Quanteda and Twitter

Rodrigo Esteves de Lima-Lopes
State University of Campinas
rll307@unicamp.br

Contents

1 Introduction

Quanteda is a package for managing and analyse text quantitatively. It is quite easy to use and will bring us a number of interesting functions.

1.1 You will need:

1. The package `Quanteda`, `quanteda.textplots` and `quanteda.textstats`, which can be installed using RStudio
2. The package `rtweet`, we installed last tutorial.
3. Package `DT` for viewing the KWIC inside R.
4. `Ggplot2` for viewing some graphics

1.2 Data

We are going to use the same data we have used in the previous tutorials

2 Making some comparissions

First we need to save our texts as vectors

```
LI <- subset(presidentes, screen_name == "LulaOficial")
JB <- subset(presidentes, screen_name != "LulaOficial")
```

Kwic in the terminal

```
kwic(JB.t,"Brasil") |>
  DT::datatable()
```

Kwic in the Viewer (it views but does not save)

```
kwic(LI.t,"Brasil") |>
  DT::datatable()
```

Now lets keep on our analysis. We can use the same DFMs we created before, but I will create a couple just to apply some new commands

First step now is to make all lower caps:

```
LI.lower <- char_tolower(LI.t)
JB.lower <- char_tolower(JB.t)
```

Now, we create a lower character vector

```
boulos.word.v <- tokens(boulos.lower.v,
                        remove_punct = TRUE)
covas.word.v <- tokens(covas.lower.v,
                       remove_punct = TRUE)
```

So now, we have our second DFM:

```
JB.dfm <- dfm(JB.tok,
              verbose = TRUE) |>
  dfm_remove(stopwords("portuguese"),
             verbose = TRUE)

LI.dfm <- dfm(LI.tok,
              verbose = TRUE) |>
  dfm_remove(stopwords("portuguese"),
             verbose = TRUE)
```

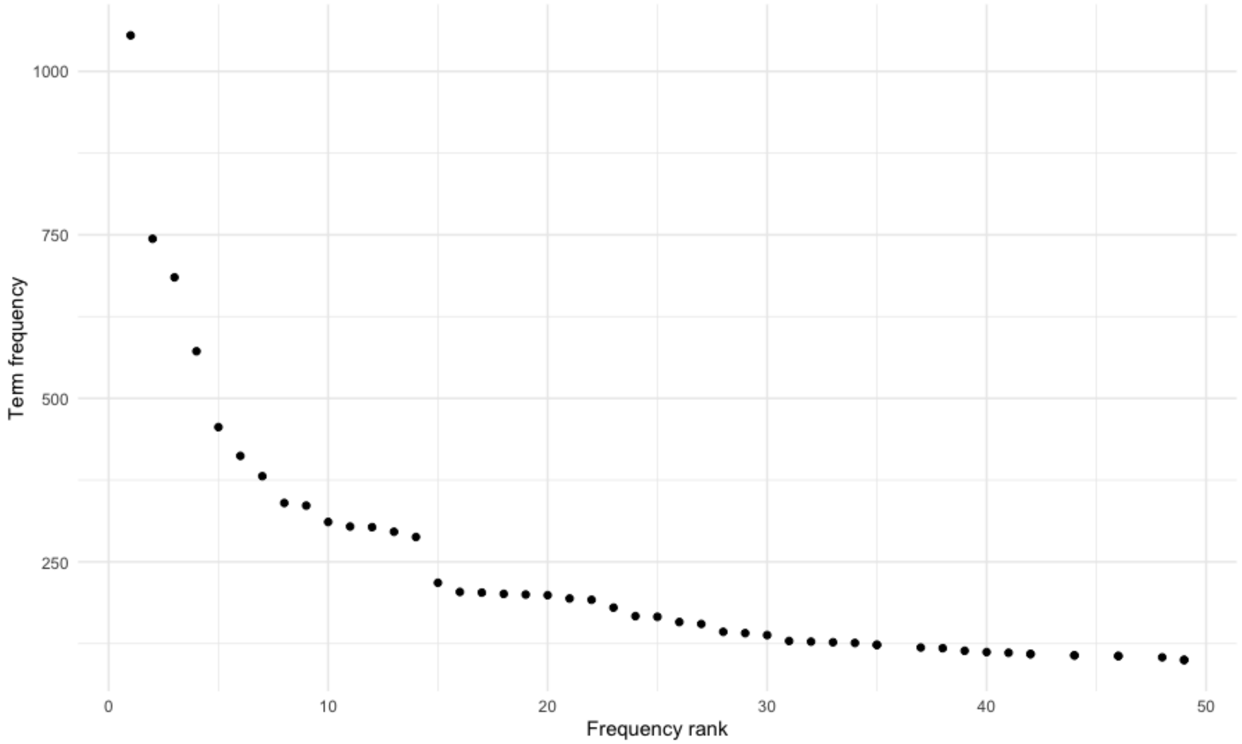
Now we are creating our word list

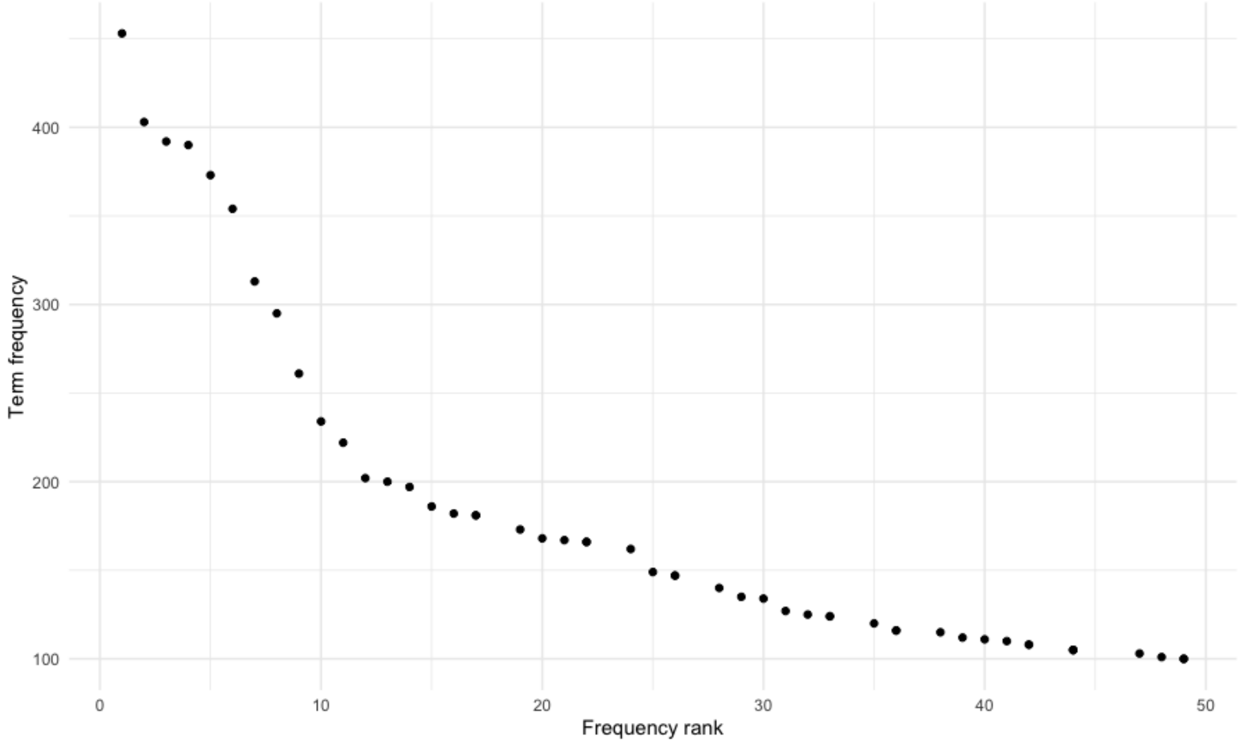
```
LI.wl <- textstat_frequency(LI.dfm)
JB.wl <- textstat_frequency(JB.dfm)
View(JB.wl)
View(LI.wl)
```

Lets us plot a single candidate at a time:

```
library(ggplot2)
theme_set(theme_minimal())
textstat_frequency(LI.dfm, n = 50) %>%
  ggplot(aes(x = rank, y = frequency)) +
  geom_point() +
  labs(x = "Frequency rank", y = "Term frequency")
```

```
library(ggplot2)
theme_set(theme_minimal())
textstat_frequency(JB.dfm, n = 50) %>%
  ggplot(aes(x = rank, y = frequency)) +
  geom_point() +
  labs(x = "Frequency rank", y = "Term frequency")
```





Quanteda makes key wordlists. Keywords are a comparison between two groups of texts. One is the reference and tells me the statistical baseline for comparison, the other is my research text (or texts). Here, for the sake of the exercise, we will compare Jair Bolsonaro and Lula.

```
corpus.all <- corpus(presidentes, text_field = 'text', docid_field = 'status_id')
tokens.all <- tokens(corpus.all, remove_punct = TRUE,
                     verbose = TRUE) |>
```

```
  tokens_group(groups = screen_name)
dfm.all <- dfm(tokens.all,
               verbose = TRUE) |>
  dfm_remove(stopwords("portuguese"),
             verbose = TRUE)
dfm.all <- dfm_group(dfm.all, groups = "screen_name")

textstat_keyness(dfm.all, target = "LulaOficial") |>
  textplot_keyness(n= 25)
```

The result is something like

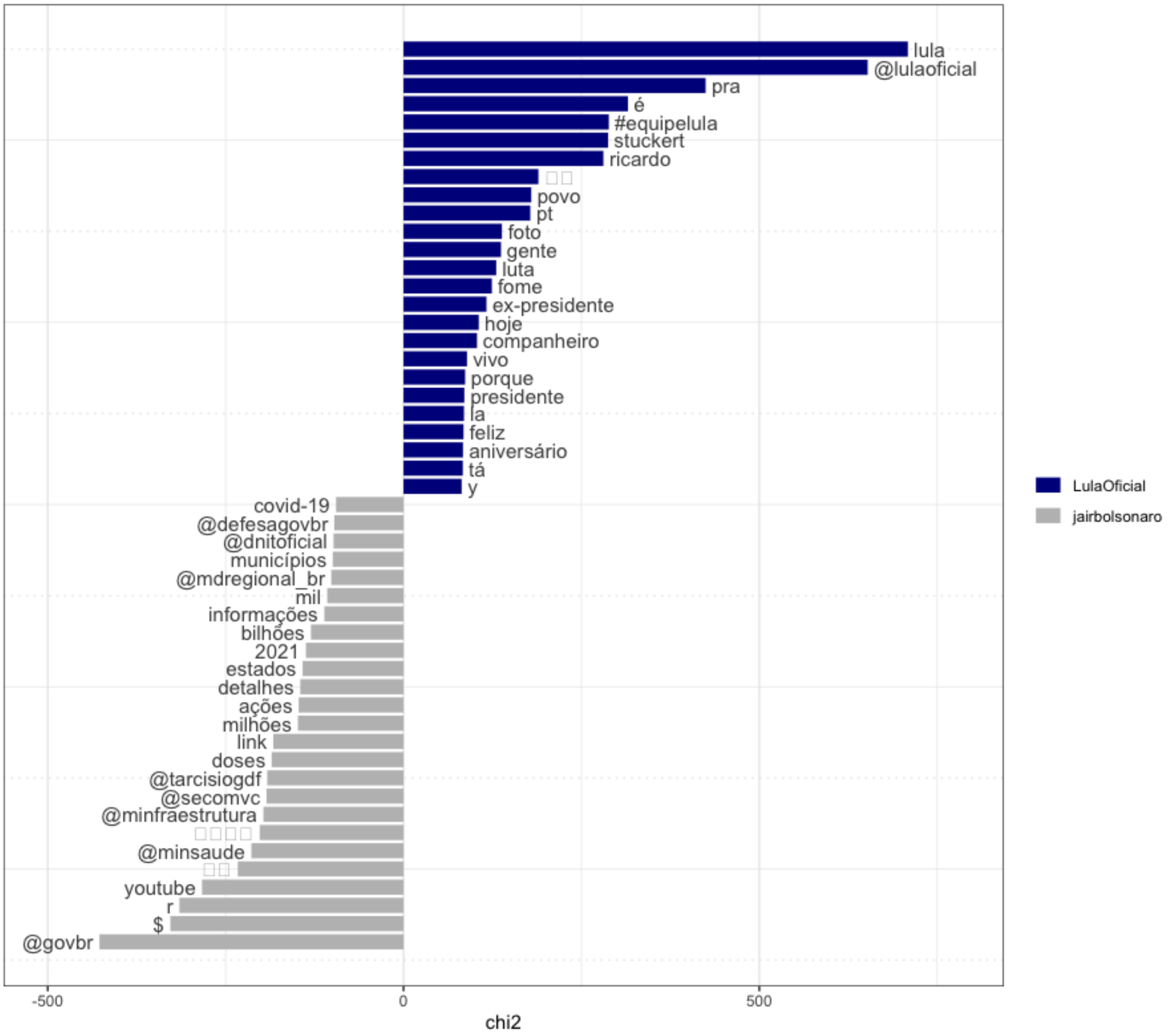


Figure 1: Keywords comparisson

3 Some collocates

The command `textstat_collocations` calculates n-grams based on a corpus previously created. Unfortunately, for the time being the only calculation possible is `lambda`.

```
tri.grams<-textstat_collocations(
  LI.tok,
  method = "lambda",
  size = 3,
  min_count = 5,
  smoothing = 0.5,
  tolower = TRUE)
```