Twitter

FranciscoPérezHernández

7/4/2017

Crear Credenciales

Lo primero será ir al siguiente enlace https://apps.twitter.com y registrarnos para obtener nuestras credenciales quedando un fichero llamado "credenciales.R" con la siguiente estructura:

```
#Cargamos las librerías
library("ROAuth")
library("base64enc");
library("twitteR");
library("streamR");

#Cargar parámetros de configuración
reqURL <- "https://api.twitter.com/oauth/request_token"
accessURL <- "https://api.twitter.com/oauth/access_token"
authURL <- "https://api.twitter.com/oauth/authorize"
options(httr_oauth_cache=T)

#Cargar las credenciales obtenidas del paso anterior
consumer_key <- "pegar aquí la credencial"
consumer_secret <-"pegar aquí la credencial"
access_token <-"pegar aquí la credencial"
access_secret <-"pegar aquí la credencial"

#Ejecutar la autenticación de TwitteR
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

#streamR authentication
credentials_file <- "my_oauth.Rdata"
if (file.exists(credentials_file)){
  load(credentials_file)
} else {
  cred <- OAuthFactory$new(consumerKey = consumer_key, consumerSecret =
                             consumer_secret, requestURL = reqURL, accessURL = accessURL, authURL = aut
  cred$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))
  save(cred, file = credentials_file)
}
```

Obtener datos de twitter

```
# Cargar la librería específica de TwitterR
library(twitteR);

# Leer el fichero de credenciales creado anteriormente, ¡cuidado con la ruta del fichero!.
source('../credenciales.R')
```

```
## Loading required package: RCurl
```

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

```
## Loading required package: rjson
```

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

```
# Función que permite buscar: #hastag, @usuarios, palabras
tweets <- searchTwitter("#brexit", n=100, lang="en")

# Quedarse solo con el primer tweet para datos concretos del mismo
tweet <- tweets[[1]];
# Mostrar la estructura del tweet
#str(tweet)
# Obtener el texto del tweet:
tweet$getText()
```

```
## [1] "@MandaJJennings @007harvey @dominic_putney @Simplex2014 @LeaveEUOfficial @theresa_may @BorisJohn
```

```
# Obtener información acerca del usuario:
usuario <- getUser(tweet$getScreenName());
# Mostrar la estructura del usuario
#str(usuario)
# Obtener el nombre del usuario
usuario$getName()
```

```
## [1] "angie"
```

Instalación de paquetes necesarios

```
# Instalar el paquete Sentiment
require('pacman')
```

```
## Loading required package: pacman
```

```
#if (!require('pacman')) install.packages('pacman&')
#pacman::p_load(devtools, installr)
#install.Rtools()
#install_url('http://cran.r-project.org/src/contrib/Archive/Rstem/Rstem_0.4-1.tar.gz')
#install_url('http://cran.r-project.org/src/contrib/Archive/sentiment/sentiment_0.2.tar.gz')
if (!require('pacman')) install.packages('pacman')
pacman::p_load(twitteR, sentiment, plyr, ggplot2, wordcloud, RColorBrewer, httpuv, RCurl, base64enc)
options(RCurlOptions = list(cainfo = system.file('CurlSSL', 'cacert.pem', package = 'RCurl')))
#setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
#setup_twitter_oauth(api_key,api_secret)
```

Análisis de sentimientos

Sacar tweets

Lo primero que vamos a hacer será sacar twetts sobre el brexit y sacar de ellos su texto

```
tweets <- searchTwitter("#brexit", n=10000, lang="en")
texto_tweets = sapply(tweets, function(x) x$getText())
```

Limpiado del texto

Vamos a ver un ejemplo de los primeros tweets encontrados de como vamos limpiando el texto

```
head(texto_tweets)
```

```
## [1] "@MandaJJennings @007harvey @dominic_putney @Simplex2014 @LeaveEUOfficial @theresa_may @BorisJohn
## [2] "RT @Far_Right_Watch: These people who keep our #NHS going are increasingly returning home. #Brex
## [3] "RT @politicshome: Bank of England tells City to prepare contingencies for worst-case Brexit scen
## [4] "RT @Dwalingen: The great BIG LIE &amp;amp; CRIME of our era.\n\n#EU #euistheproblem #LeaveEU #Brexit
## [5] "RT @disasterhistory: the impact of #Brexit is so appalling that #TheresaMay dare not tell us. ST
## [6] "RT @trilegseaspider: We urge MPs not to write the Govt a blank cheque for hard #Brexit. Show you
```

```
cat("\nEliminamos retweet\n")
```

```
##
## Eliminamos retweet
```

```
texto_tweets = gsub('(RT|via)((?:\\b\\W*@\\w+)+)', '', texto_tweets)
head(texto_tweets)
```

```
## [1] "@MandaJJennings @007harvey @dominic_putney @Simplex2014 @LeaveEUOfficial @theresa_may @BorisJohn
## [2] ": These people who keep our #NHS going are increasingly returning home. #Brexit will destroy the
## [3] ": Bank of England tells City to prepare contingencies for worst-case Brexit scenario: https://t
## [4] ": The great BIG LIE &amp;amp; CRIME of our era.\n\n#EU #euistheproblem #LeaveEU #Brexit #Nexit #Frex
## [5] ": the impact of #Brexit is so appalling that #TheresaMay dare not tell us. STOP IT NOW https://t
## [6] ": We urge MPs not to write the Govt a blank cheque for hard #Brexit. Show your support &amp;amp; RT
```

```
cat("\nEliminar usuarios\n")
```

```
##
## Eliminar usuarios
```

```
texto_tweets = gsub('@\\w+', '', texto_tweets)
head(texto_tweets)
```

```
## [1] "       Remoaners lost... https://t.co/5NAK8ybIU6"
## [2] ": These people who keep our #NHS going are increasingly returning home. #Brexit will destroy the
## [3] ": Bank of England tells City to prepare contingencies for worst-case Brexit scenario: https://t
## [4] ": The great BIG LIE &amp;amp; CRIME of our era.\n\n#EU #euistheproblem #LeaveEU #Brexit #Nexit #Frex
## [5] ": the impact of #Brexit is so appalling that #TheresaMay dare not tell us. STOP IT NOW https://t
## [6] ": We urge MPs not to write the Govt a blank cheque for hard #Brexit. Show your support &amp;amp; RT
```

```
cat("\nEliminamos puntuación\n")
```

```
##
## Eliminamos puntuación
```

```
texto_tweets = gsub('[[:punct:]]', '', texto_tweets)
head(texto_tweets)
```

```
## [1] "       Remoaners lost httpstco5NAK8ybIU6"
## [2] " These people who keep our NHS going are increasingly returning home Brexit will destroy the NHS
## [3] " Bank of England tells City to prepare contingencies for worstcase Brexit scenario httpstcoOLF7
## [4] " The great BIG LIE amp CRIME of our era\n\nEU euistheproblem LeaveEU Brexit Nexit Frexit Italexi
```

```
## [5] " the impact of Brexit is so appalling that TheresaMay dare not tell us STOP IT NOW httpstcocvIf
## [6] " We urge MPs not to write the Govt a blank cheque for hard Brexit Show your support amp RT https
```

```
cat("\nEliminamos números\n")
```

```
##
## Eliminamos números
```

```
texto_tweets = gsub('[[:digit:]]', '', texto_tweets)
head(texto_tweets)
```

```
## [1] "       Remoaners lost httpstcoNAKybIU"
## [2] " These people who keep our NHS going are increasingly returning home Brexit will destroy the NHS
## [3] " Bank of England tells City to prepare contingencies for worstcase Brexit scenario httpstcoOLFd
## [4] " The great BIG LIE amp CRIME of our era\n\nEU euistheproblem LeaveEU Brexit Nexit Frexit Italexi
## [5] " the impact of Brexit is so appalling that TheresaMay dare not tell us STOP IT NOW httpstcocvIf
## [6] " We urge MPs not to write the Govt a blank cheque for hard Brexit Show your support amp RT https
```

```
cat("\nEliminamos enlaces html\n")
```

```
##
## Eliminamos enlaces html
```

```
texto_tweets = gsub('http\\w+', '', texto_tweets)
head(texto_tweets)
```

```
## [1] "       Remoaners lost "
## [2] " These people who keep our NHS going are increasingly returning home Brexit will destroy the NHS
## [3] " Bank of England tells City to prepare contingencies for worstcase Brexit scenario  brexit arti
## [4] " The great BIG LIE amp CRIME of our era\n\nEU euistheproblem LeaveEU Brexit Nexit Frexit Italexi
## [5] " the impact of Brexit is so appalling that TheresaMay dare not tell us STOP IT NOW "
## [6] " We urge MPs not to write the Govt a blank cheque for hard Brexit Show your support amp RT  htt
```

```
cat("\nEliminamos espacios innecesarios\n")
```

```
##
## Eliminamos espacios innecesarios
```

```
texto_tweets = gsub('[ \t]{2,}', '', texto_tweets)
texto_tweets = gsub('^\\s+|\\s+$', '', texto_tweets)
head(texto_tweets)
```

```
## [1] "Remoaners lost"
## [2] "These people who keep our NHS going are increasingly returning home Brexit will destroy the NHS
## [3] "Bank of England tells City to prepare contingencies for worstcase Brexit scenariobrexit arti"
## [4] "The great BIG LIE amp CRIME of our era\n\nEU euistheproblem LeaveEU Brexit Nexit Frexit Italexi
## [5] "the impact of Brexit is so appalling that TheresaMay dare not tell us STOP IT NOW"
## [6] "We urge MPs not to write the Govt a blank cheque for hard Brexit Show your support amp RThtt"
```

```
#Función para eliminar posibles errores al pasar a minúscula
try.error = function(x){
  # creamos un missing value
  y = NA
  # tryCatch error
  try_error = tryCatch(tolower(x), error=function(e) e)
  # if not un error
  if (!inherits(try_error, 'error'))
    y = tolower(x)
  return(y)
```

```
}
cat("\nPasamos a minúsucla si no hay error\n")
```

```
##
## Pasamos a minúsucla si no hay error
```

```
texto_tweets = sapply(texto_tweets, try.error)
head(texto_tweets)
```

```
##                                                                                                    R
##                                                                                                  "re
##   These people who keep our NHS going are increasingly returning home Brexit will destroy the NHS No
## "these people who keep our nhs going are increasingly returning home brexit will destroy the nhs no w
##                       Bank of England tells City to prepare contingencies for worstcase Brexit scenar
##                     "bank of england tells city to prepare contingencies for worstcase brexit scenari
##   The great BIG LIE amp CRIME of our era\n\nEU euistheproblem LeaveEU Brexit Nexit Frexit Italexit Ma
## "the great big lie amp crime of our era\n\neu euistheproblem leaveeu brexit nexit frexit italexit mar
##                                  the impact of Brexit is so appalling that TheresaMay dare not tell u
##                                "the impact of brexit is so appalling that theresamay dare not tell us
##                       We urge MPs not to write the Govt a blank cheque for hard Brexit Show your supp
##                     "we urge mps not to write the govt a blank cheque for hard brexit show your suppo
```

```
cat("\nEliminamos NAs en el texto\n")
```

```
##
## Eliminamos NAs en el texto
```

```
texto_tweets = texto_tweets[!is.na(texto_tweets)]
names(texto_tweets) = NULL
head(texto_tweets)
```

```
## [1] "remoaners lost"
## [2] "these people who keep our nhs going are increasingly returning home brexit will destroy the nhs
## [3] "bank of england tells city to prepare contingencies for worstcase brexit scenariobrexit arti"
## [4] "the great big lie amp crime of our era\n\neu euistheproblem leaveeu brexit nexit frexit italexit
## [5] "the impact of brexit is so appalling that theresamay dare not tell us stop it now"
## [6] "we urge mps not to write the govt a blank cheque for hard brexit show your support amp rthtt"
```

Clasificador de sentimientos

Ahora vamos a clasificar por emociones y obtener la mejor de ellas

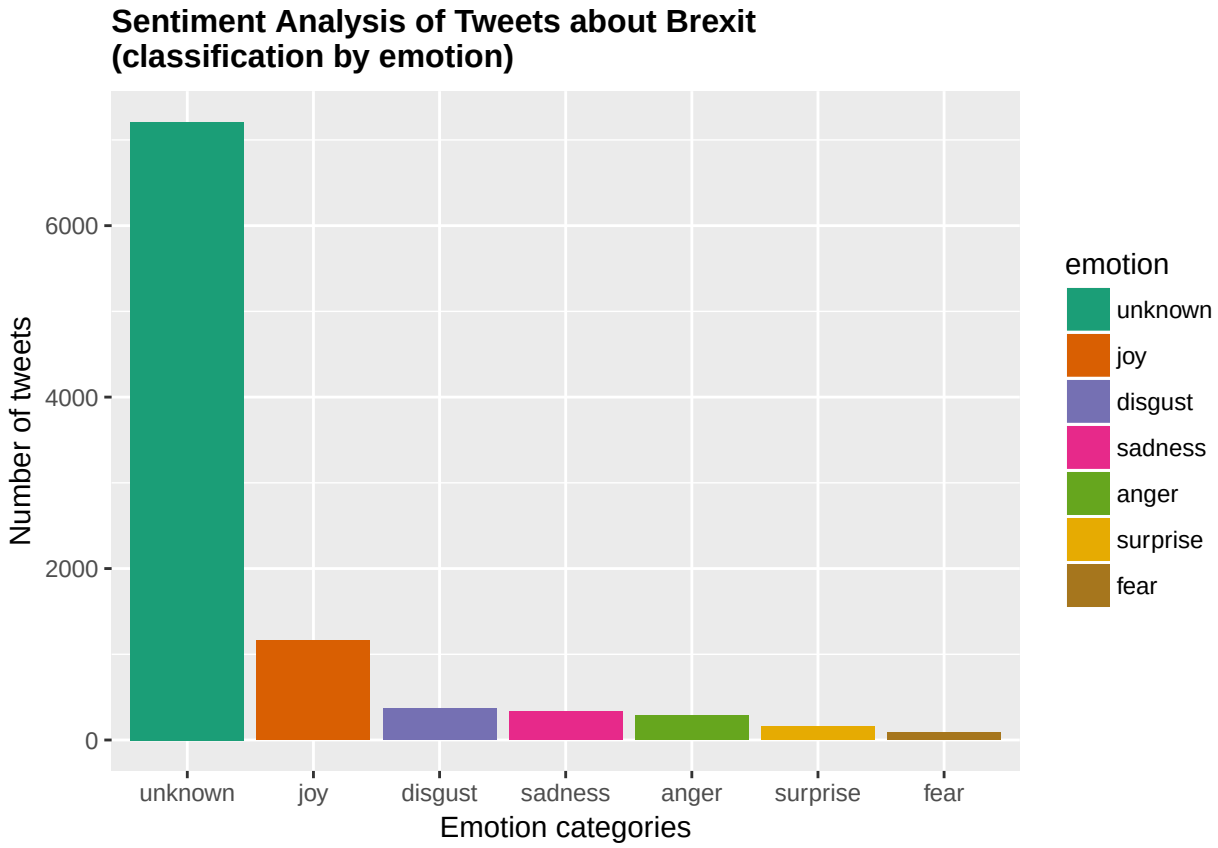
```
clasificacion_emociones = classify_emotion(texto_tweets, algorithm='bayes', prior=1.0)
emociones = clasificacion_emociones[,7]
# sustituimos NA's por 'unknown'
emociones[is.na(emociones)] = 'unknown'
```

Clasificamos por popularidad los tweets y montamos el dataframe

```
clasificacion_popularidad = classify_polarity(texto_tweets, algorithm='bayes')
popularidad = clasificacion_popularidad[,4]
data = data.frame(text=texto_tweets, emotion=emociones, polarity=popularidad, stringsAsFactors=FALSE)
data = within(data, emotion <- factor(emotion, levels=names(sort(table(emotion), decreasing=TRUE))))
```

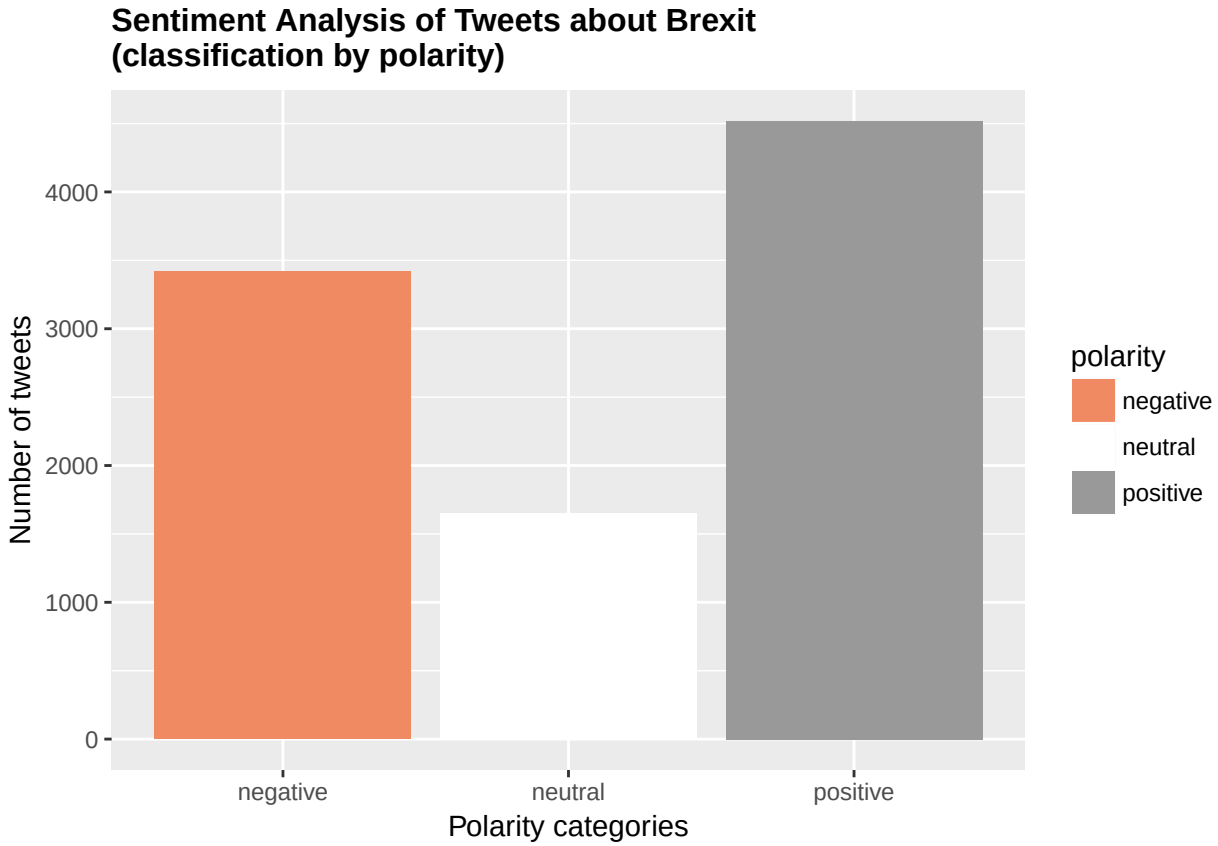
Vamos a mostrar una gráfica en función de la distribución de las emociones sacadas.

```
ggplot(data, aes(x=emotion)) +
geom_bar(aes(y=..count.., fill=emotion)) +
scale_fill_brewer(palette='Dark2') +
labs(x='Emotion categories', y='Number of tweets') +
ggtitle('Sentiment Analysis of Tweets about Brexit\n(classification by emotion)') +
theme(plot.title = element_text(size=12, face='bold'))
```



Vamos a mostrar una gráfica en función de la distribución de la popularidad.

```
ggplot(data, aes(x=polarity)) +
geom_bar(aes(y=..count.., fill=polarity)) +
scale_fill_brewer(palette='RdGy') +
labs(x='Polarity categories', y='Number of tweets') +
ggtitle('Sentiment Analysis of Tweets about Brexit\n(classification by polarity)') +
theme(plot.title = element_text(size=12, face='bold'))
```



Ahora vamos a crear una nube de palabras para ver que es lo que más se usa. Lo primero será separar el texto por emociones y visualizar las palabras

```
emociones_data = levels(factor(data$emotion))
tama_emociones_data = length(emociones_data)
documento_emociones = rep('', tama_emociones_data)
for (i in 1:tama_emociones_data)
{
tmp = texto_tweets[emociones == emociones_data[i]]
documento_emociones[i] = paste(tmp, collapse=' ')
}

# eliminamos palabras vacias como or,as,off...
documento_emociones = removeWords(documento_emociones, stopwords('english'))
# Creamos el corpus
corpus = Corpus(VectorSource(documento_emociones))
termdocumentmatrix = TermDocumentMatrix(corpus)
termdocumentmatrix = as.matrix(termdocumentmatrix)
colnames(termdocumentmatrix) = emociones_data

# comparison word cloud
comparison.cloud(termdocumentmatrix, colors = brewer.pal(tama_emociones_data, 'Dark2'),
scale = c(3,.5), random.order = FALSE, title.size = 1.5)
```


amp
sec
yet
brexiteers
maneu
brexittheconpartyproxywar
immigrants
may
culprit
reveals
strongly
joy
referendum
ruining
students
impact
dont
second
disgust
lik
favour
vote
speaking
thks
cons
localstudentspoll
union
many
won
caused
named
fed
wilbur
leeds
less
pic
remain
necessary
poorer
voted
theresa
tax
self
intolerant
freebia
pay
address
ross
free
please
appalling
though
iteu
nev
hide
must
sick
proud
tell
now
unknown
migration
turn
stuff
national
ukip
seeking
biaexit
dark
good
love
plotting
biafrans
sends
qmb
theresamay
dare
issue
blamed
biafrexit
leave
tweet
caine
even
gov
stop
enjoying
taken
michael
middleclass
master
get
land
kanu
way
gave
pressure
nation
britain
eliminate
take
sadness
rich
nhs
uks
brexit
hustings
current
richer
poor
panic
see
left
fears
day
rather
grip
bad
overthrow
mongers
doctor
brexitcost
make
servant
lets
like
sosh
fear
htt
frightening
project
means
cruel
europe
lies
truth
news
suffer
believe
hate
wonder
video
youll
islamic
hang
effects
cant
ivesaid
bro
terror
compilation
lose
nurses
nowhere
hold
back
canada
one
mayoral
despite
meanwhile
returningwill
ever
theatre
fear
million
shadow
lay
gibraltar
war
far
terrible
willing
high
youre
increasingly
keep
alert
stage
new
convenience
con
thing
manchester
myth
anger
destroy going
absolutely
home
jobs
street
deal
dreadful
dependent
groups
top
trump
thank
hence
craziness
flag
brexitover
fourth
isnt
migrants
stupid
surprise
bed
amid
really
eueeaefta
trade
wonderful
priority
tories
cause