compartidos por los usuarios que sigue y los usuarios que lo siguen, que otras lecturas le serían factibles leer contando su gusto demostrado previamente.

Si se analiza previamente el grafo, podemos observar que tiene una tendencia a compartir árticulos deportivos que, dependiendo a las personas que siga, se puede determinar que artículos serán de su agrado. Ésto tiene sentido, ya que el usuario que se decidió analizar es un conductor deportivo.

Extracción de palabras clave

La descripción de los artículos se hizo con una librería de python que se llama newspaper extrae las palabras más importantes de los artículos. En primera instancia, se obtienen todas las palabras, frases o conceptos que potencialmente pueden ser palabras clave; despues, por cada candidato se calculan sus propiedades que indican si pueden ser una palabra clave; finalmente se hace una puntuación y se selección las palabras clave: normalmente se utilizan técnicas de aprendizaje de máquina para determinar la puntuación que describe la probablidad de que sean escogidas.
Para acelerar el proceso de selección se utilizan varios parámetros como utilizar la frecuencia mínima de los candidatos, el número máximo y mínimo de las palabras y se pueden *stemmatizar* (proceso de sacar la raiz de una palabra).

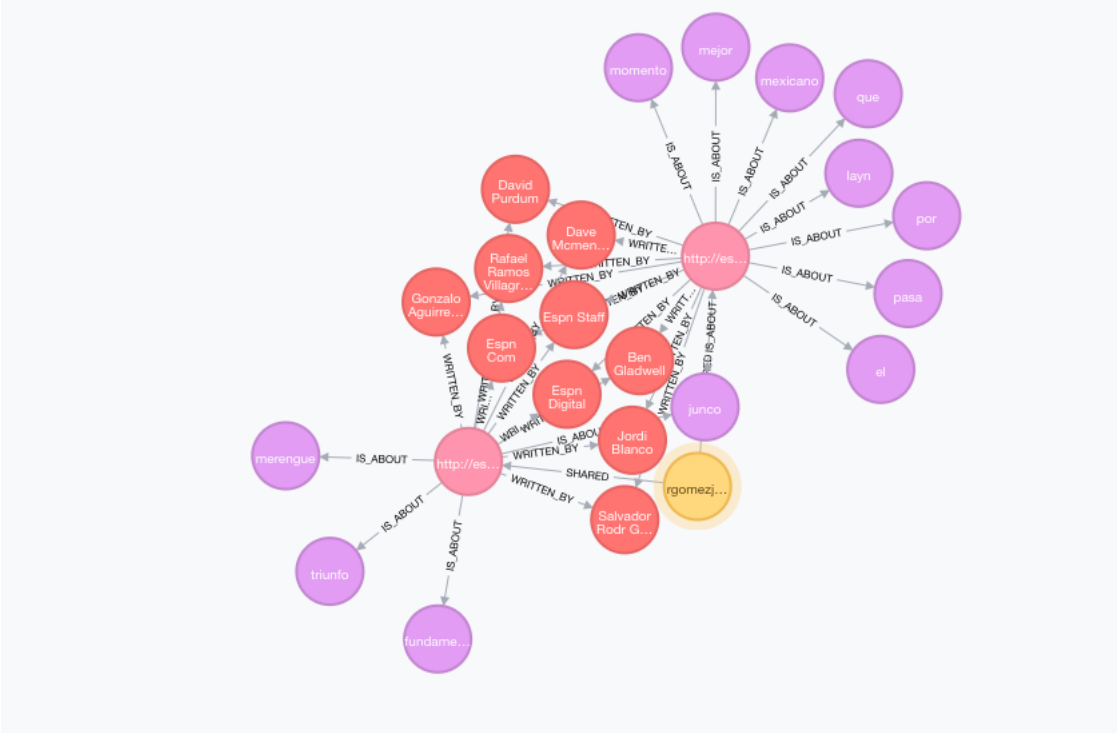

Figura 3: Diagrama de un usuario con los artículos que ha compartido

Secuencias de Cypher

Ya que se cuenta con un grafo de un usuario, sus seguidores y los artículos que han compartido, se busca cuales son los artículos que fueron compartidos por los la red que serían de gran utilidad dependiendo del historial de usuario.

```
MATCH (u:User {username: 'samo2704'})-[:SHARED]->(url:URL)
MATCH (url)-[r:IS_ABOUT]->(kw:Keyword)<-[r2:IS_ABOUT]-(u2:URL)
WHERE NOT (u)-[:SHARED]->(u2)
WITH u2.url as article, count(r2) as score
RETURN article, score ORDER BY score DESC LIMIT 10;
```

Figura 4: Instrucción Cypher para la recomendación a un usuario

Como se puede observar en la instrucción, buscamos a un usuario en especial y todas los articulos que ha compartido. De esta busqueda queremos saber las palabras clave de las páginas que ha compartido para poder compararlas con los usuarios que sigue.

Como se puede observar en la anterior tabla, aparecen varios links que se

article	score
http://bit.ly/1RRt2j3	2
http://ow.ly/10bz3L	2
http://bit.ly/1opUBSF	2
http://ow.ly/106sDM	2
http://ow.ly/10bELy	1
http://es.pn/1RsY4vz	1
http://bit.ly/22lvi15	1
http://somosinvictos.com/2016/03/21/ni-como-cr7-ni-como-messi-el-hijo-de-david-barral-delantero-espanol-quiere-ser-como-chicharito/	1
http://bit.ly/1SnoDyL	1
http://bit.ly/1qmvFx5	1

Returned 10 rows in 148 ms.

Figura 5: Tabla de las recomendaciones para un usuario

recomiendan al usuario por su gusto en deportes y que 2 de las personas que sigue han compartido.

El único inconveniente que apareció fue que probamos esto para nuestras cuentas de twitter y solo una de ellas no alcanzaba el limite otorgado por twitter. El análasis realizado es escalable, pero solo se realizó la recomendación para algunos usuarios.

Conclusiones

Una de las desventajas de dar una recomendación en relación a los Url que se han compartido es que los nuevos usuarios de Twitter o gente que no ha compartido muchas cosas, no tendrá recomendaciones, ya que no tendrá tópicos que se puedan acomodar a las recomendaciones que pudieran hacerce.

Fue muy interesante poder practicar las herramientas que vimos en clase como poder bajar datos de twitter, poder convertirlos en una base de datos de grafos y finalmente utilizar secuencias de cypher y poder analizar los datos que se bajaron. En general, no es tan fácil utilizar las secuencias de Cypher, pero después de un tiempo de utilizarlas, fue más sencillo realizar las peticiones.
Otro problema con el que nos enfrentamos fue el limite que te pone twitter para bajar datos. Como recorriamos muchos de los post que ponían los usuarios, el árbol de petición crecía exponcialmente en relación a los seguidores. El grafo final lo realizamos con una cuenta que solo seguía a 80 personas y analizamos dicho grafo.
Para posteriores análisis, se puede implementar análisis de sentimientos y de-

terminar si los artículos recomendados son positivos y negativos.

Referencias

[1] Panzarino, Onofrio, 'Learning Cypher', *Packt Publishing Ltd.* (2014).

[2] Robinson, Ian and Webber, Jim and Eifrem, Emil, 'Graph Databases: New Opportunities for Connected Data', *O'Reilly Media, Inc.* (2015).