% !TEX encoding = UTF-8 Unicode

Mini-Project 1 - Sentiment Analysis in Social Networks

The objective is to capture data from the social network Twitter and perform sentiment analysis with the captured data. In order for this project to run, several packages must be installed and loaded.

The entire project will be described according to its stages. First we will use the sentiment score calculation and then we will use a classifier with the Naive Bayes algorithm.

```
# install.packages("twitteR")
# install.packages("httr")
# install.packages("knitr")
# install.packages("rmarkdown")
library(twitteR)

## Warning: package 'twitteR' was built under R version 4.1.3

library(httr)

## Warning: package 'httr' was built under R version 4.1.3

library(knitr)
library(rmarkdown)

# Carregando a biblioteca com funções de limpeza
source('utils.R')
options(warn=-1)
```

Step 1 - Authentication

Authentication process.

```
# Creating Twitter authentication
key <- "EkZGfFA14vVloolZqlJfWbklv"
secret <- "jWXFDQREeBvaguVDkHzZ9hQvfm6KJKSB9fOCEXcRnJ1cbmaQ8L"
token <- "1344471742778060806-ydgX37AqF2OvGsun5XbyYgEZP9NCVA"
tokensecret <- "VDjtN6K59RyeB9uBct86A9vjBb9oJO1Nh40qnTiSKygDR"

# Authentication. Answer 1 (Yes) when asked about using direct
connection.
setup_twitter_oauth(key, secret, token, tokensecret)

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

Step 2 - Connection

Here we will test the connection and capture the tweets. We will search for tweets referencing the hashtag #BigData.

```r
# Checking the user's timeline
userTimeline("AndreKi62981839")
```

```
## [[1]]
## [1] "AndreKi62981839: andre.kim1@hotmail.com"
```

```r
# Capturing the tweets
tema <- "BigData"
qtd_tweets <- 1000
lingua <- "pt"
tweetdata = searchTwitter(tema, n = qtd_tweets, lang = lingua)

# Viewing the first lines of the tweetdata object
head(tweetdata)
```

```
## [[1]]
## [1] "mecaengordos: #Cyberpunk gnrrs!\n#bigdata #DiaDeLaMadre
https://t.co/7g2RSOsocu"
##
## [[2]]
## [1] "Bot_Corona_V: RT @oscarhbp1: Doxa 1100s
...\nhttps://t.co/4jAtlWo0el\nProxima meta Doxa 1400 y 200 K\n#doxa
#liderazgo #marketing #asesoría #economía #socio…"
##
## [[3]]
## [1] "oscarhbp1: Doxa 1100s ...\nhttps://t.co/4jAtlWo0el\nProxima meta
Doxa 1400 y 200 K\n#doxa #liderazgo #marketing #asesoría…
https://t.co/97dA7SRhMq"
##
## [[4]]
## [1] "veradatabr: Saiba como utilizar o Big data no varejo e tirar
proveito desta tecnologia em seu negócio.\n#ArtificialIntelligence…
https://t.co/NwG4RuJFMc"
##
## [[5]]
## [1] "JohnM19018119: RT @gp_pulipaka: OpenVINO\231 for AI Inference.
#BigData #Analytics #DataScience #AI #MachineLearning #IoT #IIoT #PyTorch
#Python #RStats #Ten…"
##
## [[6]]
## [1] "SanjayMartolia: RT @gp_pulipaka: OpenVINO\231 for AI Inference.
#BigData #Analytics #DataScience #AI #MachineLearning #IoT #IIoT #PyTorch
#Python #RStats #Ten…"
```

Step 3 - Treatment of data collected through text mining

Here we will install the tm package, for text mining. We are going to convert the collected tweets into a Corpus-like object, which stores data and metadata, and then we will do some cleaning process, such as removing punctuation, converting the data

to lowercase letters and removing stopwords (common words in the English language, in this case) .

```
# Installing the package for Text Mining.
# install.packages("tm")
# install.packages("SnowballC")
library(SnowballC)
library(tm)

## Carregando pacotes exigidos: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:httr':
##
##     content

# Treatment (cleaning, organization and transformation) of collected data
tweetlist <- sapply(tweetdata, function(x) x$getText())
tweetlist <- iconv(tweetlist, to = "utf-8", sub="")
tweetlist <- limpaTweets(tweetlist)
tweetcorpus <- Corpus(VectorSource(tweetlist))
tweetcorpus <- tm_map(tweetcorpus, removePunctuation)
tweetcorpus <- tm_map(tweetcorpus, content_transformer(tolower))
tweetcorpus <- tm_map(tweetcorpus, function(x)removeWords(x,
stopwords()))

# Converting the Corpus object to plain text
termo_por_documento = as.matrix(TermDocumentMatrix(tweetcorpus), control
= list(stopwords = c(stopwords("portuguese"))))
```

Step 4 - Wordcloud, association between words and dendrogram

Let's create a word cloud (wordcloud) to check the relationship between the words that occur most frequently. We create a table with the frequency of the words and then we generate a dendrogram, which shows how the words relate to and associate with the main theme (in our case, the term BigData).

```
# Installing the wordcloud package
# install.packages("RColorBrewer")
# install.packages("wordcloud")
library(RColorBrewer)
library(wordcloud)

# Generating a word cloud
pal2 <- brewer.pal(8,"Dark2")

wordcloud(tweetcorpus,
          min.freq = 2,
          scale = c(5,1),
```

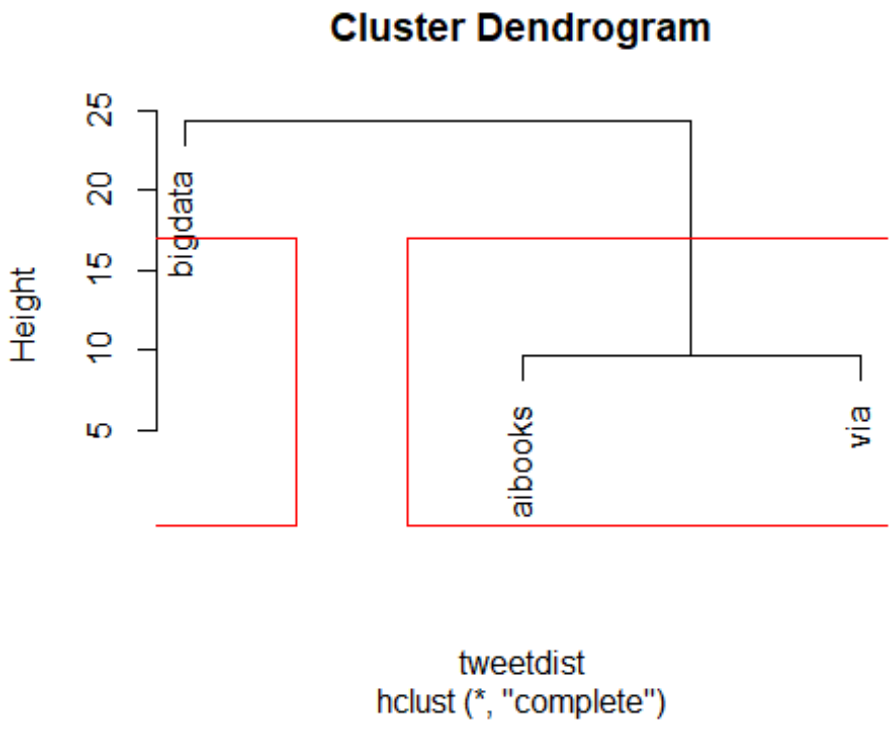
```
          random.color = F,
          max.word = 60,
          random.order = F,
          colors = pal2)
```


```
# Converting the text object to array format
tweettdm <- TermDocumentMatrix(tweetcorpus)
tweettdm

## <<TermDocumentMatrix (terms: 294, documents: 205)>>
## Non-/sparse entries: 756/59514
## Sparsity           : 99%
## Maximal term length: 15
## Weighting          : term frequency (tf)

# Finding the words that appear most often
findFreqTerms(tweettdm, lowfreq = 11)

##  [1] "aibooks"     "django"      "via"         "afa"         "com"
##  [6] "que"         "para"        "azure"       "bigdata"     "dados"
## [11] "engenharia"  "fundamentos" "nafa"        "safa"        "uma"
## [16] "querem"      "proa"        "twitter"

# Searching for associations
findAssocs(tweettdm, 'datascience', 0.60)

## $datascience
## numeric(0)
```

```r
# Removing sparse terms (not often used)
tweet2tdm <-removeSparseTerms(tweettdm, sparse = 0.9)

# scaling the data
tweet2tdmscale <- scale(tweet2tdm)

# Distance Matrix
tweetdist <- dist(tweet2tdmscale, method = "euclidean")

# Preparing the dendrogram
tweetfit <- hclust(tweetdist)

# Creating the dendrogram (checking how the words are grouped together)
plot(tweetfit)

# Checking the groups
cutree(tweetfit, k = 2)
```

```
## aibooks     via bigdata
##       1       1       2
```

```r
# Visualizing the word groups in the dendrogram
rect.hclust(tweetfit, k = 2, border = "red")
```



Step 5 - Sentiment Analysis

Now we can proceed with sentiment analysis. We built a function (called sentiment.score) and a list of positive and negative words (these lists are part of this project). Our function checks each item in the dataset and compares it to the given word lists and from there calculates the sentiment score, whether positive, negative or neutral.

```
# Creating a function to evaluate sentiment
# install.packages("stringr")
# install.packages("plyr")
library(stringr)
library(plyr)

##
## Attaching package: 'plyr'

## The following object is masked from 'package:twitteR':
##
##     id

sentimento.score = function(sentences, pos.words, neg.words, .progress =
'none')
{

  # Creating an array of scores with lapply
  scores = laply(sentences,
                 function(sentence, pos.words, neg.words)
                 {
                   sentence = gsub("[[:punct:]]", "", sentence)
                   sentence = gsub("[[:cntrl:]]", "", sentence)
                   sentence =gsub('\\d+', '', sentence)
                   tryTolower = function(x)
                   {
                     y = NA

                     # Tratamento de Erro
                     try_error = tryCatch(tolower(x), error=function(e)
e)
                     if (!inherits(try_error, "error"))
                       y = tolower(x)
                     return(y)
                   }

                   sentence = sapply(sentence, tryTolower)
                   word.list = str_split(sentence, "\\s+")
                   words = unlist(word.list)
                   pos.matches = match(words, pos.words)
                   neg.matches = match(words, neg.words)
                   pos.matches = !is.na(pos.matches)
```

```
                    neg.matches = !is.na(neg.matches)
                    score = sum(pos.matches) - sum(neg.matches)
                    return(score)
                  }, pos.words, neg.words, .progress = .progress )

  scores.df = data.frame(text = sentences, score = scores)
  return(scores.df)
}

# Mapping the positive and negative words
pos = readLines("palavras_positivas.txt")
neg = readLines("palavras_negativas.txt")

# Creating mass data for testing
teste = c("Big Data is the future", "awesome experience",
          "analytics could not be bad", "learn to use big data")

# Testing the function on our dummy data mass
testesentimento = sentimento.score(teste, pos, neg)
class(testesentimento)

## [1] "data.frame"

# Checking the score
#0 - expression has no word in our positive and negative word lists or
# found a negative and a positive word in the same sentence
# 1 - expression has a word with a positive connotation
# -1 - expression has a word with a negative connotation
testesentimento$score

## [1]  0  1 -1  0
```

Step 6 - Generating Sentiment Analysis Score

With the score calculated, let's break it down by country, in this case Canada and the US, as a way of comparing sentiment in different regions. We then generate a boxplot and a histogram using the lattice package.

```
# Tweets by country
catweets = searchTwitter("ca", n = 500, lang = "en")
usatweets = searchTwitter("usa", n = 500, lang = "en")

# Getting text
catxt = sapply(catweets, function(x) x$getText())
usatxt = sapply(usatweets, function(x) x$getText())

# Vector of country tweets
paisTweet = c(length(catxt), length(usatxt))

# Joining the texts
```

```r
paises = c(catxt, usatxt)

# Applying function to calculate sentiment score
scores = sentimento.score(paises, pos, neg, .progress = 'text')
```

```
##   |
|                                                                      |
0%  |
|                                                                      |
1%  |
|=                                                                     |
1%  |
|=                                                                     |
2%  |
|==                                                                    |
2%  |
|==                                                                    |
3%  |
|==                                                                    |
4%  |
|===                                                                   |
4%  |
|===                                                                   |
5%  |
|====                                                                  |
5%  |
|====                                                                  |
6%  |
|=====                                                                 |
6%  |
|=====                                                                 |
7%  |
|=====                                                                 |
8%  |
|======                                                                |
8%  |
|======                                                                |
9%  |
|=======                                                               |
9%  |
|=======                                                               |
10%  |
|=======                                                               |
11%  |
|========                                                              |
11%  |
|========                                                              |
12%  |
|=========                                                             |
12%  |
```

```
|=========                                                              |
13%  |
|=========                                                              |
14%  |
|==========                                                             |
14%  |
|==========                                                             |
15%  |
|===========                                                            |
15%  |
|===========                                                            |
16%  |
|============                                                           |
16%  |
|============                                                           |
17%  |
|============                                                           |
18%  |
|=============                                                          |
18%  |
|=============                                                          |
19%  |
|==============                                                         |
19%  |
|==============                                                         |
20%  |
|==============                                                         |
21%  |
|===============                                                        |
21%  |
|===============                                                        |
22%  |
|================                                                       |
22%  |
|================                                                       |
23%  |
|================                                                       |
24%  |
|=================                                                      |
24%  |
|=================                                                      |
25%  |
|==================                                                     |
25%  |
|==================                                                     |
26%  |
|===================                                                    |
26%  |
|===================                                                    |
27%  |
```

```
|==================                                                    |
28%  |
|====================                                                  |
28%  |
|====================                                                  |
29%  |
|======================                                                |
29%  |
|======================                                                |
30%  |
|======================                                                |
31%  |
|======================                                                |
31%  |
|======================                                                |
32%  |
|=======================                                               |
32%  |
|=======================                                               |
33%  |
|=======================                                               |
34%  |
|========================                                              |
34%  |
|========================                                              |
35%  |
|=========================                                             |
35%  |
|=========================                                             |
36%  |
|=========================                                             |
36%  |
|=========================                                             |
37%  |
|=========================                                             |
38%  |
|==========================                                            |
38%  |
|==========================                                            |
39%  |
|===========================                                           |
39%  |
|===========================                                           |
40%  |
|===========================                                           |
41%  |
|============================                                          |
41%  |
|============================                                          |
42%  |
```

```
|============================                                          |
42%  |
|============================                                          |
43%  |
|=============================                                         |
44%  |
|=============================                                         |
44%  |
|==============================                                        |
45%  |
|==============================                                        |
45%  |
|===============================                                       |
46%  |
|===============================                                       |
46%  |
|================================                                      |
47%  |
|================================                                      |
48%  |
|=================================                                     |
48%  |
|=================================                                     |
49%  |
|==================================                                    |
49%  |
|==================================                                    |
50%  |
|===================================                                   |
51%  |
|===================================                                   |
51%  |
|====================================                                  |
52%  |
|====================================                                  |
52%  |
|=====================================                                 |
53%  |
|=====================================                                 |
54%  |
|======================================                                |
54%  |
|======================================                                |
55%  |
|=======================================                               |
55%  |
|=======================================                               |
56%  |
|========================================                              |
56%  |
```

```
|=======================================                                |
57%  |
|========================================                               |
58%  |
|========================================                               |
58%  |
|=========================================                              |
59%  |
|=========================================                              |
59%  |
|=========================================                              |
60%  |
|==========================================                             |
61%  |
|==========================================                             |
61%  |
|==========================================                             |
62%  |
|==========================================                             |
62%  |
|==========================================                             |
63%  |
|==========================================                             |
64%  |
|==========================================                             |
64%  |
|==========================================                             |
65%  |
|==========================================                             |
65%  |
|==========================================                             |
66%  |
|==========================================                             |
66%  |
|==========================================                             |
67%  |
|==========================================                             |
68%  |
|==========================================                             |
68%  |
|==========================================                             |
69%  |
|==========================================                             |
69%  |
|==========================================                             |
70%  |
|==========================================                             |
71%  |
|==========================================                             |
71%  |
```

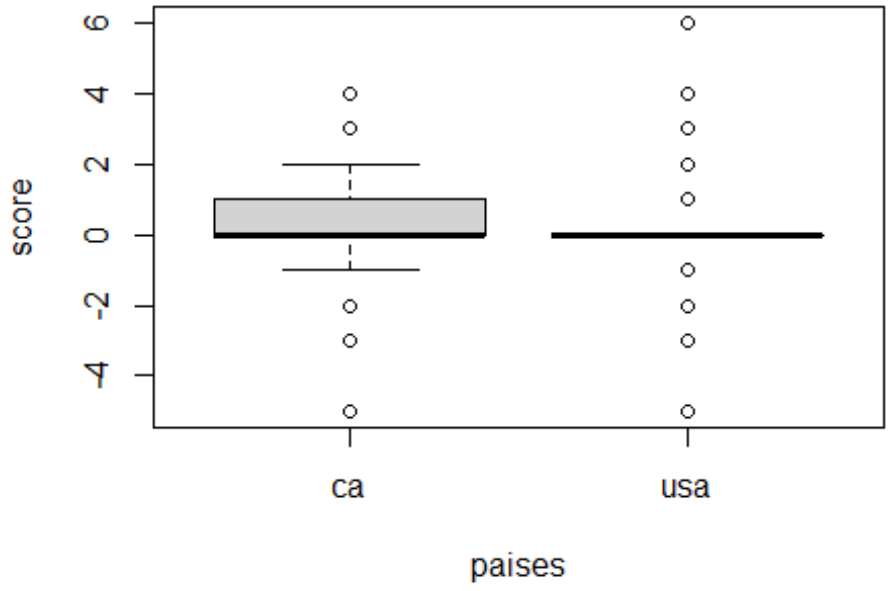
```
|=============================================                         |
72%  |
|==============================================                        |
72%  |
|==============================================                        |
73%  |
|==============================================                        |
74%  |
|==============================================                        |
74%  |
|==============================================                        |
75%  |
|==============================================                        |
75%  |
|==============================================                        |
76%  |
|==============================================                        |
76%  |
|==============================================                        |
77%  |
|==============================================                        |
78%  |
|==============================================                        |
78%  |
|==============================================                        |
79%  |
|==============================================                        |
79%  |
|==============================================                        |
80%  |
|==============================================                        |
81%  |
|==============================================                        |
81%  |
|==============================================                        |
82%  |
|==============================================                        |
82%  |
|==============================================                        |
83%  |
|==============================================                        |
84%  |
|==============================================                        |
84%  |
|==============================================                        |
85%  |
|==============================================                        |
85%  |
|==============================================                        |
86%  |
```

```
|=============================================================          |
86%  |
|==============================================================         |
87%  |
|===============================================================        |
88%  |
|================================================================       |
88%  |
|=================================================================      |
89%  |
|==================================================================     |
89%  |
|===================================================================    |
90%  |
|====================================================================   |
91%  |
|=====================================================================  |
91%  |
|====================================================================== |
92%  |
|=======================================================================|
92%  |
|=======================================================================|
93%  |
|=======================================================================|
94%  |
|=======================================================================|
94%  |
|=======================================================================|
95%  |
|=======================================================================|
95%  |
|=======================================================================|
96%  |
|=======================================================================|
96%  |
|=======================================================================|
97%  |
|=======================================================================|
98%  |
|=======================================================================|
98%  |
|=======================================================================|
99%  |
|=======================================================================|
99%  |
|=======================================================================|
100%
```

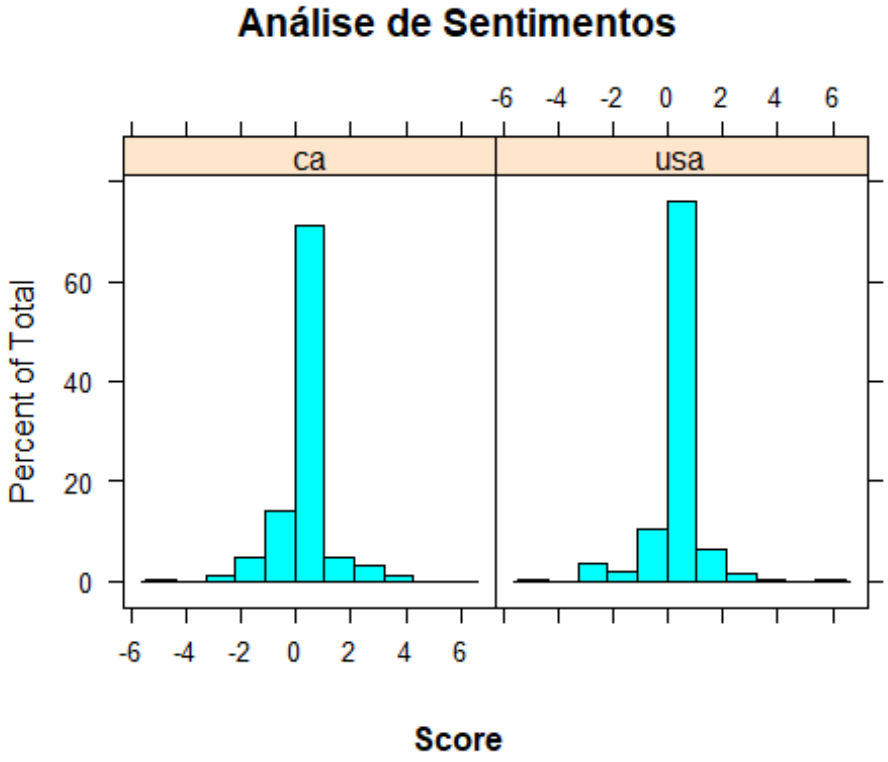
```r
# Calculating the score by country
scores$paises = factor(rep(c("ca", "usa"), paisTweet))
scores$muito.pos = as.numeric(scores$score >= 1)
scores$muito.neg = as.numeric(scores$score <= -1)

# Calculating the total
numpos = sum(scores$muito.pos)
numneg = sum(scores$muito.neg)

# Score global
global_score = round( 100 * numpos / (numpos + numneg) )
head(scores)
```

```
##
text
## 1                                          [IE] San Bernardino I210 E Wo /
Highland Ave Arden **Trfc Collision-1141 Enrt** https://t.co/5I7RBcbWQc
## 2                                      #Bawa: #EFCC  investigating source
of money used in buying party nomination forms  https://t.co/cwmp1Htwvs
## 3
They say he wrote this for your moms  https://t.co/7CfTAdJDqQ
## 4 RT @DavidKravinchuk: Make my mom proud this Mother's Day!  Register
for #WCFC22 now at https://t.co/uiy6I6YjZ2 and as Betty's little treat,…
## 5                                                                CA Lawmakers
Use 'Gut-And-Amend' To Resurrect Failed Gun Tax https://t.co/1xjsamFvgq
## 6
SNEAK PEEK : "Better Call Saul: Black and Blue" https://t.co/evrhy0HocX
##   score paises muito.pos muito.neg
## 1     0     ca         0         0
## 2     0     ca         0         0
## 3     0     ca         0         0
## 4     1     ca         1         0
## 5    -1     ca         0         1
## 6     0     ca         0         0
```

```r
boxplot(score ~ paises, data = scores)

# Generating a histogram with lattice
# install.packages("lattice")
library("lattice")
```




```
histogram(data = scores, ~score|paises, main = "Análise de Sentimentos",
xlab = "", sub = "Score")
```



Extra

Using Naive Bayes Classifier for sentiment analysis

Here we will do the sentiment analysis in a similar way as seen before, but using the sentiment package. This package has been discontinued from CRAN as it will no longer be updated, but can still be obtained via the CRAN archive link. The packages are available together with the project files and the installation procedure is described below.

```
# install.packages("/opt/DSA/Projetos/Projeto01/Rstem_0.4-1.tar.gz",
repos = NULL, type = "source")
# install.packages("/opt/DSA/Projetos/Projeto01/sentiment_0.2.tar.gz",
repos = NULL, type = "source")
# install.packages("ggplot2")
library(Rstem)

##
## Attaching package: 'Rstem'

## The following objects are masked from 'package:SnowballC':
##
##     getStemLanguages, wordStem

library(sentiment)
library(ggplot2)

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate
```

Collecting Tweets

The collection of tweets is done using the searchTwitter() function of the twittereR package.

```
# Collecting the tweets
tweetpt = searchTwitter("bigdata", n = 1500, lang = "pt")

# Getting the text
tweetpt = sapply(tweetpt, function(x) x$getText())
```

Cleaning, Organizing and Transforming Data

Here regular expressions, through the gsub() function to remove characters that can get in the way of the parsing process.

```
# Removing special characters
tweetpt = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", tweetpt)
# Removing @
tweetpt = gsub("@\\w+", "", tweetpt)
# Removing punctuation
tweetpt = gsub("[[:punct:]]", "", tweetpt)
# Removing digits
tweetpt = gsub("[[:digit:]]", "", tweetpt)
# Removing html links
tweetpt = gsub("http\\w+", "", tweetpt)
# Removing unnecessary spaces
tweetpt = gsub("[ \t]{2,}", "", tweetpt)
tweetpt = gsub("^\\s+|\\s+$", "", tweetpt)

# Creating function for lowerer
try.error = function(x)
{
  # Creating missing value
  y = NA
  try_error = tryCatch(tolower(x), error=function(e) e)
  if (!inherits(try_error, "error"))
    y = tolower(x)
  return(y)
}

# Lower case
tweetpt = sapply(tweetpt, try.error)

# Removing the NAs
tweetpt = tweetpt[!is.na(tweetpt)]
names(tweetpt) = NULL
```

Naive Bayes Classifier

We use the classify_emotion() and classify_polarity() functions from the sentiment package, which use the Naive Bayes algorithm for sentiment analysis. In this case, the algorithm itself sorts the words and we don't need to create lists of positive and negative words.

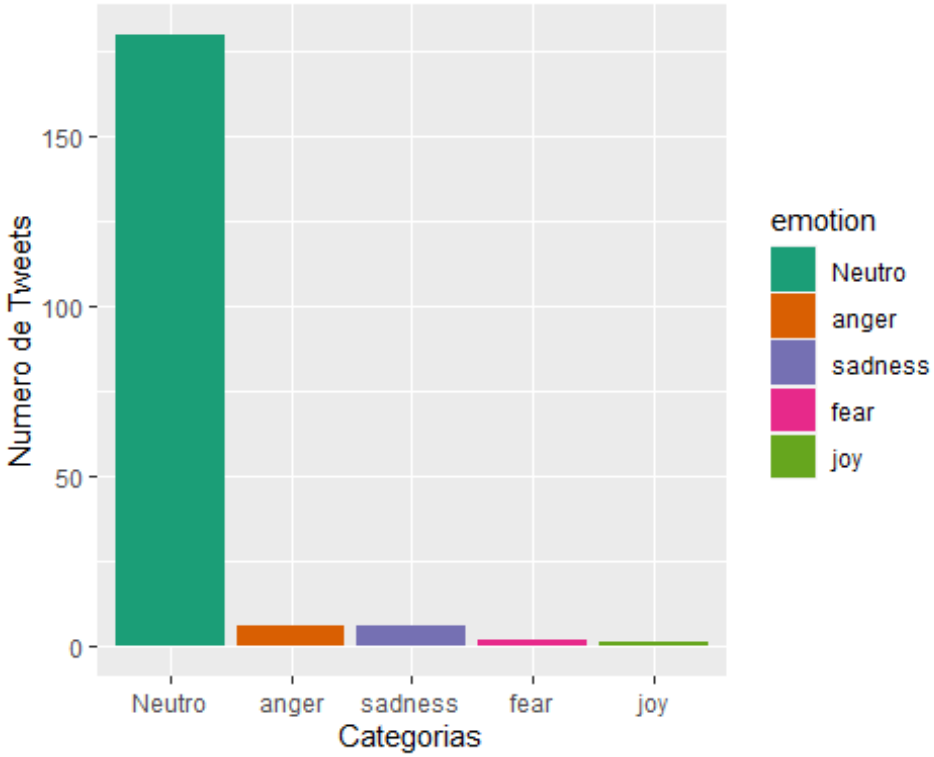
```
# Sorting emotion
class_emo = classify_emotion(tweetpt, algorithm = "bayes", prior = 1.0)
emotion = class_emo[,7]

# Replacing NA's with "Neutral"
emotion[is.na(emotion)] = "Neutro"

# Sorting polarity
class_pol = classify_polarity(tweetpt, algorithm = "bayes")
polarity = class_pol[,4]
```

```
# Generating a dataframe with the result
sent_df = data.frame(text = tweetpt, emotion = emotion,
                     polarity = polarity, stringsAsFactors = FALSE)

# Sorting the dataframe
sent_df = within(sent_df,
                 emotion <- factor(emotion, levels =
names(sort(table(emotion),

decreasing=TRUE))))
```

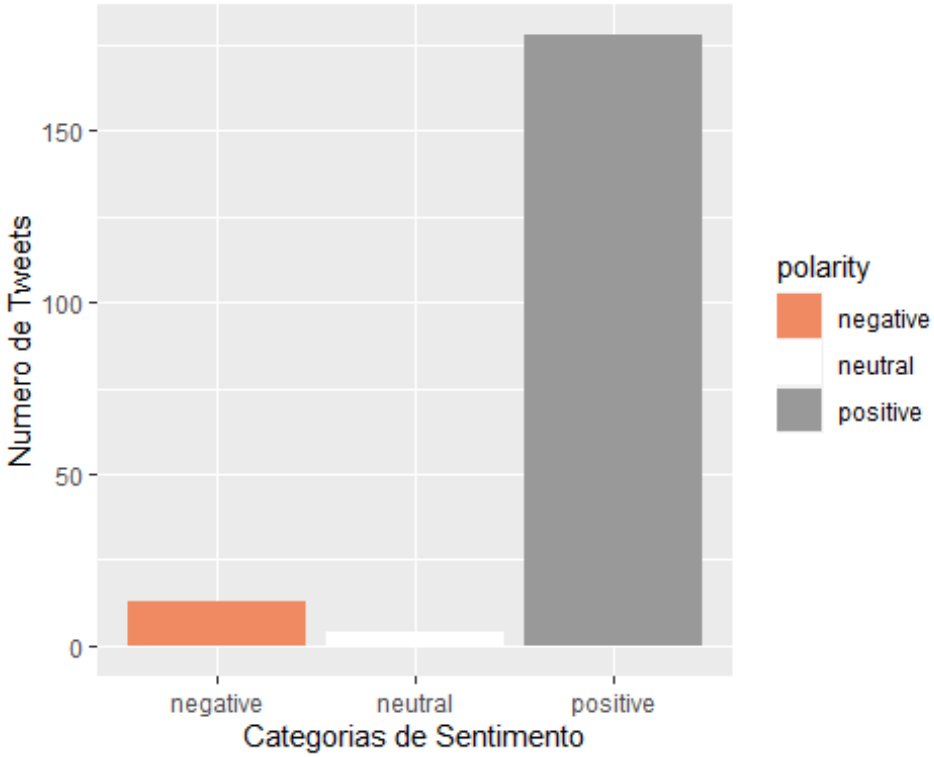
View

Finally, we use ggplot2 to visualize the results.

```
# Emotions found
ggplot(sent_df, aes(x = emotion)) +
  geom_bar(aes(y = ..count.., fill = emotion)) +
  scale_fill_brewer(palette = "Dark2") +
  labs(x = "Categorias", y = "Numero de Tweets")
```




```
# Polarity
ggplot(sent_df, aes(x = polarity)) +
  geom_bar(aes(y = ..count.., fill = polarity)) +
  scale_fill_brewer(palette = "RdGy") +
  labs(x = "Categorias de Sentimento", y = "Numero de Tweets")
```


150
100
50
0
Numero de Tweets
negative
neutral
positive
Categorias de Sentimento
polarity
negative
neutral
positive

The end