Universidad de Alcalá
Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

Análisis de datos en redes sociales: Caso de estudio aplicado en Twitter

ESCUELA POLITECNICA SUPERIOR

Autor: David Márquez Mínguez

Tutor: Juan José Cuadrado Gallego

2021

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

Trabajo Fin de Grado

Análisis de datos en redes sociales: Caso de estudio aplicado en Twitter

Autor: David Márquez Mínguez

Tutor: Juan José Cuadrado Gallego

Tribunal:

Presidente: Name of the tribunal president

Vocal 1º: Name of the first vocal

Vocal 2º: Name of the second vocal

Fecha de depósito: X de X de 2021

A nuestros alumnos pasados, presentes y futuros...

"Empieza haciendo lo necesario, luego haz lo posible y de pronto empezarás a hacer lo imposible."
Francisco de Asís

Agradecimientos

A todos los que la presente vieren y entendieren.

Inicio de las Leyes Orgánicas. Juan Carlos I

Aqui va la parte de agradecimientos.....

Resumen

Resumen..... correo de contacto: David Márquez Mínguez <david.marquez@edu.uah.es>.

Palabras clave: Trabajo fin de /grado, LaTeX, soporte de español e inglés, hasta cinco....

Abstract

Abstract..... contact email: David Márquez Mínguez <david.marquez@edu.uah.es>.

Keywords: Bachelor final project , LaTeX, English/Spanish support, maximum of five....

Resumen extendido

Con un máximo de cuatro o cinco páginas. Se supone que sólo está definido como obligatorio para los TFGs y PFCs de UAH.

Índice general

Índice de figuras

Índice de tablas

Índice de listados de código fuente

Índice de algoritmos

Capítulo 1

Ejemplo práctico: Text Mining con R

En este capítulo se va a desarrollar un caso práctico sobre minería de textos, con el fin de corroborar su utilidad y su importancia en el mundo actual. Para dicho análisis necesitaremos una fuente de información de donde extraer los textos a analizar, dicha fuente bien podría ser un artículo de opinión, discursos transcritos... El conjunto de textos a analizar, serán publicaciones de los últimos tres presidentes de Estados Unidos, en este caso en Twitter, con el fin de analizar dichas publicaciones y realizar un estudio sobre las mismas.

Twitter es actualmente una dinámica e ingente fuente de contenidos que, dada su popularidad e impacto, se ha convertido en la principal fuente de información para estudios de Social Media Analytics. Multitud de empresas emplean este tipo de técnica para obtener información muy valiosa de individuos o de corporaciones. Análisis de reputación de empresas, productos o personalidades, estudios de impacto relacionados con marketing, extracción de opiniones y predicción de tendencias son sólo algunos ejemplos de aplicaciones.

Además de R existen otros lenguajes de programación como Python, MatLab u Octave. Si bien Python es el lenguaje que domina en este ámbito, este análisis se realizará a través de la programación en R, pues contiene tanto librerías, como paquetes que facilitan y extienden sus capacidades como herramienta de análisis de texto.

1.1 Introducción

Tal y como ocurre en muchas redes sociales, Twitter otorga la posibilidad de compartir sus datos tanto con empresas como con desarrolladores y/o usuarios particulares. Aunque en la mayoría de casos se trata de web Services API, con frecuencia existen librerías que permiten interactuar con la API desde diversos lenguajes de programación.

La forma en la que Twitter permite acceder a su contenido es a través de lo que se conoce como Twitter App, al crear dicha Twitter App, se adquieren una serie de claves y tokens de identificación que permiten acceder a la aplicación y consultar la información necesaria.

Algo que se deberá tener en cuenta durante este análisis, es que Twitter tiene una normativa que regula la frecuencia máxima de peticiones, así como la cantidad máxima de tweets que se pueden extraer. Durante la fase de extracción de la información, se deberá tener en cuenta dichos limites con el objetivo de evitar ser sancionado por la organización.

1.2 Acceso a la API

Para acceder a la API de Twitter, como se indica en la documentación de la misma, existen dos métodos de acceso Oauth2 y Oauth1a. El acceso con cada uno de ellos dependerá del tipo de información que se desee extraer [1]. Como veremos a continuación, el metodo de acceso elegido será Oauth1a.

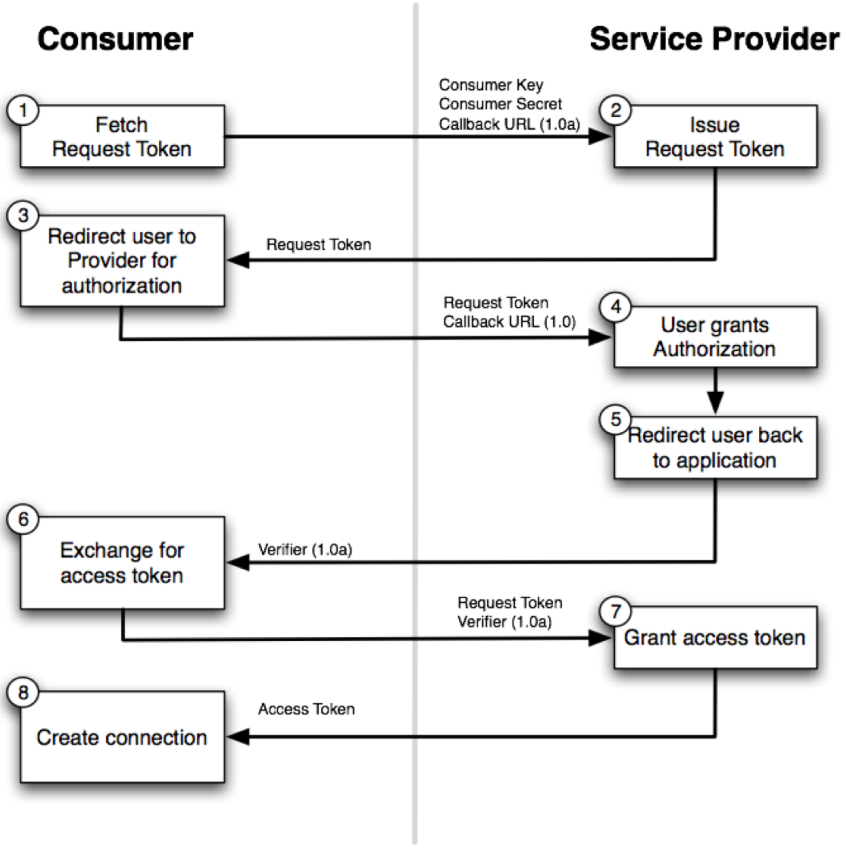

Figura 1.1: Funcionamiento de Oauth1a

Puesto que se desea extraer información específica de algunos usuarios de la plataforma, el metodo de acceso será a través de Oauth1a. Así pues, OAuth1a para funcionar requiere de cuatro elementos que provienen de la aplicación de Twitter creada por el usuario desarrollador, estos elementos son:

1. Clave del consumidor
2. Clave del consumidor secreta
3. Token de acceso
4. Token de acceso secreto

Todos estos recursos deberán ser proporcionados por la plataforma de Twitter para desarrolladores. Una vez se disponen de dichos recursos, realizamos el proceso de identificación y obtención de tokens [2], no sin antes cargar las librerías necesarias para el desarrollo de dicho caso práctico.

```
>   #En primer lugar se cargan las librerias necesarias.
>   library(twitteR)
>   library(tidyverse)
>   library(knitr)
```

Una vez que las librerías necesarias están cargadas, se procede a crear las variables que almacenarán los recursos de acceso. A conitnuación, se llama a la función de acceso, proporcionando los parametros anteriormente definidos. Esta función envuelve las funciones de protocolo de enlace de autenticación OAuth 1.0 del paquete httr para una sesión de Twitter en concreto. [3].

```
>   #Ahora se definen ciertas variables que almacenaran la informacion de acceso
>   nombre_app <- "MarquezDavidTFG"
>   clave <- "bsS2cbbZe7BDsxFPLYRM0GKJ8"
>   clave_secreta <- "WvUilwEZbgJhHiUgpjyboQcQHYSSCNKwImHF1TeINe9aWskkDo"
>   token_acceso <- "3147132436-kFG9XkuWsdI8n1KPVZQTOrf6rw45lrqPEPoUXPr"
>   token_acceso_secreto <- "zK8dyBAgB4sTrRhjVBXmNCAjX4QQvCUfQIXckQmTrcAix"

>   #Acceso a la API
>   options(httr_oauth_cache=TRUE)
>   setup_twitter_oauth(consumer_key = clave, consumer_secret = clave_secreta, access_to
```

1.3 Extracción y carga de datos

Una vez que se ha conseguido acceder a la API a través de la aplicación para desarrolladores proporcionada por Twitter, se procede con la fase de extracción de datos. Como se ha indicado en la sección 1.1, en Twitter existe una normativa que regula la frecuencia máxima de peticiones, así como la cantidad máxima de tweets que se pueden extraer [4]. Para el tipo de análisis que se quiere realizar en este trabajo, se pretenden obtener la mayor cantidad de tweets posible, puesto que el número máximo de publicaciones que se pueden extraer por consulta son 200, se sigue la siguiente estrategia:

1. Toda publicación tiene un identificador numérico global, que sigue un orden temporal. Esto permitirá identificar y distinguir los tweets mas recientes de los más antiguos.
2. Es posible recuperar y trabajar solo con publicaciones antiguas, empelando como parámetro de la función el identificador de cada tweet.
3. Antes de cada consulta, se lee el fichero donde se almacenan las publicaciones y se identifica el ID del ultimo tweets recuperado. Si no existe dicho fichero de almacenamiento para el usuario en cuestión, se crea uno.
4. Se realiza una nueva consulta empleando como argumento maxID el identificador recuperado en el paso anterior.
5. Se incorporan los nuevos datos al archivo de almacenamiento.

Una vez ha quedado claro el proceso de extracción, se pasa a crear una función que se encargue de dicha tarea [1]. Cada vez que se ejecute dicha función, se recuperan nuevo tweets y se almacenan junto a los ya extraídos previamente.

```
>   extraccion_tweets(usuario  = "JoeBiden", maxtweets  = 200)
>   extraccion_tweets(usuario  = "DonaldTrump", maxtweets  = 200)
>   extraccion_tweets(usuario  = "BarackObama", maxtweets  = 200)
```

[1] Se puede consultar más información sobre la implementación de dicha función en el apéndice A.1, donde se detalla el funcionamiento y uso de la misma.

Capítulo 2

Presupuesto

Blah, blah, blah.

Bibliografía

[1] D. Spring, "Service provider 'connect' framework," https://docs.spring.io/spring-social/docs/1.0.0.RC1/reference/html/serviceprovider.html [Ultimo acceso 16/marzo/2021].

[2] S. Huppmann, https://github.com/ropensci/rtweet/issues/251 [Ultimo acceso 17/marzo/2021].

[3] RDocumentation, "Sets up the oauth credentials for a twitter session," https://www.rdocumentation.org/packages/twitteR/versions/1.1.9/topics/setup_twitter_oauth [Ultimo acceso 17/marzo/2021].

[4] Twitter, "Rate limits," https://developer.twitter.com/en/docs/rate-limits [Ultimo acceso 19/marzo/2021].

Apéndice A

Funciones implementadas

A.1 Función de extracción de Tweets

Listado A.1: Extracción de Tweets empleando una funcion en R

```
extraccion_tweets <- function(usuario, maxtweets = 100, archivoSalida= NULL){

  #Se crea el nombre de archivo por defecto
  if(is.null(archivoSalida)){
        archivoSalida <- paste0("datos_tweets_", usuario, ".csv")
  }

  #Se comprueba si el archivo csv existe o no
  if(!(archivoSalida %in% list.files())){
    datos_new <- searchTwitter(usuario, n = maxtweets, exclude:retweets)
    datos_new_df <- twListToDF(datos_new)
    write.csv(datos_new_df, archivoSalida)
  }else{
    #Obtengo los datos antiguos
    datos_old <- read.csv(file = archivoSalida)

    #Calculo el id del nuevo tweet a obtener
    ultimo_id <- tail(datos_old, 1)["id"] %>% pull()
    ultimo_id = ultimo_id + 1

    datos_old <- map_if(.x = datos_old, .p = is.numeric, .f = as.character)

    datos_new <- searchTwitter(usuario, n = maxtweets, maxID = ultimo_id, exclude:retweets)
    datos_new <- map_if(.x = datos_new, .p = is.numeric, .f = as.character)

    datos_new_df <- as.data.frame(datos_new)
    datos_old_df <- as.data.frame(datos_old)

    #Se concatenan los nuevos tweets con los antiguos
    datos_concatenados <- bind_rows(datos_old_df, datos_new_df)

    write.csv(datos_concatenados, archivoSalida)
  }
}
```

El objetivo fundamental de esta función es extraer los tweets publicados por un usuario y almacenarlos en un archivo csv. En caso de que exista un archivo con el mismo nombre, se lee y se concatena el nuevo contenido con el antiguo. Los argumentos de entrada, son los siguientes:

1. usuario: reprsenta el identificador del usuario de Twitter.
2. maxtweets: cantidad de tweets que se van a recuperar.
3. archivoSalida: nombre del fichero de salida.

Se debe tener cuidado de no recuperar tweets repetidos, para ello se ha creado una variable que almacena el id del ultimo tweet. Con esto cada vez que se quieran recuperar nuevos tweets, se aumentará en uno dicha variable y se procederá como hasta ahora.

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá