Tipología y ciclo de vida de los datos. Práctica 1

AnaC

November 12, 2017

Objetivo

Información de diagnóstico médico basado en big data.

Se busca inicialmente si en Twitter se habla y se relacionan esos conceptos.

Datos de Twitter utilizando R

Se utiliza RStudio. Para extraer los datos de Twitter se usa la librería *twitteR* siguendo las indicaciones recomendadas en la asignatura de: http://bogdanrau.com/blog/collecting-tweets-using-r-and-the-twitter-search-api (http://bogdanrau.com/blog/collecting-tweets-using-r-and-the-twitter-search-api)

Usuario desarrollador de Twitter

Se crea la aplicación de twitter *diag-salud* en http://apps.twitter.com (http://apps.twitter.com) Y se generan los códigos de acceso de la aplicación (tokens) para que pueda utilizar la API de Twitter.

API Twitter desde RStudio

Utilizando RStudio, se instalan los paquetes: (comentados para no reinstalarlos) install.packages("twitteR") install.packages("RCurl") install.packages("RJSONIO") install.packages("stringr")

Se prepara la autenticación e inicio de sesión

```
library(twitteR)
library(RCurl)
```

```
## Loading required package: bitops
```

```
library(RJSONIO)
library(stringr)
api_key <- "PS3yvXxX89E2W08o6za6oBR98"
api_secret <- "0wjcRhIKNf1CCCUdsTZNv2sRZdiX6Tspv59a9bOvWbFfQ7zpUU"
token <- "929642822340431872-kXMXgOexjaSQ3nUrAdUoIdNX4yORb1x"
token_secret <- "5OiOHdtDwu3ljTpV2p0imNwWLd3O1wJru7lt35fSO1W7l"
options(httr_oauth_cache=T)
# option = TRUE para evitar que setup.. pregunte si usa el fichero .httr-oauth
setup_twitter_oauth(api_key, api_secret, token, token_secret)
```

```
## [1] "Using direct authentication"
```

Se usará el método *searchTwitter(…)* pero probando antes via web en la dirección http://twitter.com/search (http://twitter.com/search).

Consulta:

```
twts <- searchTwitter("'#bigdata' 'health' 'diagnosis'", n=200)
```

```
## Warning in doRppAPICall("search/tweets", n, params = params,
## retryOnRateLimit = retryOnRateLimit, : 200 tweets were requested but the
## API can only return 8
```

```
length(twts)
```

```
## [1] 8
```

La lista de resultados se convierte a tipo *data frame*:

```
twtsdf <- twListToDF(twts)
nrow(twtsdf)
```

```
## [1] 8
```

Se revisa los tipos de datos:

```
names(twtsdf)
```

```
##  [1] "text"          "favorited"     "favoriteCount" "replyToSN"
##  [5] "created"       "truncated"     "replyToSID"    "id"
##  [9] "replyToUID"    "statusSource"  "screenName"    "retweetCount"
## [13] "isRetweet"     "retweeted"     "longitude"     "latitude"
```

```
str(twtsdf)
```

```
## 'data.frame':    8 obs. of  16 variables:
##  $ text         : chr  "RT @MopeadEU: Great news ! @billgates commits $50m for  #dementia #research \n#clinicaltrials #di
agnosis #BigDa"| __truncated__ "Great news ! @billgates commits $50m for  #dementia #research \n#clinicaltrials #diagnosis #
BigData<U+0085> https://t.co/LpZeOj0xk3" "Thanks @billgates for investing $50m in #dementia #research \n#clinicaltrials #dia
gnosis #BigData<U+0085> https://t.co/HvOm939BWB" "RT @jisantangelo: Kudos @GeisingerHealth using #bigdata not just talking a
bout it. Making an impact in #healthc"| __truncated__ ...
##  $ favorited    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ favoriteCount: num  0 5 0 0 0 0 0 0
##  $ replyToSN    : logi  NA NA NA NA NA NA ...
##  $ created      : POSIXct, format: "2017-11-13 17:32:47" "2017-11-13 17:15:15" ...
##  $ truncated    : logi  FALSE TRUE TRUE FALSE FALSE FALSE ...
##  $ replyToSID   : logi  NA NA NA NA NA NA ...
##  $ id           : chr  "930126308318826496" "930121894430498816" "930119638574301184" "929994307427360768" ...
##  $ replyToUID   : logi  NA NA NA NA NA NA ...
##  $ statusSource : chr  "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>" "<a href=
\"http://twitter.com\" rel=\"nofollow\">Twitter Web Client</a>" "<a href=\"http://twitter.com\" rel=\"nofollow\">Twitter Web
Client</a>" "<a href=\"http://twitter.com\" rel=\"nofollow\">Twitter Web Client</a>" ...
##  $ screenName   : chr  "sanz_guille" "MopeadEU" "Annette_Dumas" "RelevantTrack" ...
##  $ retweetCount : num  1 1 0 8 3 8 8 8
##  $ isRetweet    : logi  TRUE FALSE FALSE TRUE TRUE TRUE ...
##  $ retweeted    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ longitude    : logi  NA NA NA NA NA NA ...
##  $ latitude     : logi  NA NA NA NA NA NA ...
```

Un caso:

```
twtsdf[1, c("id", "screenName", "created","text")]
```

```
##                   id  screenName             created
## 1 930126308318826496 sanz_guille 2017-11-13 17:32:47
##
                   text
## 1 RT @MopeadEU: Great news ! @billgates commits $50m for  #dementia #research \n#clinicaltrials #diagnosis #BigData  \n h
ttps://t.co/jJXutX1VOQ<U+0085>
```

```
sum(twtsdf$latitud)
```

```
## [1] NA
```

```
sum(twtsdf$longitude)
```

```
## [1] NA
```

```
twtsdf$created
```

```
## [1] "2017-11-13 17:32:47 UTC" "2017-11-13 17:15:15 UTC"
## [3] "2017-11-13 17:06:17 UTC" "2017-11-13 08:48:16 UTC"
## [5] "2017-11-08 10:24:58 UTC" "2017-11-06 08:20:35 UTC"
## [7] "2017-11-06 08:19:01 UTC" "2017-11-06 08:18:14 UTC"
```

Todos los casos son muy recientes, de este mes de noviembre.

Se prueba variando el tipo de resultado:

```
twtsp <- searchTwitter("'#bigdata' 'health' 'diagnosis'", n=200, resultType="popular")
```

```
## Warning in doRppAPICall("search/tweets", n, params = params,
## retryOnRateLimit = retryOnRateLimit, : 200 tweets were requested but the
## API can only return 0
```

```
twtsp <- searchTwitter("'#bigdata' 'health' 'diagnosis'", n=200, resultType="recent")
```

```
## Warning in doRppAPICall("search/tweets", n, params = params,
## retryOnRateLimit = retryOnRateLimit, : 200 tweets were requested but the
## API can only return 8
```

```
twtsp <- searchTwitter("'#bigdata' 'health' 'diagnosis'", n=200, since="2017-01-01", until="2017-08-31")
```

```
## Warning in doRppAPICall("search/tweets", n, params = params,
## retryOnRateLimit = retryOnRateLimit, : 200 tweets were requested but the
## API can only return 0
```

Tras diversas pruebas parece que el acceso mediante API *searchTwitter* devuelve sólamente resultados recientes. En cambio este último caso (utilizando el filtro de fechas entre enero y agosto de 2.017) probado en http://twitter.com/search (http://twitter.com/search) sí que devuelve resultados.

Se varía el patrón de búsqueda:

```
twtsa <- searchTwitter("'big data' 'health' 'diagnosis'", n=200)
```

```
## Warning in doRppAPICall("search/tweets", n, params = params,
## retryOnRateLimit = retryOnRateLimit, : 200 tweets were requested but the
## API can only return 10
```

```
twtsadf <- twListToDF(twtsa)
nrow(twtsadf)
```

```
## [1] 10
```

Algunos casos se repiten:

```
twtsdf$id
```

```
## [1] "930126308318826496" "930121894430498816" "930119638574301184"
## [4] "929994307427360768" "928206706118725632" "927450628590272513"
## [7] "927450235076595712" "927450034446196736"
```

```
twtsadf$id
```

```
##  [1] "929994307427360768" "929787087854895106" "929751465941504000"
##  [4] "929059303159410689" "929022683014103040" "929019592160276482"
##  [7] "928206706118725632" "927450628590272513" "927450235076595712"
## [10] "927450034446196736"
```

Se unen los dos resultados, eliminando repetidos:

```
twdf <- unique(rbind(twtsdf, twtsadf))
nrow(twdf)
```

```
## [1] 13
```

Los autores:

```
unique(twdf$screenName)
```

```
##  [1] "sanz_guille"    "MopeadEU"       "Annette_Dumas"  "RelevantTrack"
##  [5] "EpicRelevance"  "SmartMedRT"     "UtarSystems"    "mo_mist"
##  [9] "YvesMulkers"    "ianachurch"     "annawoodtechpr" "DigiCatapult"
```

Se guardan los datos actuales en archivo CSV:

```
write.csv(twdf,file="diag-salud-dat1.csv")
```

Se varía de nuevo el patrón de búsqueda utilizando sólo los términos *healthcare* y *big data* y ampliando los posibles registros de resultados a 2000:

```
twtsb <- searchTwitter("'big data' 'healthcare'", n=2000)
```

```
## Warning in doRppAPICall("search/tweets", n, params = params,
## retryOnRateLimit = retryOnRateLimit, : 2000 tweets were requested but the
## API can only return 1616
```

```
twtsbdf <- twListToDF(twtsb)
nrow(twtsbdf)
```

```
## [1] 1616
```

Se unen los resultados como anteriormente, eliminando repetidos:

```
tdf <- unique(rbind(twdf, twtsbdf))
nrow(tdf)
```

```
## [1] 1625
```

Se guardan los datos actuales en archivo CSV. Este archivo incluye el contenido de tweets repetidos, que son retweets (valores de los atributos *isRetweet* como TRUE o *rtweetCount* > 0):

```
write.csv(tdf,file="diag-salud-incRetweets.csv")
```

Se extraen los no retuiteados, se indica el número de tweets y de autores:

```
tudf <- subset(tdf,isRetweet == FALSE)
nrow(tudf)
```

```
## [1] 775
```

```
length(unique(tudf$screenName))
```

```
## [1] 614
```

Se guardan los datos en archivo CSV:

```
write.csv(tudf,file="diag-salud.csv")
```