Mini-Projeto 04 - Sentiment Analisys IV

Franklin Ferreira

14 de fevereiro, 2020

% !TEX encoding = UTF-8 Unicode

Mini-Projeto 04 - Sentiment Analisys (Análise de sentimentos) IV

O objetivo desta análise é explorar diferentes técnicas e ferramentas para a captura, manipulação e transformação de dados provenientes do Twitter. Buscaremos avaliar a frequência com que uma determinada palavra-chave é usada em uma região geográfica.

Esta técnica visa auxilar os tomadores de decisão na compreensão dos sentimentos do seu público alvo em relação a um determinado tema. Como por exemplo, determinar em quais cidades uma campanha de marketing foi mais comentada.

O projeto completo, bem como todos os arquivos auxiliares utilizados para sua criação podem ser encontrados no link do github ao final desta análise.

Importando bibliotecas necessárias

```
# Importando bibliotecas necessárias para o uso do rmarkdown.

# install.packages("knitr")
# install.packages("rmarkdown")

library(knitr)
library(rmarkdown)

## Pacotes para se conectar com o Twitter.

# install.packages("twitteR")
# install.packages("httr")

library(rtweet)
library(httr)

## Pacotes para Data Munging.

# install.packages("plyr")
# install.packages("dplyr")

library(plyr)
```

```
library(dplyr)

## Pacotes para a criação de gráficos.

# install.packages("ggplot2")

library(ggplot2)
```

Funções auxiliares

Antes de iniciar a análise, vamos definir algumas funções auxiliares para automatizar as tarefas de Data Munging de um Tweet.

```
####
## Definindo funções auxiliares.
####

# Função que realiza uma limpeza nos textos capturados de tweets.

cleanData <- function(tweet) {

  # Remove links http.

  tweet = gsub("(f|ht)(tp)(s?)(://)(.*)[.|/](.*)", " ", tweet)
  tweet = gsub("http\\w+", "", tweet)

  # Remove retweets.

  tweet = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", " ", tweet)

  # Remove "#Hashtag".

  tweet = gsub("#\\w+", " ", tweet)

  # Remove nomes de usuários "@people".

  tweet = gsub("@\\w+", " ", tweet)

  # Remove pontuação.

  tweet = gsub("[[:punct:]]", " ", tweet)

  # Remove números.

  tweet = gsub("[[:digit:]]", " ", tweet)

  # Remove espaços desnecessários.

  tweet = gsub("[ \t]{2,}", " ", tweet)

  tweet = gsub("^\\s+|\\s+$", "", tweet)
```

```
  # Convertendo encoding de caracteres e letras maíusculas em minúsculas.

  tweet = stringi::stri_trans_general(tweet, "latin-ascii")

  tweet = tryTolower(tweet)

  tweet = tweet[!is.na(tweet)]
}

# Converte caracateres maiúsculos para minúsculos.

tryTolower = function(x) {

  # Cria um dado missing (NA).

  y = NA

  # Executa um tramento de erro caso ocorra.

  try_error = tryCatch(tolower(x), error = function(e) e)

  # Se não houver erro, converte os caracteres.

  if (!inherits(try_error, "error"))
    y = tolower(x)

  return(y)
}
```

Executando a autenticação para se conectar com o Twitter

Utiliza-se o pacote *rtweet* para estabelecer uma conexão com o Twitter. Note que ao efetuar o acesso, é necessário que se tenha uma conta nesta rede social e que possua as chaves de autenticação solicitadas para o estabelicimento da conexão. Caso não tenha as chaves, poderá obtê-las aqui: https://apps.twitter.com/.

```
# Definindo as chaves de autenticação no Twitter.

key         <- "Insert your key here!"
secret      <- "Insert your secret here!"
token       <- "Insert your token here!"
tokenSecret <- "Insert your token secret here!"

# Realizando o processo de autenticação para iniciar uma sesssão com o rtweet.

token <- create_token (
  consumer_key    = key,
  consumer_secret = secret,
  access_token    = token,
  access_secret   = tokenSecret
)
```

Explorando as funções de captura de Tweets do pacote rtweet

O pacote *rtweet* permite a busca por tweets dentro de uma timeline específica.

```
# Definindo o nome da timeline a ser analisada.

timeLine <- "dsacademybr"

# Definindo o número de tweets a serem capturados.

n <- 100

# Capturando Tweets.

tlTweets <- get_timelines(timeLine, n = n)
```

Esta biblioteca também oferece funções para a captura do stream de tweets por determinado período de tempo.

```
# Definindo a key word a ser utilizada para filtrar os Tweets que devem ser capturados.

keyWord <- ''

# Capturando por um período de tempo (o padrão é 30 segundos), tweets aleatórios.

randomTweets <- stream_tweets(keyWord)
```

```
# Definindo a key word a ser utilizada para filtrar os Tweets que devem ser capturados.

keyWord <- 'dataScience'

# Capturando por um período de tempo (o padrão é 30 segundos), tweets que contenham a
# keyWord especificada.

kwTweets <- stream_tweets(keyWord)
```

Outra maneira de se obter os dados é a partir da captura das tendências dos Tweets de uma determinada região.

```
# Defindo a região da qual as tendências serão capturadas.

place <- "Brazil"

# Capturando as têndencias em um determinada região.

trendsTweets <- get_trends(place)

# Exibindo os primeiros tweets capturados.

trendsTweets[1:5, 'trend']
```

```
## # A tibble: 5 x 1
```

```
##   trend
##   <chr>
## 1 #YayaNoBotafogo
## 2 #RaveDeFavelaClipe
## 3 #dancela
## 4 Até a Bianca
## 5 #edecasa
```

Caso o número de Tweets necessários exceda o limite de 18.000, podemos configurar o comando *retryonrate-limit* como TRUE para que o processo de captura aguarde o limite de mensagens por período de tempo se renovar e os dados voltem a ser obtidos até que a quantidade solicitada seja alcançada.

```
# Definindo a key word a ser utilizada para filtrar os Tweets que devem ser capturados.

keyWord <- 'DataScience'

# Definindo o número de tweets a serem capturados.

n <- 20000

# Capturando 20.000 de tweets que contenham a key word especificada.

dsTweets <- search_tweets(keyWord, n = n, retryonratelimit = TRUE)
```

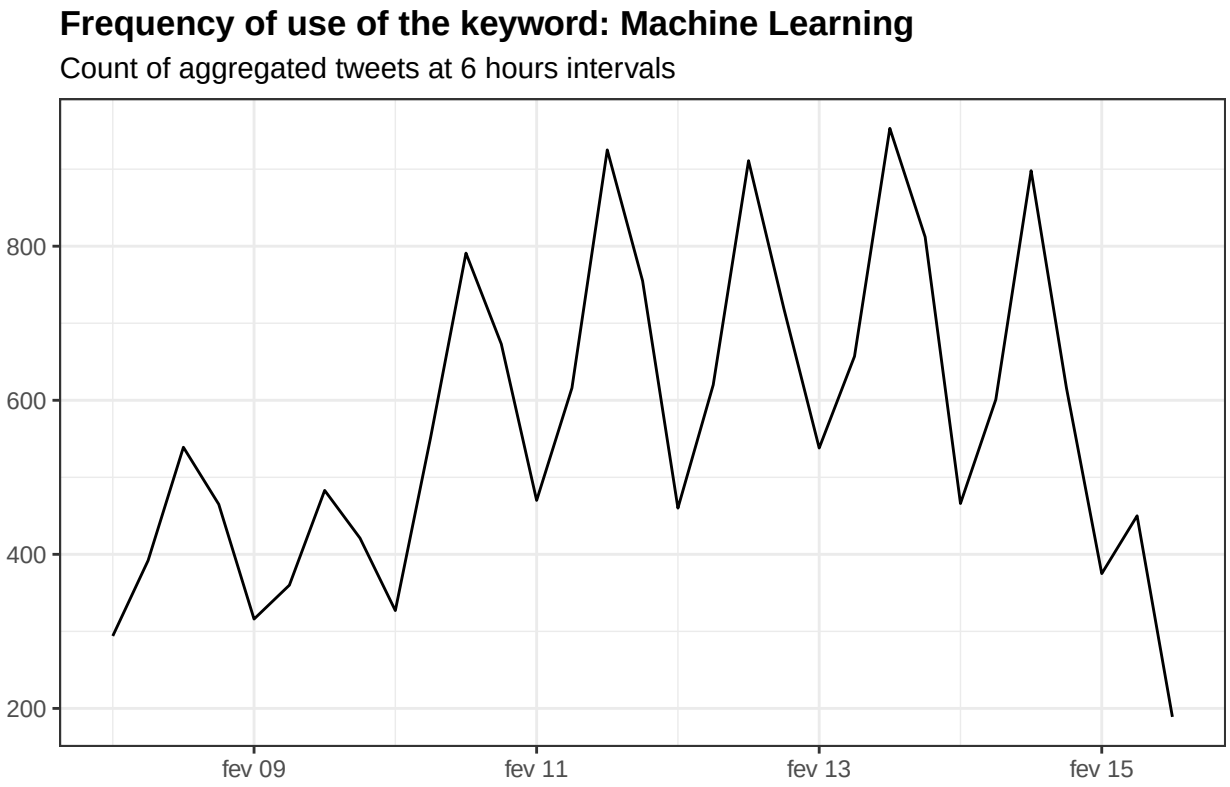
Série temporal sobre a frequência de uso de uma palavra-chave

O objetivo nesta etapa é avaliar o comportamento do uso de uma palavra-chave ao longo do tempo.

```
# Definindo a key word a ser utilizada para filtrar os Tweets que devem ser capturados.

keyWord <- "Machine Learning"

# Definindo o número de tweets a serem capturados.

n <- 10000

# Capturando mensagens no fluxo de tweets que contenham a palavra-chave especificada.

mlTweets <- search_tweets(keyWord, n = n , include_rts = FALSE, retryonratelimit = TRUE)
```

```
# Definindo o intervalo de tempo com o qual os dados na série temporal devem ser exibidos.

tsTime <- "6 hours"

# Plotando o gráfico da série temporal.

ts_plot(mlTweets, tsTime) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold")) +
  xlab(NULL) +
  ylab(NULL) +
  labs (
```

```
    title    = paste('Frequency of use of the keyword:', keyWord),
    subtitle = paste("Count of aggregated tweets at", tsTime, "intervals"),
    caption  = "\nSource: Data collected from Twitter with the rtweet package"
  )
```



O gráfico exibe que a frequência de uso da palavra-chave *Machine Learning* cresce nas primeiras 12h do dia, alcança sua frequência máxima por volta das 12h e então passa a decair.

Série temporal sobre a frequência de uso de uma palavra-chave em diferentes regiões

O objetivo nesta etapa é avaliar o comportamento do uso de uma palavra-chave ao longo do tempo em diferentes regiões.

```
##
### Capturando tweets que contenham a keyword especificada durante os últimos.
##

# Definindo a keyWord.

keyWord <- "Big Data"

# Definindo o número máximo de tweets que podem ser capturados.
```

```
n <- 5000

# Capturando tweets em diferentes regiões.

mlTweetsInRJ <- search_tweets(keyWord, geocode = lookup_coords("rio de janeiro"),
                              n = n, include_rts = FALSE, retryonratelimit = TRUE)
```

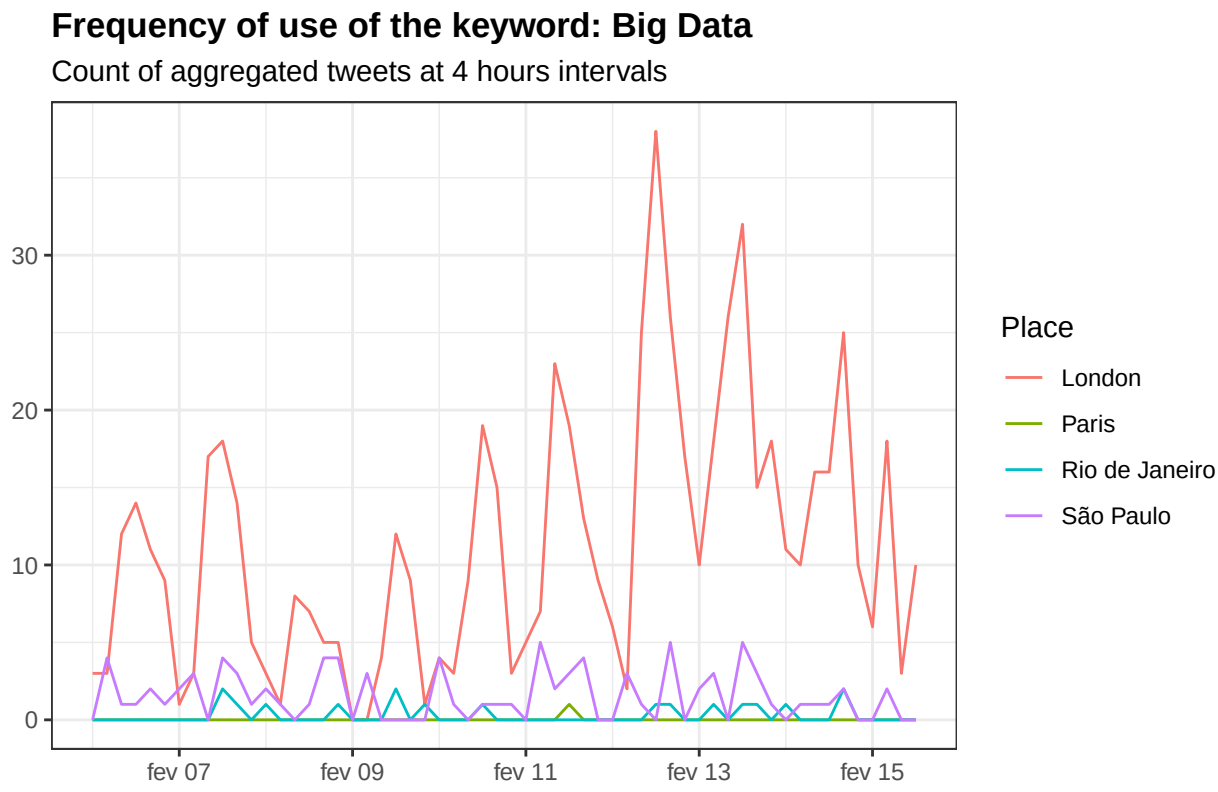
```
## retry on rate limit...
## waiting about 13 minutes...
```

```
mlTweetsInSP <- search_tweets(keyWord, geocode = lookup_coords("são paulo"),
                              n = n, include_rts = FALSE, retryonratelimit = TRUE)
mlTweetsInLD <- search_tweets(keyWord, geocode = lookup_coords("london"),
                              n = n, include_rts = FALSE, retryonratelimit = TRUE)
mlTweetsInPA <- search_tweets(keyWord, geocode = lookup_coords("paris"),
                              n = n, include_rts = FALSE, retryonratelimit = TRUE)
```

Vamos organizar todos os tweets capturados em um único dataset para efetuar a plotagem do gráfico.

```
# Criando um dataset com todos os tweets capturados.

dataTweets <- rbind(mlTweetsInRJ, mlTweetsInSP, mlTweetsInLD, mlTweetsInPA)

# Contabilizando o número de tweets capturados para cada estado.

nTweets <- c(nrow(mlTweetsInRJ), nrow(mlTweetsInSP),
             nrow(mlTweetsInLD), nrow(mlTweetsInPA))

# Atribuindo o nome do estado a qual cada tweet pertence dentro do dataset criado.

dataTweets$place_name <- rep(c("Rio de Janeiro", "São Paulo", "London", 'Paris'), nTweets)
```

```
# Definindo o intervalo de tempo com o qual os dados na série temporal devem ser exibidos.

tsTime <- "4 hours"

# Plotando gráfico de série temporal para exibir a frequência com que a keyWord foi
# buscada em cada estado pesquisado.

ts_plot(group_by(dataTweets, place_name), tsTime) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold")) +
  xlab(NULL) +
  ylab(NULL) +
  labs (
    title    = paste('Frequency of use of the keyword:', keyWord),
    color    = 'Place',
    subtitle = paste("Count of aggregated tweets at", tsTime, "intervals"),
    caption  = "\nSource: Data collected from Twitter with the rtweet package"
  )
```



Podemos visualizar que Londres apresenta as maiores frequências do uso da palavra-chave *Big Data* quando comparado com as demais regiões analisadas.

Contato

- **E-mail:** franklinfs390@gmail.com
- **Linkedin:** https://www.linkedin.com/in/franklinfs390/
- **Github:** https://github.com/franklin390