Apêndice I: Análise de Sentimentos com Dicionários Léxicos no R

Dados do Twitter sobre o Coronavirus (Inglês)

Coleta de Dados

Uma aplicação muito recorrente da Análise de Sentimentos é analisar dados de mídias sociais. E, para o caso da análise de texto, o Twitter é o mais utilizado. Felizmente, é possível obter dados de Tweets diretamente no R através da API com uma conta de desenvolvedor e por meio do pacote `rtweet`. Para isso é necessário obter as chaves de acesso da conta de desenvolvedor.

```
library(rtweet)
library(dplyr)
library(tidyr)
library(tidytext)
library(textdata)
library(ggplot2)
library(reshape2)
library(wordcloud)
library(stringr)
library(shiny)
library(DT)
```

```
token <- create_token(
  app = "AppName",
  consumer_key = "XXXXXXXXXXXXXXXXXXXXX",
  consumer_secret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  access_token="XXXXXXXXXXXXXXXXXXXXX-XXXXXXXXXXXXXXXXXXX",
  access_secret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
```

Feito isso, é possível coletar os dados através da função `search_tweets()`, pesquisando uma *hashtag*. Neste exempo, serão obtidos cerca de 5 mil tweets com **#Corona**, em inglês e sem considerar os *retweets*.

Vejamos um exemplo de tweet coletado:

```
Corona <- search_tweets("#corona", n=5000, include_rts = F, lang = "en")

cat(Corona$text[150])
```

```
## Very useful thread and catches in health insurance claims #corona #healthinsurance
## Be aware and be safe https://t.co/KcqgVLeoz7
```

São fornecidas uma série de informações como o nome de usuário, localidade, *hashtags* utilizadas, além do texto do tweet.

```
tweets.Corona <- Corona %>% select(screen_name, text)
```

Pré-Processamento

Antes das análises é importante aplicar algumas técnicas de pré-processamento. Serão removidos links dos tweets e será aplicada a *tokenization* para dividir os textos em palavras, além de transformar todos os caracteres em *lowercase* e remover pontuações.

```
tweets.Corona <- tweets.Corona %>%
  mutate(stripped_text=gsub("http\\S+","",tweets.Corona$text)) # Remove links

cat(tweets.Corona$stripped_text[150])
```

```
## Very useful thread and catches in health insurance claims #corona #healthinsurance
## Be aware and be safe
```

```
tweets.Corona_stem <- tweets.Corona %>% # Tokenization
  select(stripped_text) %>%
  unnest_tokens(word, stripped_text)

knitr::kable((tweets.Corona_stem[700:710,]))
```

word
closing
in
the
capital
11pm
6am
since
friday
night
most
publicans

Por fim, serão removidas as *stopwords*.

```
cleaned_tweets.Corona <- tweets.Corona_stem %>% # Remove stopwords
  anti_join(stop_words)

knitr::kable(cleaned_tweets.Corona[700:710,])
```

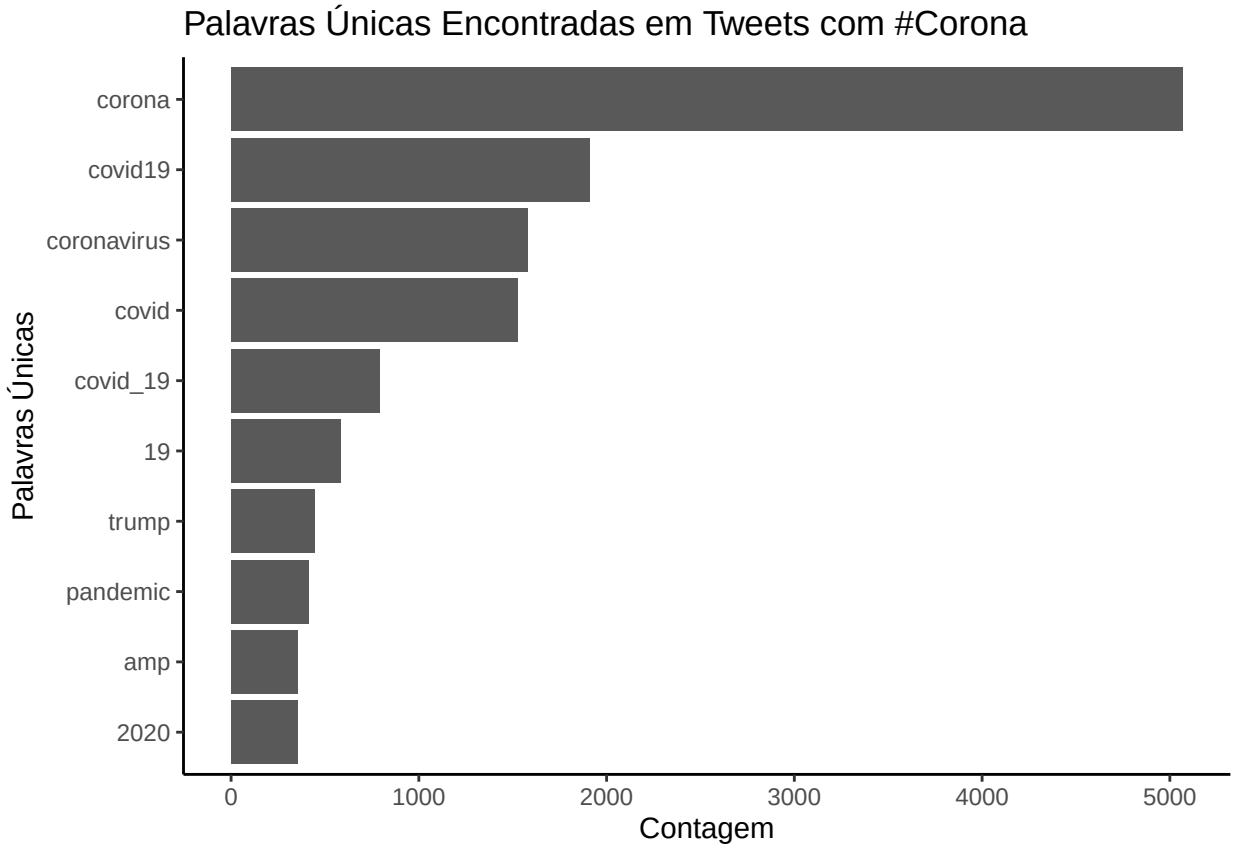
word
vicious
corona
circle
absolutely
recent

word
claims
trump
walk
water
corona
halo

Análise Descritiva Geral

Nesta primeira análise todos os textos foram considerados como um só para que se possa ter uma noção geral do que se têm falado acerca do coronavírus.

```
cleaned_tweets.Corona %>%
  count(word, sort=TRUE) %>%
  top_n(10) %>%
  mutate(word = reorder(word,n)) %>%
  ggplot(aes(x=word, y=n)) + geom_col() + xlab(NULL) + coord_flip() +
  theme_classic() +
  labs(y= "Contagem", x="Palavras Únicas",
       title="Palavras Únicas Encontradas em Tweets com #Corona")
```



É possível ver que as palavras mais utilizadas são variações de corona e covid-19.

Dicionários Léxicos

O pacote `tidytext` oferece, por meio da função `get_sentiments()`, 4 dicionários léxicos: *bing*, *afinn*, *nrc* e *loughran*. Todos possuem palavras do inglês e uma informação sentimental correspondente.
No caso dos dicionários *bing*, *nrc* e *loughran*, cada palavra é associada a um sentimento.

```
get_sentiments("bing") %>% group_by(sentiment) %>% count()
```

```
## # A tibble: 2 x 2
## # Groups:   sentiment [2]
##   sentiment     n
##   <chr>     <int>
## 1 negative   4781
## 2 positive   2005
```

```
get_sentiments("nrc") %>% group_by(sentiment) %>% count()
```

```
## # A tibble: 10 x 2
## # Groups:   sentiment [10]
##    sentiment        n
##    <chr>        <int>
##  1 anger         1247
##  2 anticipation   839
##  3 disgust       1058
##  4 fear          1476
##  5 joy            689
##  6 negative      3324
##  7 positive      2312
##  8 sadness       1191
##  9 surprise       534
## 10 trust         1231
```

```
get_sentiments("loughran") %>% group_by(sentiment) %>% count()
```

```
## # A tibble: 6 x 2
## # Groups:   sentiment [6]
##   sentiment        n
##   <chr>        <int>
## 1 constraining   184
## 2 litigious      904
## 3 negative      2355
## 4 positive       354
## 5 superfluous     56
## 6 uncertainty    297
```

Entretanto, são categorias diferentes. O dicionário **bing** possui apenas as categorias positivo e negativo. Os dicionários **nrc** e **loughran** oferencem outras categorias além dessas.

Já no dicionário *afinn*, cada palavra possui uma intensidade sentimental que é um valor inteiro entre -5 e 5. Se esse valor é positivo, o sentimento é positivo. Se for igual a zero, o mesmo é neutro. Caso contrário, é negativo.

```
get_sentiments("afinn") %>% group_by(value) %>% count()
```

```
## # A tibble: 11 x 2
## # Groups:   value [11]
##    value     n
##    <dbl> <int>
##  1    -5    16
##  2    -4    43
##  3    -3   264
##  4    -2   966
##  5    -1   309
##  6     0     1
##  7     1   208
##  8     2   448
##  9     3   172
## 10     4    45
## 11     5     5
```

Neste exemplo será usado o dicionário *bing* que possui apenas as categorias de sentimento positivo e negativo.

```
bing_Corona <- cleaned_tweets.Corona %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=TRUE) %>%
  ungroup()

bing_Corona
```

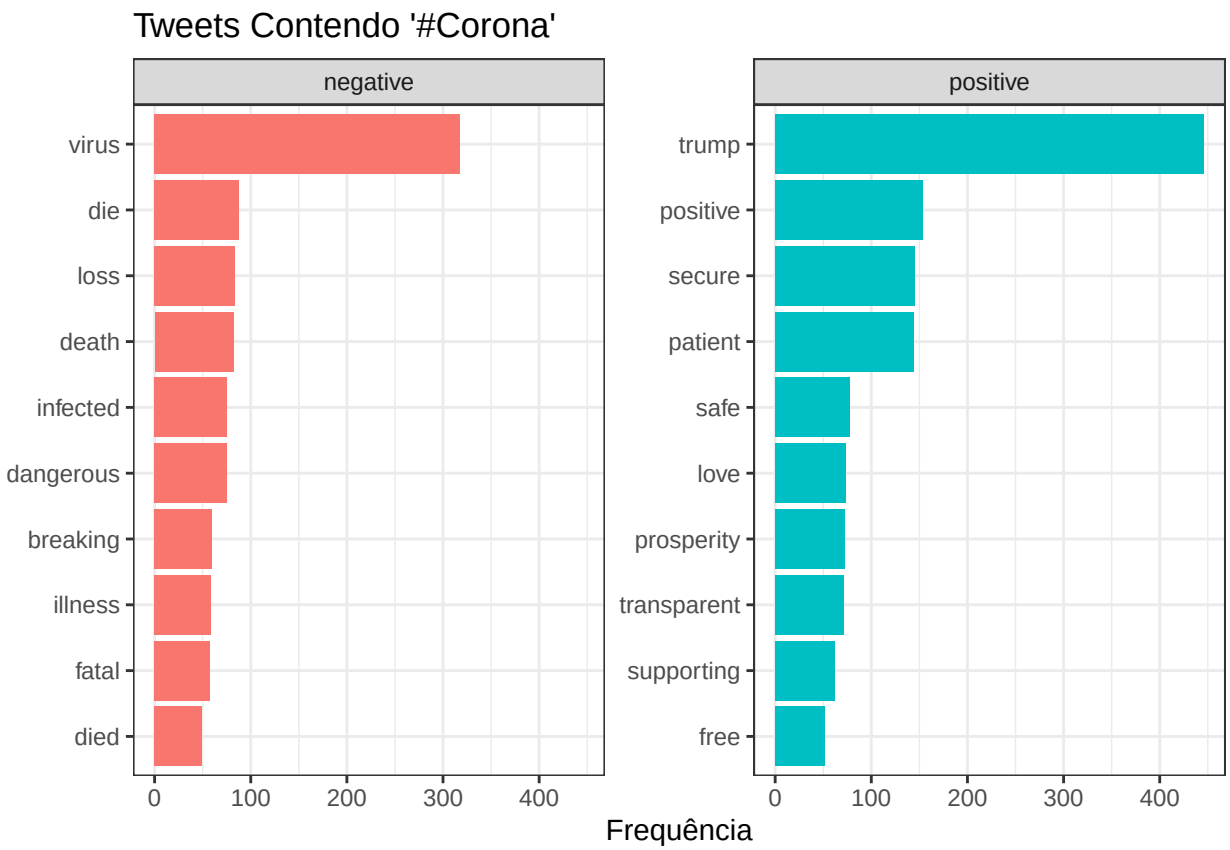
```
## # A tibble: 1,263 x 3
##    word      sentiment     n
##    <chr>     <chr>     <int>
##  1 trump     positive    446
##  2 virus     negative    318
##  3 positive  positive    153
##  4 secure    positive    145
##  5 patient   positive    144
##  6 die       negative     88
##  7 loss      negative     84
##  8 death     negative     82
##  9 safe      positive     77
## 10 dangerous negative     75
## # ... with 1,253 more rows
```

Dessa forma, cada palavra do banco que está presente no dicionário é associada a um sentimento positivo ou negativo.

É possível, então, verificarmos quais as palavras mais frequentes para cada sentimento.

```
bing_Corona %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word,n)) %>%
```

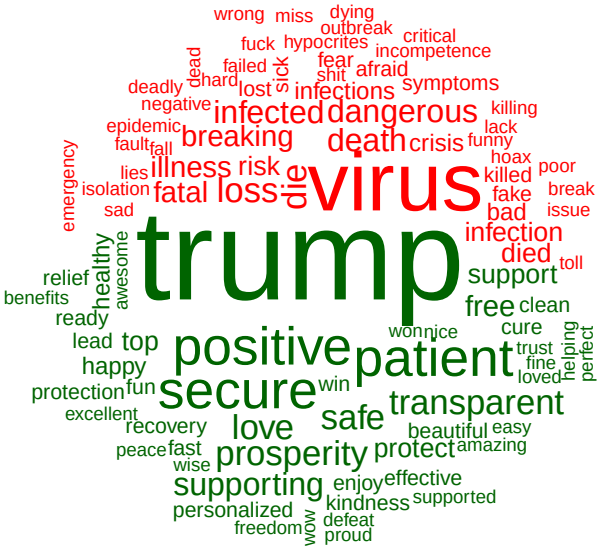
```
  ggplot(aes(word, n, fill=sentiment)) + geom_col(show.legend = F) +
  facet_wrap(~sentiment, scales="free_y") +
  labs(title="Tweets Contendo '#Corona'",
       y="Frequência",
       x=NULL) +
  coord_flip() + theme_bw()
```



Outra descritiva muito interessante é a nuvem de palavras que permite observar as palavras de modo que seu tamanho é relativo à frequência e a cor ao sentimento associado.

```
bing_Corona %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("red", "darkgreen"),
                   max.words = 100)
```

negative



positive

Dicionário Léxico por Tweet

Além da análise geral é possível utilizar o dicionário léxico em cada texto separadamente e, assim, associoar um sentimento a cada tweet individualmente.

Para isso, será utilizado o dicionário *afinn* que associa um valor numérico a cada palavra. Para classificar o tweet serão somados os valores de intensidade de sentimento de cada palavra do texto. Se a soma for positiva, o tweet será classificado como positivo. Se a soma resultar em zero, o sentimento é neutro. Caso contrário, negativo.

```
for(i in 1:nrow(tweets.Corona)){
  d <- (tweets.Corona[i,3] %>% unnest_tokens(word, stripped_text) %>%
    anti_join(stop_words) %>% inner_join(get_sentiments("afinn")))$value %>% sum()
  tweets.Corona$value[i] <- d
}

tweets.Corona <- tweets.Corona %>%
  mutate(sentiment=case_when(value>0 ~ "positive",
                             value <0 ~ "negative",
                             TRUE ~ "neutral"))
```

```
tweets.Corona$sentiment %>% table()
```

```
## .
## negative  neutral positive
```

```
##     1374     1994     1388
```

Com isso, é possível ver um equilíbrio entre tweets positivos e negativos.

Utilizando as técnicas aplicadas acima é possível classificar qualquer texto em inglês utilizando o dicionário *afinn*.
Por exemplo, a frase: "You're awesome!"

```
d <- (as_tibble("You're awesome!") %>% unnest_tokens(word, value) %>%
    anti_join(stop_words) %>% inner_join(get_sentiments("afinn")))$value %>% sum()
if(d>0){sentiment="Positive"}else if(d==0){sentiment="Neutral"}else{sentiment="Negative"}

knitr::kable(tibble(Texto="You're awesome!", Sentimento=sentiment, Intensidade=d))
```

Texto	Sentimento	Intensidade
You're awesome!	Positive	4

Dicionário Léxico em Português

De forma análoga à apresentada anteriormente, é possível utilizar um dicionário léxico para classificar textos em português. Será usado o dicionário oferecido pelo pacote `lexiconPT` que associa, a cada palavra, uma polaridade negativa (-1), positiva (1) ou neutra (0).

```
devtools::install_github("sillasgonzaga/lexiconPT")
```

```
library(lexiconPT)
library(readr)

knitr::kable(lexiconPT::oplexicon_v3.0[1100:1110,c(1,3)])
```

	term	polarity
1100	acidificar	1
1101	acido	-1
1102	acidos	-1
1103	aciganada	-1
1104	aciganadas	-1
1105	aciganado	-1
1106	aciganados	-1
1107	acintosa	-1
1108	acintosas	-1
1109	acintoso	-1
1110	acintosos	-1

Além do dicionário, será usado uma lista de stopwords em português disponibilizado pelo LabAPE.

```
stopwordspt <- read_csv(
    file = "http://www.labape.com.br/rprimi/ds/stopwords.txt",
    col_names = 'word')
```

Qual o sentimento presente na frase “Estou triste hoje.”?

```
lexicon_pt <- lexiconPT::oplexicon_v3.0 %>% select(term, polarity) %>%
  rename(word=term, value=polarity)
```

```
  d <- (as_tibble("Estou triste hoje.") %>% unnest_tokens(word, value) %>%
    anti_join(stopwordspt) %>% inner_join(lexicon_pt))$value %>% sum()
if(d>0){sentiment="Positive"}else if(d==0){sentiment="Neutral"}else{sentiment="Negative"}

  knitr::kable(tibble(Texto="Estou triste hoje.", Sentimento=sentiment, Intensidade=d))
```

Texto	Sentimento	Intensidade
Estou triste hoje.	Negative	-1