# Análise de Sentimento - Boulos (Eleição 2020)

@victorpasson

28 de Novembro de 2020

## Carregando os Pacotes Necessários:

```
library(twitteR)
library(stringr)
library(lubridate)
library(tidytext)
library(dplyr)
library(ggplot2)
library(stringr)
library(tm)
library(rmarkdown)
```

## Setando a Key da API:

**Deixarei a Key apagada por motivos de segurança.**

```
setup_twitter_oauth(consumer_key = "[redacted]",
                    consumer_secret = "[redacted]",
                    access_token = "[redacted]",
                    access_secret = "[redacted]")
```

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

## Coletando os Tweets:

**Nesse momento estamos coletando os tweets relacionados ao Boulos, desejamos 10000 tweets. Além disso, definimos a linguagem para português.**

```
tweets.boulos <- searchTwitter("Boulos", n = 10000, lang = "pt-br")
```

```
## [1] "Rate limited .... blocking for a minute and retrying up to 119 times ..."
## [1] "Rate limited .... blocking for a minute and retrying up to 118 times ..."
## [1] "Rate limited .... blocking for a minute and retrying up to 117 times ..."
## [1] "Rate limited .... blocking for a minute and retrying up to 116 times ..."
## [1] "Rate limited .... blocking for a minute and retrying up to 115 times ..."
## [1] "Rate limited .... blocking for a minute and retrying up to 114 times ..."
## [1] "Rate limited .... blocking for a minute and retrying up to 113 times ..."
```

## Extraindo Informações:

**Dos tweets coletados, extraimos o conteúdo do tweets, o usuário de quem escreveu, a data e o id. Ao final juntamos tudo isso em um objeto do tipo dataframe.**

```
text <- as.character(rep(NA, length(tweets.boulos)))
screenname <- as.character(rep(NA, length(tweets.boulos)))
created = as.POSIXct(rep(NA, length(tweets.boulos)))
id = c()

for (i in 1:length(tweets.boulos)) {
  text[i] = tweets.boulos[[i]]$text
  screenname[i] = tweets.boulos[[i]]$screenName
  created[i] = tweets.boulos[[i]]$created
  id[i] = tweets.boulos[[i]]$id
}

x = data.frame(id = id,
               screenname = screenname,
               created = created,
               text = text,
               stringsAsFactors = FALSE)

head(x, 3)
```

```
##                    id  screenname             created
## 1 1332841245593522177 EsquerdaSil 2020-11-28 21:18:10
## 2 1332841244435894272   alinnneld 2020-11-28 21:18:10
## 3 1332841242691067904 adriano9270 2020-11-28 21:18:09
##
## 1                     RT @romulo_cortes: VAMOS GASTAR OS DEDOS??? DEÊM RT... COM VONTADE...\n\nVOTE
## 2 RT @GuilhermeBoulos: Vai ser com emoção, mas nós vamos virar! #ViraSP50 #Boulos50\n\nhttps://t.co/
## 3                                                 RT @rmotta2: Quantas pessoas foram contaminadas pe
```

**Em seguida limpamos os tweets**

```
x$text <- str_replace_all(x$text,"\n", " ")
```

## Dicionário de Palavras:

**Para realizar a análise iremos usar o *oplexicon*, que basicamente nos diz a conotação das palavras em português. Tenha em mente: dependendo do contexto da análise as conotações das palavras podem mudar, por exemplo, se você estiver fazendo uma análise de tweets sobre futebol, talvez as palavras não tenham a mesma conotação de tweets sobre política. Porém, temos poucos dicionários desse tipo em português e o *oplexicon* é uma mão na roda, com ele conseguimos fazer uma série de análises.**

**Para baixar o *oplexicon* e obter mais informações [texto(link) : acesse o [site] (http://ontolp .inf.pucrs.br/Recursos/downloads-OpLexicon.php).]**

```
bing = read.delim("LearnR/Sentimento/Lexicos/oplexicon_v3.0/lexico_v3.0.txt",
                  header = FALSE,
                  sep = ",")
bing$V2 <- NULL
bing$V4 <- NULL
```

```
names(bing) <- c("word", "sentiment")
bing$sentiment <- ifelse(bing$sentiment == 1, "positive",
                         ifelse(bing$sentiment == 0, "neutral",
                                "negative"))
```

**Em seguida trazemos para um objeto as *stopwords* do pacote tm. Stopwords são palavras que não há conotação nem positiva, nem negativa, como: o, a, os, as, que, porém. Retira-las do nosso texto facilita e agiliza nossa análise, por isso a etapa de limpeza dos dados é tão importante.**

```
stopwords <- tm::stopwords("pt-br")
```

## Análise de Sentimentos:

**Agora iremos começar a análise de sentimento de fato. Primeiro pegamos palavra por palavra de cada tweet, além de transformar tudo em minúscula. Em seguida retiramos as *stopwords* e todos os *rt*, pois eles não nos idicam nada. Por fim, juntamos as palavras com sua conotação e contamos sua ocorrência. Abaixo mostro as 50 mais usadas:**

```
x %>%
  unnest_tokens(token = "words", word, text) %>%
  select(id, word) %>%
  filter(!word %in% stopwords, !word == "rt") %>%
  inner_join(bing, by = c("word" = "word")) %>%
  count(word, sentiment, sort = TRUE) %>%
  dplyr_row_slice(1:50)
```

```
##          word sentiment    n
## 1     arrumar   neutral 1037
## 2       votar  negative  606
## 3      ganhar   neutral  597
## 4         dar  negative  388
## 5       virar  negative  385
## 6       mundo  positive  357
## 7         ser  positive  345
## 8      exigir  positive  305
## 9   infectado  negative  302
## 10   entender  positive  283
## 11   derivada  negative  269
## 12   gratuito   neutral  267
## 13  confirmar  positive  249
## 14        ter   neutral  249
## 15    segundo   neutral  233
## 16      saber   neutral  204
## 17      dizer   neutral  202
## 18      estar  positive  202
## 19     vencer  negative  195
## 20      fazer   neutral  182
## 21     aliada  positive  177
## 22   esquerda   neutral  174
## 23     virada   neutral  173
## 24      bruno  negative  167
## 25   negativo  negative  156
```

```
## 26   positivo  positive  156
## 27     alegre  positive  155
## 28 aglomerado   neutral  153
## 29     eleito   neutral  139
## 30    diretor  positive  132
## 31 presidente   neutral  116
## 32       azul   neutral   95
## 33    invasor  negative   95
## 34         ir   neutral   94
## 35  derrotado  negative   91
## 36   declarar   neutral   90
## 37    invadir   neutral   89
## 38       cara  negative   85
## 39     perder  negative   82
## 40   primeiro   neutral   81
## 41        ver  positive   81
## 42     mortal  negative   80
## 43       rica  positive   78
## 44    sozinho   neutral   78
## 45     chamar  positive   77
## 46       nada   neutral   75
## 47    radical   neutral   74
## 48       real  positive   74
## 49    direita   neutral   73
## 50       vivo  positive   73
```

**Contamos a quantidade de palavras negativas e positivas. Vemos que há muito mais palavras positivas do que negativas, isso talvez reflita como o candidato é visto pela maioria das pessoas nas redes sociais. Isso não indica se é um bom ou mal candidato, só reflete, em partes, qual o sentimento das pessoas sobre ele no Twitter. De fato, na campanha de 2020 o candidato colocou todas as suas forças nas redes sociais e pesquisas mostram que seu eleitorado é na maioria composto por jovens universitários, isso reflete diretamente no seu posicionamento no Twitter.**

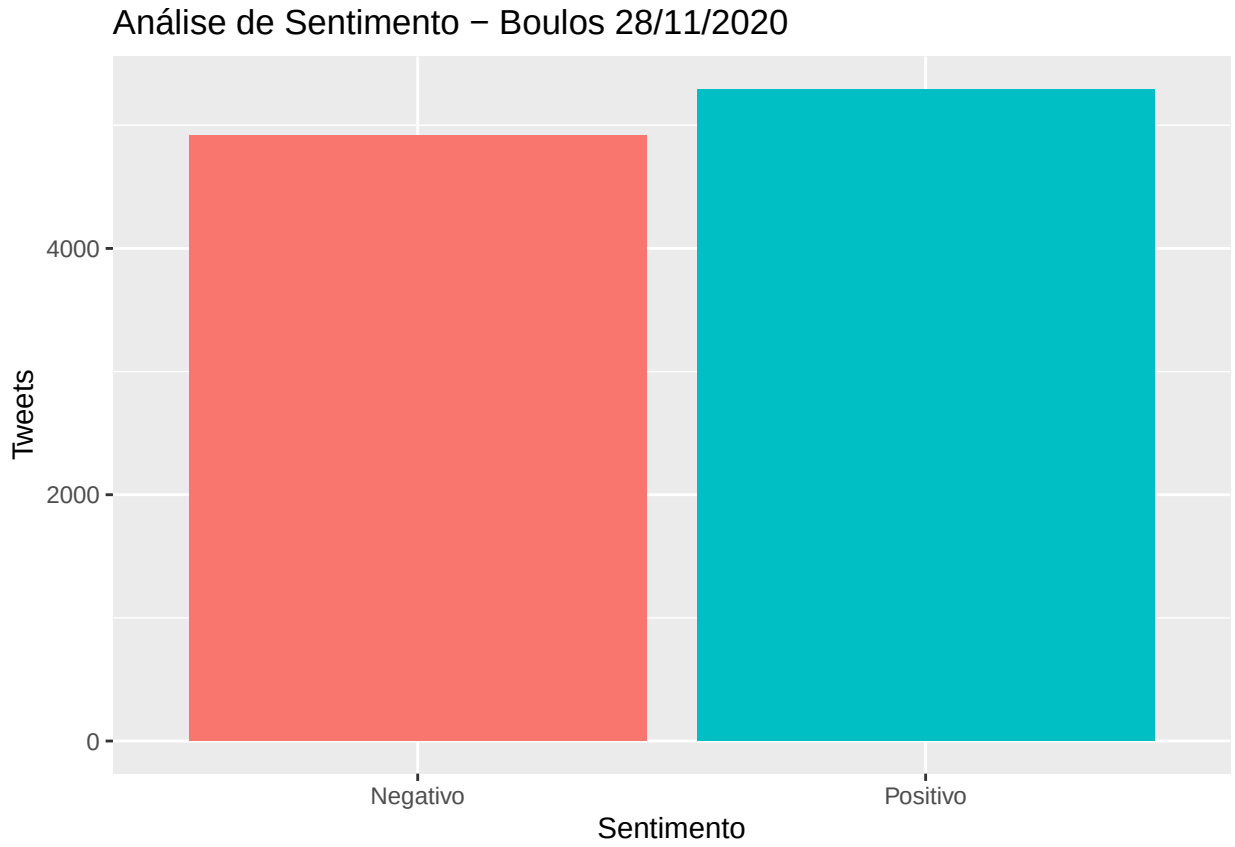
```
x %>%
  unnest_tokens(token = "words", word, text) %>%
  select(id, word) %>%
  filter(!word %in% stopwords, !word == "rt") %>%
  inner_join(bing, by = c("word" = "word")) %>%
  count(word, sentiment, sort = TRUE) %>%
  group_by(sentiment) %>%
  summarise(n = sum(n))
```

```
## # A tibble: 3 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   4919
## 2 neutral    7284
## 3 positive   5294
```

**O gráfico abaixo só reflete o que eu havia dito anteriormente, porém de maneira visual.**

```
x %>%
  unnest_tokens(token = "words", word, text) %>%
```

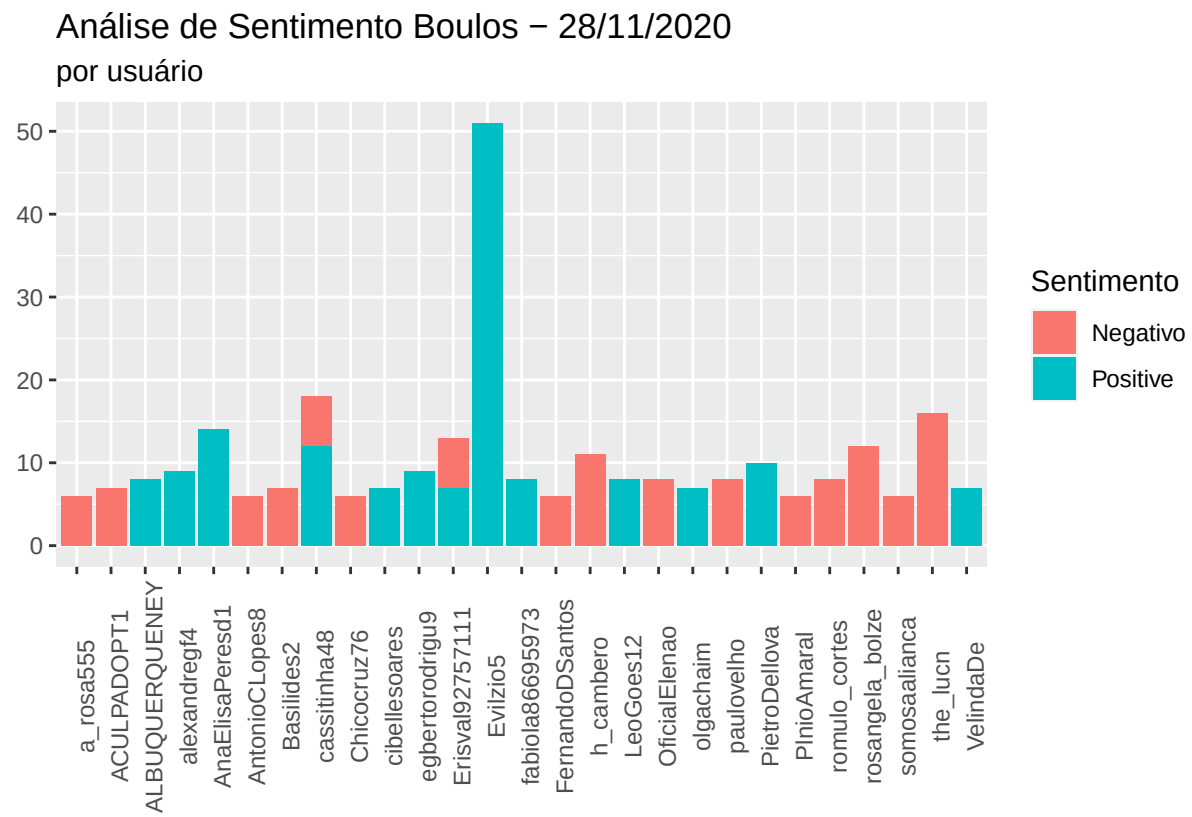
```
  filter(!word %in% stopwords, !word == "rt") %>%
  select(id, created, word) %>%
  inner_join(bing, by = c("word" = "word")) %>%
  select(id, created, sentiment) %>%
  count(id, created, sentiment) %>%
  filter(sentiment == "positive" | sentiment == "negative") %>%
  group_by(Date = as.Date(ymd_hms(created)), sentiment) %>%
  summarize(total = sum(n)) %>%
  ggplot(aes(x = sentiment, y = total, fill = sentiment)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  labs(x = "Sentimento",
       y = "Tweets",
       title = "Análise de Sentimento - Boulos 28/11/2020") +
  scale_x_discrete(position = "bottom",
                   labels =  c("negative" = "Negativo",
                               "positive" = "Positivo"))
```




### Para finalizar pegamos, dos dados que coletamos, os usuários com mais matchs entre palavras. Geralmente, nesse tipo de análise, a definição se um tweet possui conotação positiva ou negativa é feita pela subtração do número de palavras positivas pelo número de palavras negativas. Se o resultado da subtração for positivo, muito provavelmente, o sentimento do tweet é positivo, o inverso também é válido.

**No exemplo abaixo podemos ver realmente isso, pegando como exemplo o usuário com maior diferença entre positivos do que negativos vemos que de fato ele tem a visão pró Boulos e isso é refletido pelo grande excesso de palavras positivas.**

```
x %>%
  unnest_tokens(token = "words", word, text) %>%
  filter(!word %in% stopwords, !word == "rt") %>%
  select(id, screenname, created, word) %>%
  inner_join(bing, by = c("word" = "word")) %>%
  count(screenname, sentiment, sort = TRUE) %>%
  filter(sentiment != "neutral") %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ggplot(aes(x=factor(screenname), y = n, fill= sentiment))+
  geom_bar(stat='identity') +
  theme(axis.text.x = element_text(angle=90)) +
  labs(x = '', y = '',
       title = "Análise de Sentimento Boulos - 28/11/2020",
       subtitle = "por usuário",
       fill = "Sentimento") +
  scale_fill_discrete(labels = c("negative" = "Negativo",
                                 "positive" = "Positive"))
```



**Para essa análise tive como base o código disponibilizado no RPubs pelo Sumit Kumar. Deixo o link abaixo para acesso:**

http://rpubs.com/sumitkumar-00/twitter_sentiment_analysis