Análise de Sentimento em R

Fernando Tsutomu Hara

6/9/2020

% !TEX encoding = UTF-8 Unicode

Análise de Sentimentos em Redes Sociais utilizando R

Este projeto visa construir uma análise de sentimento baseado em frases de usuários do Twitter sobre determinado assunto, neste caso eu fiz sobre o emprego. Para fazer a comparação da análise de setimento foi utilizado o arquivo SentilexPT. O SentilexPT é um léxico para o idioma Português que está disponivel para pesquisa e desenvolvimento, nele existem diversas palavras e expresssões que têm um "score" de sentimento (negativo=-1, neutro=0 e positivo=1).

Etapa 1 - Autenticação.

Para realizar a autenticação com o Twitter é necessário ter uma conta de usuário e cadastrar-se como developer para criar uma app. Este processo pode demorar algumas semanas. Após criada a app será gerada 4 chaves que serão utilizadas nesse projeto. Após a finalização do projeto essas chaves serão excluídas por questão de segurança.

```
library(twitteR)
library(httr)

# Chaves de autenticação no Twitter
key <- "HJpuSXz3JRn2pkis1PvnWf1d0"
secret <- "sHwZe2q5ytoZMHP5Bowa2WcDmcwErG0lhowmsJzeKyMdLlxfat"
token <- "155073443-q9YjaNnGkNaLTis6vSs2UOUtLTf0GJvyMvj4zajE"
tokensecret <- "kDCs0Lou5eYCZdIkyb5vCkyd7PopnSsWXJXyKvqY1awT8"

# Autenticação. Responda 1 quando perguntado sobre utilizar direct connection.
setup_twitter_oauth(key, secret, token, tokensecret)
```

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

Etapa 2 - Captura dos Tweets

Nesta etapa os tweets sobre o assunto emprego são capturados. Estão sendo capturados 2000 tweets em lingua portuguesa.

```
# Captura dos twitters
tema <- "emprego"
qnt_tweets <- 3000
lingua <- 'pt'
tweet <- searchTwitter(tema, n = qnt_tweets, lang = lingua)

# Visualizando as primeiras linhas do objeto tweet
head(tweet)
```

```
## [[1]]
## [1] "caba_vazquez: @lucianoayan É só pedir um emprego pro Luciano hang na havan que ele paga o proces
##
## [[2]]
## [1] "DelcioLuz: RT @jrguzzofatos: O primeiro mandamento do Brasil politicamente correto é fazer tudo
##
## [[3]]
## [1] "masxeroso: RT @maiaratoniote: qq custa me oferece um emprego"
##
## [[4]]
## [1] "LucianoNudes: Previsões de queda do PIB chegando a 10%, índices de desemprego recorde, taxa de
##
## [[5]]
## [1] "nottheraven: RT @marianafss_: Povo de BH, ja falei aqui e seguimos na luta rs\n\nMeu namorado p
##
## [[6]]
## [1] "lialinetti: alguém me dá um emprego pra eu poder viajar pra ver essa menina POR FAVO https://t.
```

Etapa 3 - Limpeza e preparação dos dados através de text mining.

Nesta etapa vamos realizar o processo de limpeza dos dados como remover pontuação, converter os dados para letras minúsculas e remover as stopwords (palavras comuns do idioma português, neste caso). Também vamos converter os tweets coletados em um objeto do tipo Corpus, que armazena dados e metadados. Assim os dados ficarão prontos a análise de setimento.

```
# Limpeza dos dados coletados (text mining)
library(SnowballC)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

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

```
options(warn=-1)

# Criando uma cópia dos dados originais
tweets <- tweet

#transformando a lista em um vetor.
tweets <- sapply(tweets, function(x) x$getText())
#tranformando os textos em UTF-8
tweets <- iconv(tweets, to = "utf-8", sub="")
# Removendo caracteres especiais
tweets = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", tweets)
# Removendo @
tweets = gsub("@\\w+", "", tweets)
# Removendo pontuação
tweets = gsub("[[:punct:]]", "", tweets)
# Removendo digitos
tweets = gsub("[[:digit:]]", "", tweets)
# Removendo links html
```

```
tweets = gsub("http\\w+", "", tweets)
# Removendo \n
tweets = gsub("\n", " ", tweets)
# Removendo espacos desnecessários
tweets = gsub("[ \t]{2,}", " ", tweets)
tweets = gsub("^\\s+|\\s+$", "", tweets)
# Tranformando letras maiusculas em minusculas
tweets = tolower(tweets)

# Extraindo stopwords (palavras comuns) dos tweets
stopwords = stopwords("portuguese")
stopwords = stopwords[-157]
rm_stopwords <- function(string, words) {
  stopifnot(is.character(string), is.character(words))
  spltted <- strsplit(string, " ", fixed = TRUE) # fixed = TRUE for speedup
  vapply(spltted, function(x) paste(x[!tolower(x) %in% words],
                                    collapse = " "), character(1))
}
tweets_sem_stopwords <- rm_stopwords(tweets, stopwords)

# Excluindo tweets vazios
vet <- c()
j=1
for(i in 1:length(tweets_sem_stopwords)){
  if(tweets_sem_stopwords[i] == ""){
    vet[j] <- i
    j=j+1
  }
}
if(length(vet) > 0)
  tweets_sem_stopwords <- tweets_sem_stopwords[-vet]

#Tirando os tweets duplicados
tweets_unicos <- (unique(tweets_sem_stopwords))
# Transformando os dados em Corpus
tweetc <- Corpus(VectorSource(tweets_sem_stopwords))
# Remove pontuação
tweetc <- tm_map(tweetc, removePunctuation)
# Remove números
tweetc <- tm_map(tweetc, function(x)removeWords(x, stopwords()))
# Convertendo o objeto texto para o formato de matriz
tweet_mat <- TermDocumentMatrix(tweetc)
```

Etapa 4 - Associações.

Aqui faremos algumas associações com as palavra que está sendo buscada, neste caso, a palavra emprego. E também será feita uma word cloud (nuvem de palavras) com as palavras que estão associadas à emprego. Para isso, serão utilizados os pacotes RColorBrewer e wordcloud.

```
# Encontrando as palavras que aparecem com mais frequência
findFreqTerms(tweet_mat, lowfreq = 15)
```

```
##   [1] "emprego"       "pedir"         "pro"           "brasil"
##   [5] "correto"       "covid"         "diz"           "fazer"
```

```
##   [9] "mandamento"    "matéria"       "oms"           "politicamente"
##  [13] "primeiro"      "tudo"          "desemprego"    "aqui"
##  [17] "causa"         "falei"         "fazendo"       "luta"
##  [21] "marmitas"      "namorado"      "perdeu"        "povo"
##  [25] "seguimos"      "alguém"        "dá"            "poder"
##  [29] "pra"           "ver"           "entrevista"    "gente"
##  [33] "arrumar"       "né"            "pandemia"      "agora"
##  [37] "nao"           "preciso"       "acho"          "qualquer"
##  [41] "sendo"         "passar"        "ser"           "sim"
##  [45] "nada"          "pessoas"       "cobre"         "empresa"
##  [49] "governador"    "stf"           "digno"         "comprar"
##  [53] "quero"         "dinheiro"      "tá"            "pai"
##  [57] "mil"           "fetiche"       "fim"           "lista"
##  [61] "mental"        "saúde"         "ter"           "vida"
##  [65] "mãe"           "toda"          "porque"        "tô"
##  [69] "caso"          "chamado"       "aí"            "medo"
##  [73] "salário"       "todos"         "trabalhar"     "cara"
##  [77] "vou"           "após"          "empresas"      "fala"
##  [81] "perde"         "deus"          "ajudar"        "quarentena"
##  [85] "botou"         "esqueça"       "fechou"        "miséria"
##  [89] "negócio"       "prefeito"      "tomou"         "coisas"
##  [93] "sei"           "bom"           "mim"           "nunca"
##  [97] "sempre"        "cargo"         "gostar"        "ministro"
## [101] "pré"           "requisito"     "vaga"          "conseguir"
## [105] "dia"           "meio"          "ontem"         "casa"
## [109] "semana"        "tanto"         "trabalho"      "momento"
## [113] "sair"          "bem"           "brasileiro"    "proposta"
## [117] "quer"          "remunerado"    "consegui"      "perder"
## [121] "porra"         "cabide"        "posso"         "dar"
## [125] "vai"           "então"         "outro"         "hoje"
## [129] "lá"            "vão"           "formato"       "maior"
## [133] "fico"          "manter"        "tão"           "hora"
## [137] "logo"          "sabe"          "queria"        "vagas"
## [141] "ano"           "tava"          "fica"          "pessoa"
## [145] "tipo"          "meses"         "alguns"        "enfermagem"
## [149] "inciativa"     "socorrista"    "técnica"       "coisa"
## [153] "dessa"         "merda"         "procurar"      "assim"
## [157] "aguento"       "arruma"        "desse"         "ficar"
## [161] "faz"           "daqui"         "arranjar"      "sobre"
## [165] "vamos"         "galera"        "onde"          "todo"
## [169] "atrás"         "benção"        "estar"         "tempos"
## [173] "trabalhando"   "\U0001f3fd"    "cloroquina"    "começar"
## [177] "perdi"         "amigos"        "anos"          "sonha"
## [181] "tirar"         "falando"       "ainda"         "novo"
## [185] "guzzo"         "conta"         "currículo"     "vem"
## [189] "menos"         "renda"         "inferno"       "foda"
## [193] "presidente"    "pode"          "realmente"     "boa"
## [197] "mundo"         "preocupado"    "faculdade"     "amiga"
## [201] "grande"        "maioria"       "funciona"      "gripezinha"
## [205] "precisava"
```

```r
# Buscando associações
findAssocs(tweet_mat, 'emprego', 0.1)
```

```
## $emprego
```

```
##         perdeu      namorado         causa         falei       fazendo
##           0.26          0.23          0.22          0.22          0.22
##           luta      marmitas      seguimos          aqui          povo
##           0.22          0.22          0.22          0.21          0.21
##         queria           pra    entrevista       arrumar           ter
##           0.20          0.14          0.12          0.12          0.12
## entrevistador       preciso    governador         tomou          vaga
##           0.12          0.11          0.11          0.11          0.10
```

```
# Gerando uma ""nuvem de palavras"
library(RColorBrewer)
library(wordcloud)

# Gerando uma nuvem palavras
pal2 <- brewer.pal(8,"Dark2")

wordcloud(tweetc,
          min.freq = 2,
          scale = c(4,1),
          random.color = F,
          max.word = 60,
          random.order = F,
          colors = pal2)
```

Etapa 5 - Análise de Sentimento.

Agora que os dados estão prontos, limpos e até ja fizemos alguma exploração com eles, vamos realizar a etapa de análise de sentimento, onde será criado um data frame com os tweets limpos, uma pontuação (score) e o sentimento (negativo, positivo ou neutro). O arquivo SenlexPT já foi extraido, limpo e gravado em um

aquivo .csv em programa em R que estará junto no GitHub.

```
# Lendo o arquivo SentilexPT
sentilex <- read.csv("sentilex.csv", header = TRUE, row.names = 1)
# Convertendo a coluna Word de factor para character
sentilex$Word <- as.character(sentilex$Word)

# Criando o data frame sentimentos, com os tweets e a pontuação. Inicialmente igual a 0.
sentimentos <- data.frame("tweet"=tweets_unicos,
                          "score"=rep(0, length(tweets_unicos)))
# Convertendo a coluna tweet de factor para character
sentimentos$tweet <- as.character(sentimentos$tweet)

# Calculando o score de sentimento em cada tweet.
score <- lapply(sentimentos$tweet,
                function(sentence, word, feeling){
                  sentence <- strsplit(sentence, " ", fixed = TRUE)
                  unlist_sentence <- unlist(sentence, use.names=FALSE)
                  list_score <- match(unlist_sentence, word)
                  list_score<-list_score[!is.na(list_score)]
                  score <- sum(feeling[list_score])
                  return(score)
                }, sentilex$Word, sentilex$Feeling)

# Gravando o score em na coluna score do data frame.
score <- unlist(score, use.names=FALSE)
sentimentos$score <- score

#Criando a coluna sentimento com o tipo sentimento em cada publicação.
sentimento <- sapply(sentimentos$score,
                     function(x){
                       if(x > 0)
                         return("Positivo")
                       else if(x == 0)
                         return("Neutro")
                       else
                         return("Negativo")
                       })
sentimentos$sentimento <- sentimento
```

Etapa 6 - Análise Gráfica.

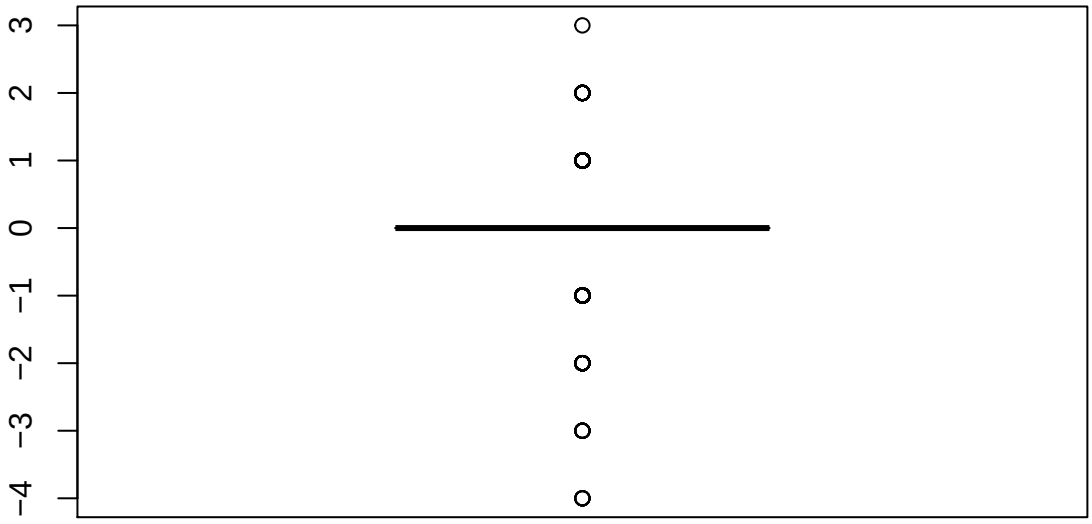
Nesta ultima etapa iremos utilizar gráficos para visualizar os resultados. Iremos criar um boxplot para verificar como os sentimentos estão ditribuidos, um histograma para ver como está a distribuição do score de sentimento e por ultimo um gráfico de barras para que irá mostrar a contagem dos 3 tipos de sentimentos.

```
#Tabela com a proporção de cada tipo de sentimento
prop.table(table(sentimentos$sentimento))
```

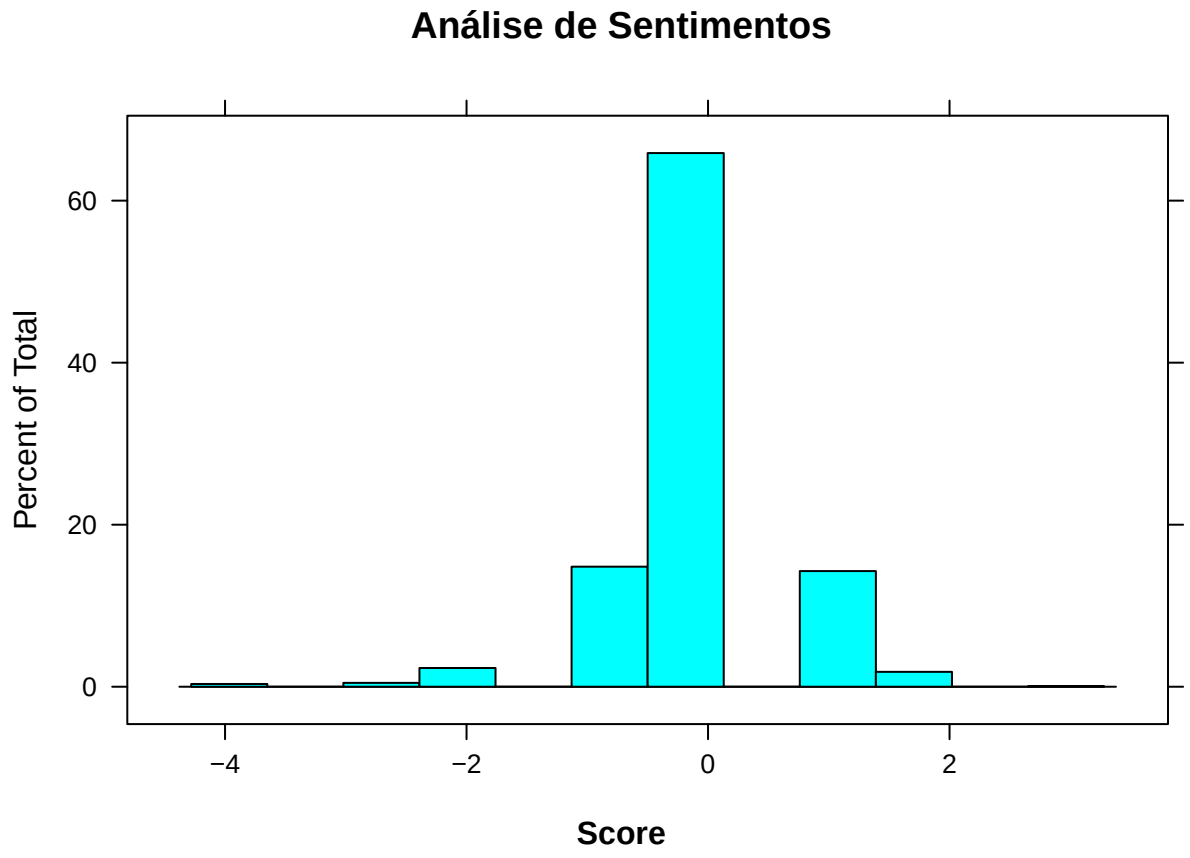
```
##
##  Negativo    Neutro  Positivo
## 0.1794697 0.6587356 0.1617947
```

```
#boxplot
boxplot(sentimentos$score)
```

```r
#Histograma
library("lattice")
```




```r
histogram(data = sentimentos, ~score, main = "Análise de Sentimentos",
          xlab = "", sub = "Score")
```

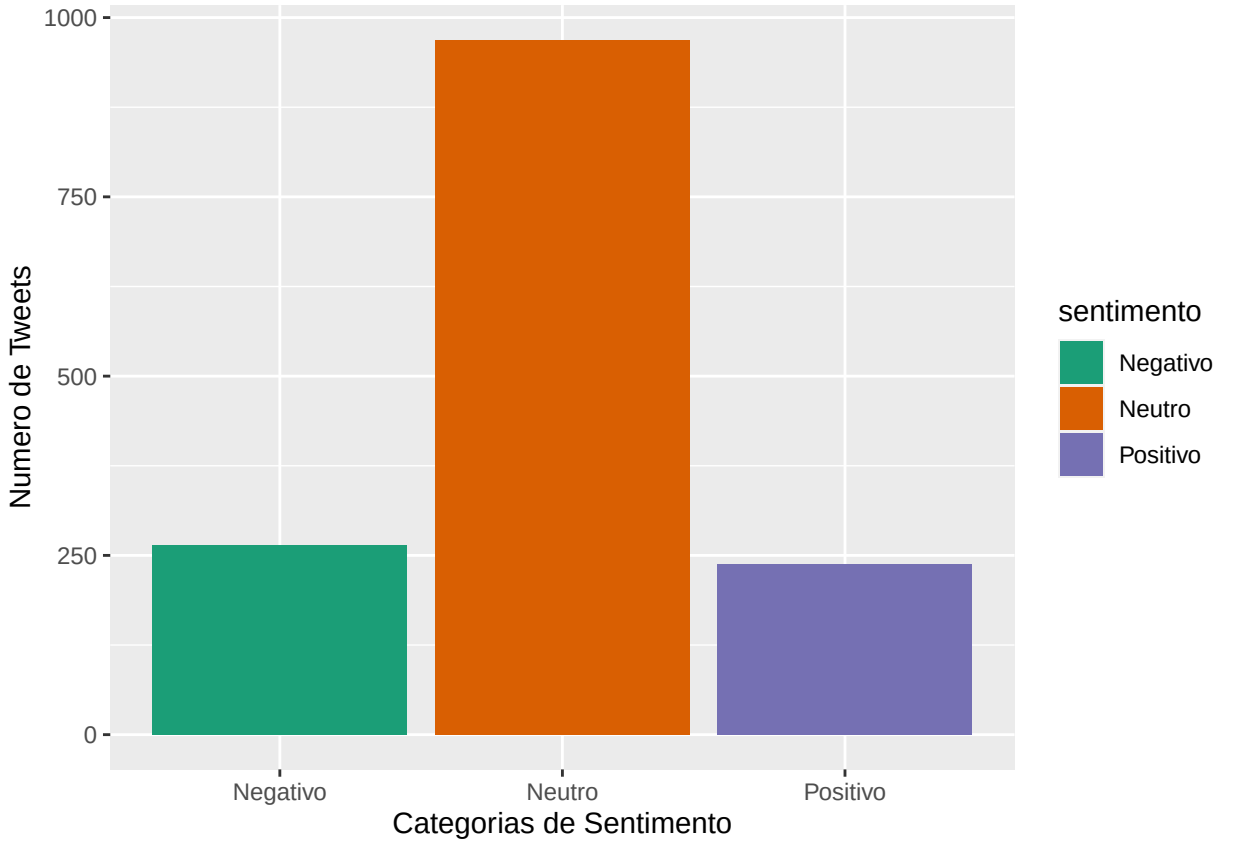



```r
#Gráfico de barras
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

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

```
ggplot(sentimentos, aes(x=sentimento)) +
  geom_bar(aes(y=..count.., fill=sentimento)) +
  scale_fill_brewer(palette="Dark2") +
  labs(x = "Categorias de Sentimento", y = "Numero de Tweets")
```



Conclusão.

A análise de sentimentos atráves de um dicionário de palavras, neste caso o sentiLexPT, é uma forma facil e divertida de analizar o sentimento das pessoas, porém não é muito eficiente, pois ele mede o sentimento através de palavras separadas. Quando algumas palavras são conectadas, elas podem expressar um sentimento completamente diferente, mas mesmo assim o algoritmo ainda consegue acertar em algumas frases.

Fim

Fernando Tsutomu Hara