

Data Challenge - Carrefour

Análise de sentimentos aplicado a mensagens do Twitter de clientes do Carrefour

Para esse projeto tentaremos responder a partir de mensagens do Twitter, a seguinte pergunta: “Qual é o sentimento dos usuários do Twitter em relação ao Carrefour?”.

Requisitos

1. Ter uma conta no Twitter (<https://twitter.com/>).
2. Estar cadastrado como Twitter Developer (<https://developer.twitter.com/>).
3. Acesso a API do Twitter.
4. MongoDB instalado.

Pacotes Utilizados

- **dplyr**: Utilizado para manipulação de dados.
- **rtweet**: Faz a interface de conexão entre o R e a API do Twitter.
- **tm**: Possui inúmeras funções direcionadas a atividade de mineração de texto.
- **wordcloud**: Permite para criar uma nuvem de palavras.
- **syuzhet**: Utilizado para classificar os sentimentos. Ele disponibiliza algumas funções úteis para a identificação das emoções presentes em textos, entre elas, a função chamada **get_nrc_sentiment()** que usa um dicionário de termos, denominado de NRC Emotion Lexicon, no qual associa palavras à emoções e sentimentos, afim de realizar a comparação das palavras e identificar as emoções e sentimentos presentes no texto.
- **mongolite**: Faz a interface de conexão com MongoDB.

Carregando pacotes

```
library(dplyr)
library(rtweet)
library(tm)
library(wordcloud)
library(syuzhet)
library(mongolite)
```

Buscando Tweets com função `search_tweets()` do pacote `rtweet`

```
carrefour_tweets <- search_tweets(
  "#carrefour",
  include_rts = FALSE
)
```

Quantidade de Tweets

```
nrow(carrefour_tweets)
```

```
## [1] 100
```

Estabeleço uma conexão local com MongoDB - Base de dados Carrefour - Collection - carrefour_tweets

```
vConexao <- mongo(  
  collection = "carrefour_tweets",  
  db = "carrefour",  
  url = "mongodb://localhost",  
  verbose = FALSE,  
  options = ssl_options()  
)
```

Armazendo os Tweets obtidos no MongoDB

```
vConexao$insert(carrefour_tweets)
```

```
## List of 5  
## $ nInserted : num 100  
## $ nMatched : num 0  
## $ nRemoved : num 0  
## $ nUpserted : num 0  
## $ writeErrors: list()
```

Menor data

```
min(carrefour_tweets$created_at)
```

```
## [1] "2021-09-24 05:00:00 UTC"
```

Maior data

```
max(carrefour_tweets$created_at)
```

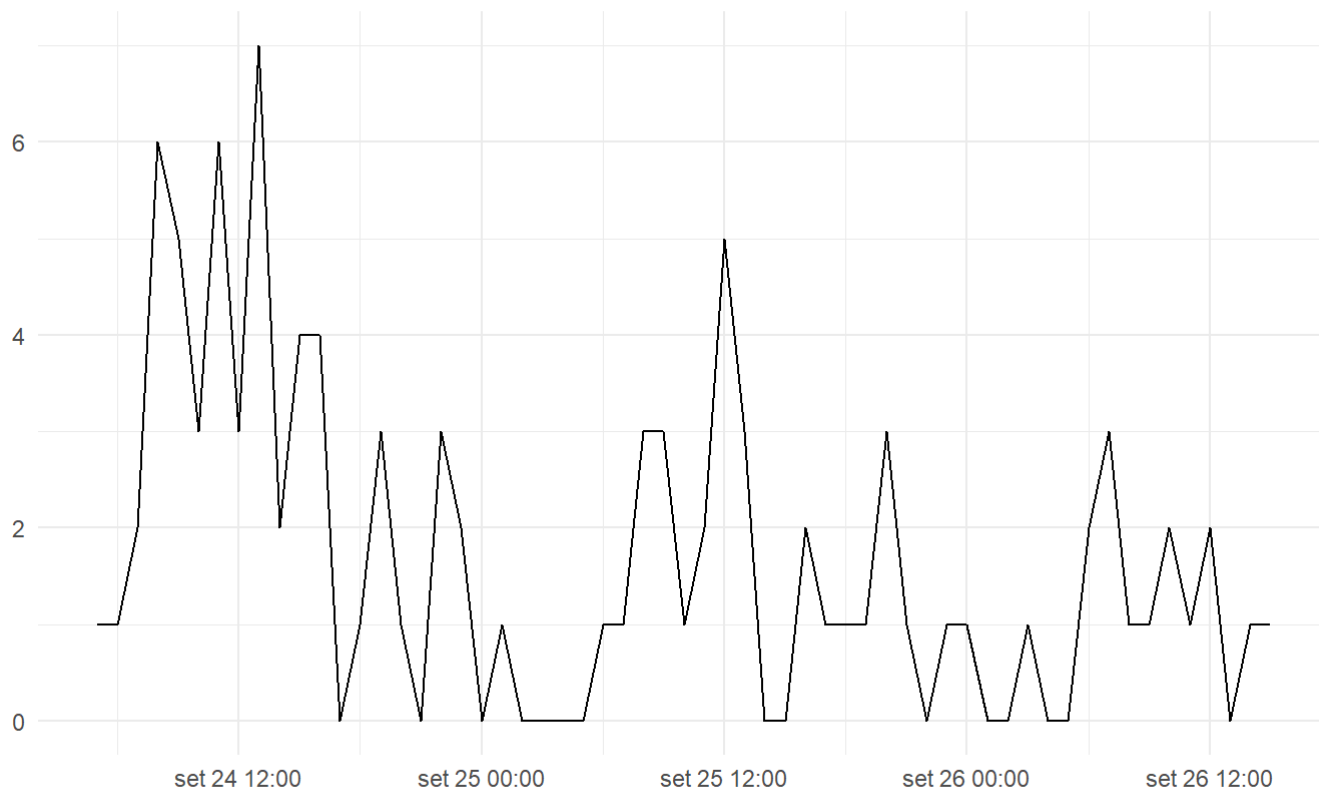
```
## [1] "2021-09-26 15:00:37 UTC"
```

Visualizando a série temporal de frequência dos tweets no decorrer do tempo usando a função ts_plot()

```
ts_plot(carrefour_tweets, "hour") +  
theme_minimal() +  
theme(plot.title = ggplot2::element_text (face = "bold")) +  
labs (  
  x = NULL, y = NULL,  
  title = "Frequência do uso da hashtag #carrefour nas ultimas horas",  
  subtitle = "Contagem de tweets agrupados em intervalos de horas",  
  caption = "\nFonte: Dados coletados do Twitter"  
)
```

Frequência do uso da hashtag #carrefour nas últimas horas

Contagem de tweets agrupados em intervalos de horas



Fonte: Dados coletados do Twitter

Separando apenas a coluna de Tweets do DataFrame obtido pelo rtweet

```
carrefour_text <- carrefour_tweets$text
```

Para fazer a limpeza dos textos podemos utilizar as funções do pacote tm, ou podemos criar as nossas próprias funções

```
# Função para limpeza dos textos
limpar_texto <- function(texto) {
  # Convertendo o texto para minúsculo
  texto <- tolower(texto)
  # Removendo o usuário adicionado no comentário
  texto <- gsub("@\\w+", "", texto)
  # Removendo as pontuações
  texto <- gsub("[[:punct:]]", "", texto)
  # Removendo links
  texto <- gsub("http\\w+", "", texto)
  # Removendo tabs
  texto <- gsub("[ \\t]{2,}", "", texto)
  # Removendo espaços no início do texto
  texto <- gsub("^ ", "", texto)
  # Removendo espaços no final do texto
  texto <- gsub(" $", "", texto)
  return(texto)
}
```

Executando a função de limpeza de dados

```
carrefour_text <- limpar_texto(carrefour_text)
```

Convertendo os textos em corpus.

O Corpus, são uma coleção de documentos criada pelo R.

```
carrefour_corpus <- VCorpus(VectorSource(carrefour_text))
```

Removendo Stopwords.

Stopwords são palavras que não tenham valor semântico, geralmente são palavras conectivas (com, para, e, a).

```
carrefour_corpus %>% tm_map(removeWords, stopwords("portuguese"))  
carrefour_corpus %>% tm_map(removeWords, stopwords("french"))  
carrefour_corpus %>% tm_map(removeWords, stopwords("english"))
```

Através de uma Wordcloud podemos visualizar os termos mais frequentes no conjunto de dados

```
wordcloud(  
  carrefour_corpus,  
  min.freq = 5,  
  max.words = 30,  
  random.order = F,  
  colors = brewer.pal(8, "Dark2")  
)
```



Agora transformaremos o corpus em uma matriz de documentos-termos para criarmos um gráfico de barras com os termos e sua frequência.

```
# Transformando o corpus em matriz de documentos-termos
carrefour_doc <- DocumentTermMatrix(carrefour_corpus)

# Removendo os termos menos frequentes
carrefour_doc1 <- removeSparseTerms(carrefour_doc, 0.97)

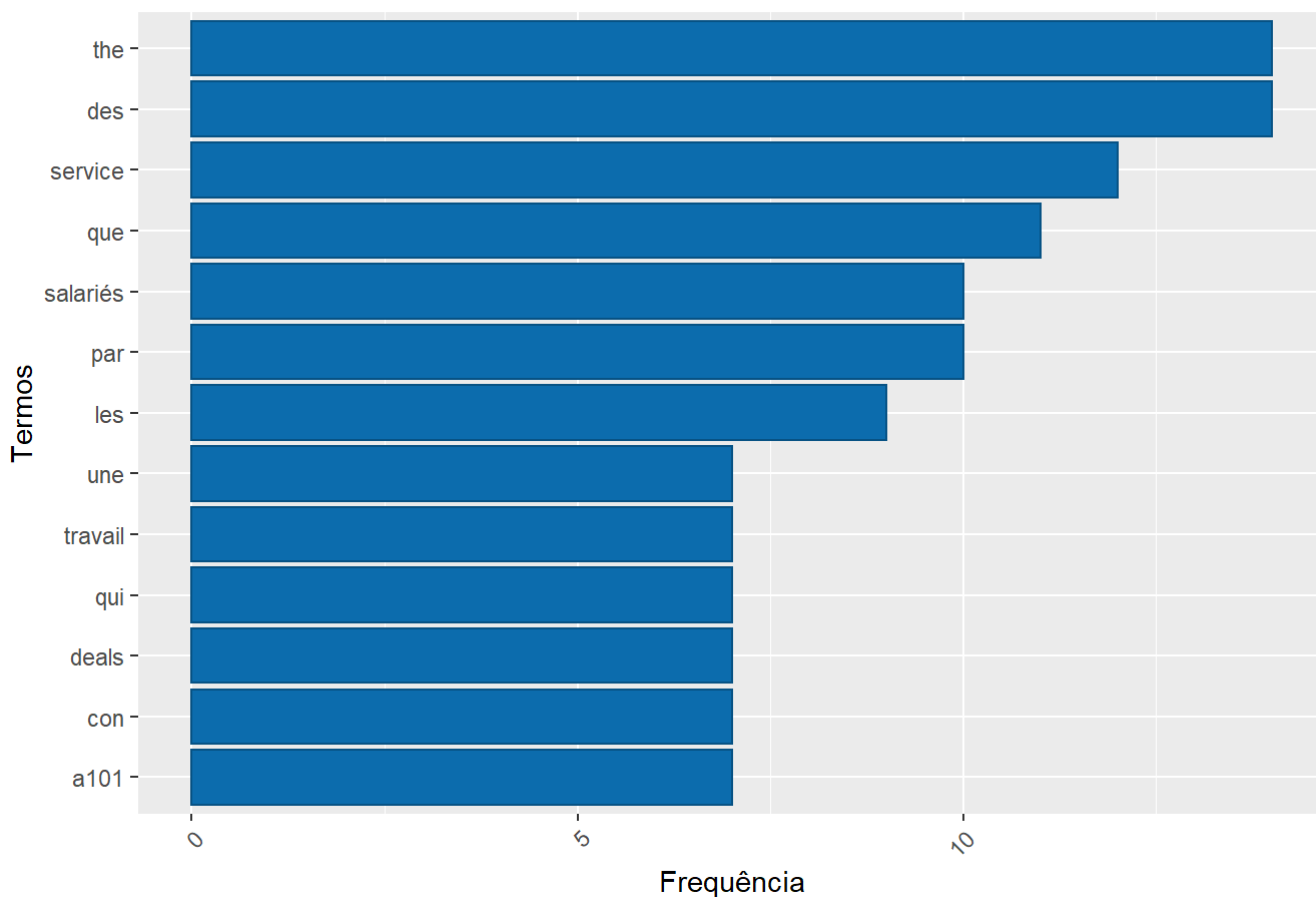
# Gerando uma matrix ordenada, com o termos mais frequentes
carrefour_freq <-
  carrefour_doc1 %>%
  as.matrix() %>%
  colSums() %>%
  sort(decreasing = T)

# Criando um dataframe com as palavras mais frequentes
df_carrefour_freq <- data.frame(
  word = names(carrefour_freq),
  freq = carrefour_freq)

# Gerando um gráfico da frequência

df_carrefour_freq %>%
  filter(!word %in% c("carrefour")) %>%
  subset(freq > 6) %>%
  ggplot(aes(x = reorder(word, freq), y = freq)) +
  geom_bar(stat = "identity", fill = "#0c6cad", color = "#075284") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Termos relacionados ao Carrefour mais frequentes no Twitter") +
  labs(y = "Frequência", x = "Termos") +
  coord_flip()
```

Termos relacionados ao Carrefour mais frequentes no Twitter



Realizando a análise de sentimentos dos tweets.

Para isso será utilizado a função **get_nrc_sentiment()** do pacote **syuzhet** e passando como parâmetro os termos da matriz de documentos-termos. Após a obtenção das emoções dos termos, será o cálculo da frequência dos sentimentos que utilizaram a **#carrefour**.

```
# Obtendo os emoções
carrefour_sentimento <- get_nrc_sentiment(
  carrefour_doc$dimnames$Terms,
)

# Calculando a frequência dos sentimentos
carrefour_sentimento_freq <- carrefour_sentimento %>%
  colSums() %>%
  sort(decreasing = T)
```

Criando um dataframe com os sentimentos traduzidos, que será utilizado como conversão de do minio.

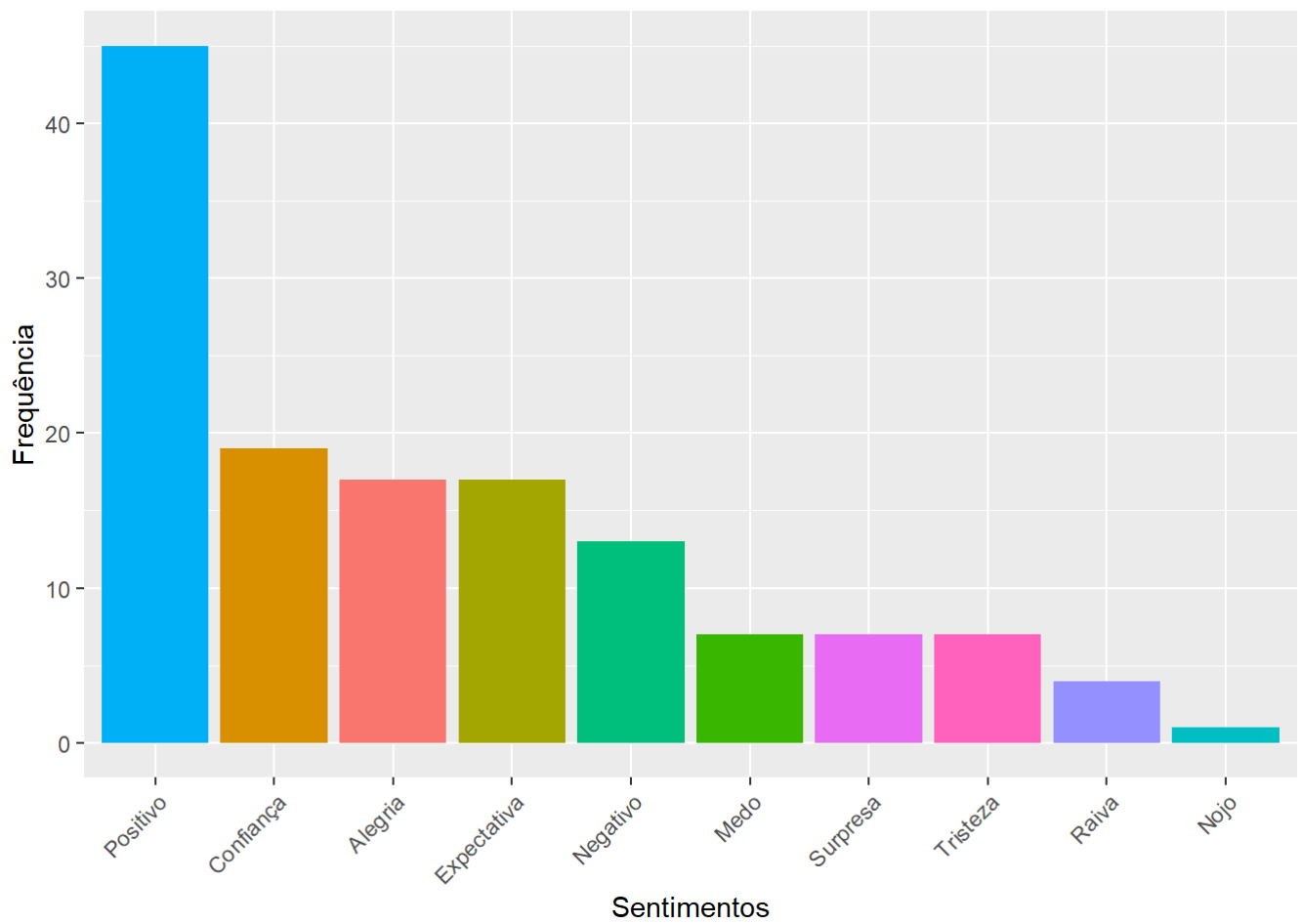
```
sentimetos_traducao <-  
  data.frame(  
    sentiment = c(  
      "positive",  
      "negative",  
      "trust",  
      "anticipation",  
      "fear",  
      "joy",  
      "sadness",  
      "surprise",  
      "anger",  
      "disgust"  
    ),  
    sentimentos = c(  
      "Positivo",  
      "Negativo",  
      "Confiança",  
      "Expectativa",  
      "Medo",  
      "Alegria",  
      "Tristeza",  
      "Surpresa",  
      "Raiva",  
      "Nojo"  
    )  
  )  
)
```

Transformando os resultados da frequência em um dataframe e juntando ao dataframe de tradução

```
df_sentimento <-  
  data.frame(  
    sentiment = names(carrefour_sentimento_freq),  
    freq = carrefour_sentimento_freq  
  ) %>%  
  left_join(sentimetos_traducao, by = "sentiment") %>%  
  dplyr::select(-sentiment) %>%  
  arrange(desc(freq))
```

Visualizando a frequência dos sentimentos em relação ao #Carrefour

```
ggplot(data = df_sentimento,  
       aes(x = reorder(sentimentos, -freq), y = freq)) +  
  geom_bar(aes(fill=sentimentos), stat = "identity") +  
  theme(legend.position = "none",  
        axis.text.x = element_text(angle = 45, hjust = 1)) +  
  xlab("Sentimentos") +  
  ylab("Frequência")
```



Conclusão

Podemos notar que