Quanteda and Twitter

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

Contents

1 Introduction **1**
1.1 You will need: 1

2 Scraping Tweets **1**

3 Doing some analysis **2**
3.1 Creating the corpora 2
3.2 Creating a network of hashtags for each candidate 2
3.3 Analysing some hashtags 3
3.4 New ways to compare 5
3.5 Analysing user interactions 10

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`, which can be installed using RStudio
2. The package `rtweet`, we installed last tutorial.
3. Package `DT` for viewing the KWIC inside R.

2 Scraping Tweets

I will download two Twitter timelines: GuilhermeBoulos and brunocovas. Both are candidates in the second round of São Paulo's mayor elections.

```
library(rtweet)
covas <- get_timelines("brunocovas", n = 3200)
boulos <- get_timelines("GuilhermeBoulos", n = 3200)
boulos_and_covas <-rbind(covas,boulos)
```

If you want to download the same data I used in this tutorial, there is a image saved on data/quanteda directory.

3 Doing some analysis

3.1 Creating the corpora

We are now creating three corpora:

1. Boulos's Tweets
2. Cova's Tweets
3. All together

```
boulos.corpus<-corpus(boulos)
covas.corpus<-corpus(covas)
all.corpora<-corpus(boulos_and_covas)
```

3.2 Creating a network of hashtags for each candidate

```
boulos.dfm<-dfm(boulos.corpus,
                remove_punct = TRUE,
                case_insensitive=TRUE,
                remove = stopwords("portuguese"),verbose = TRUE)
covas.dfm<-dfm(covas.corpus,
               remove_punct = TRUE,
               case_insensitive=TRUE,
               remove = stopwords("portuguese"),
               verbose = TRUE)
all.dfm<-dfm(all.corpora,
             remove_punct = TRUE,
             case_insensitive=TRUE,
             remove = stopwords("portuguese"),
             verbose = TRUE)
```

```
head(boulos.dfm,5)
```

```
## Loading required package: quanteda
## Package version: 2.1.2
## Parallel computing: 2 of 4 threads used.
## See https://quanteda.io for tutorials and examples.
##
## Attaching package: 'quanteda'
## The following object is masked from 'package:utils':
##
##     View
## Document-feature matrix of: 5 documents, 11,372 features (99.8% sparse) and 89 docvars.
##        features
## docs    viola catarina rossi violões gustavo medeiros arranjo sopros sérgio
##   text1     1        1     1       1       1        1       1      1      1
##   text2     0        0     0       0       0        0       0      0      1
##   text3     0        0     0       0       0        0       0      0      0
##   text4     0        0     0       0       1        1       0      0      0
##   text5     0        0     0       0       0        0       0      0      0
##        features
## docs    wontroba
```

```
##   text1        1
##   text2        1
##   text3        0
##   text4        0
##   text5        0
## [ reached max_nfeat ... 11,362 more features ]
```

```
head(all.dfm,5)
```

```
## Document-feature matrix of: 5 documents, 16,944 features (99.9% sparse) and 89 docvars.
##        features
## docs    legados importantes pandemia é valorização ciência fundamental apoiar
##   text1       1           1        1 2           1       1           1      1
##   text2       0           0        0 1           0       0           0      0
##   text3       0           0        0 1           0       0           0      0
##   text4       0           0        0 0           0       0           0      0
##   text5       0           0        0 1           0       0           0      0
##        features
## docs    investir instituições
##   text1        1            1
##   text2        0            0
##   text3        0            0
##   text4        0            0
##   text5        0            0
## [ reached max_nfeat ... 16,934 more features ]
```

3.3 Analysing some hashtags

Frist we will do the magic for Guilherme Boulos. We are going to:

1. Select the hashtags using the command `dfm_select`
2. Select the 50 more frequent using `topfeatures` command

```
tag.dfm.boulos <- dfm_select(boulos.dfm, pattern = ("#*"))
toptag.boulos <- names(topfeatures(tag.dfm.boulos, 50))
```

Let us see the result:

```
head(toptag.boulos)
```

```
## [1] "#virasp50"          "#boulos50"          "#viradacomboulos50"
## [4] "#bouloserundina50"  "#boulosnaband"      "#debatenaband"
```

Now let us construct a feature-occurrence matrix for the hashtags

```
tag_fcm.boulos <- fcm(tag.dfm.boulos)
```

Now let us see it:

```
head(tag_fcm.boulos)
```

```
## Feature co-occurrence matrix of: 6 by 6 features.
##                            features
## features                    #virasp #boulos50 #virasp50 #viradailustrada50
##   #virasp                         0         3         0                  0
##   #boulos50                       0         0        99                  1
##   #virasp50                       0         0         1                  1
##   #viradailustrada50              0         0         0                  0
##   #mulheresnocorrecomboulos       0         0         0                  0
```

```
##   #quemtemmedodossemteto          0         0         0                   0
##                             features
## features                    #mulheresnocorrecomboulos #quemtemmedodossemteto
##   #virasp                                           0                     0
##   #boulos50                                         1                     1
##   #virasp50                                         1                     1
##   #viradailustrada50                                1                     0
##   #mulheresnocorrecomboulos                         0                     0
##   #quemtemmedodossemteto                            0                     0
```

First let us make a FCM only with the top hashtags

```
topgat_fcm.boulos <- fcm_select(tag_fcm.boulos, pattern = toptag.boulos )
```

And then we make our network

```
textplot_network(topgat_fcm.boulos,
                 min_freq = 0.1,
                 edge_alpha = 0.8,
                 edge_size = 5)
```

Let us see how is the final product

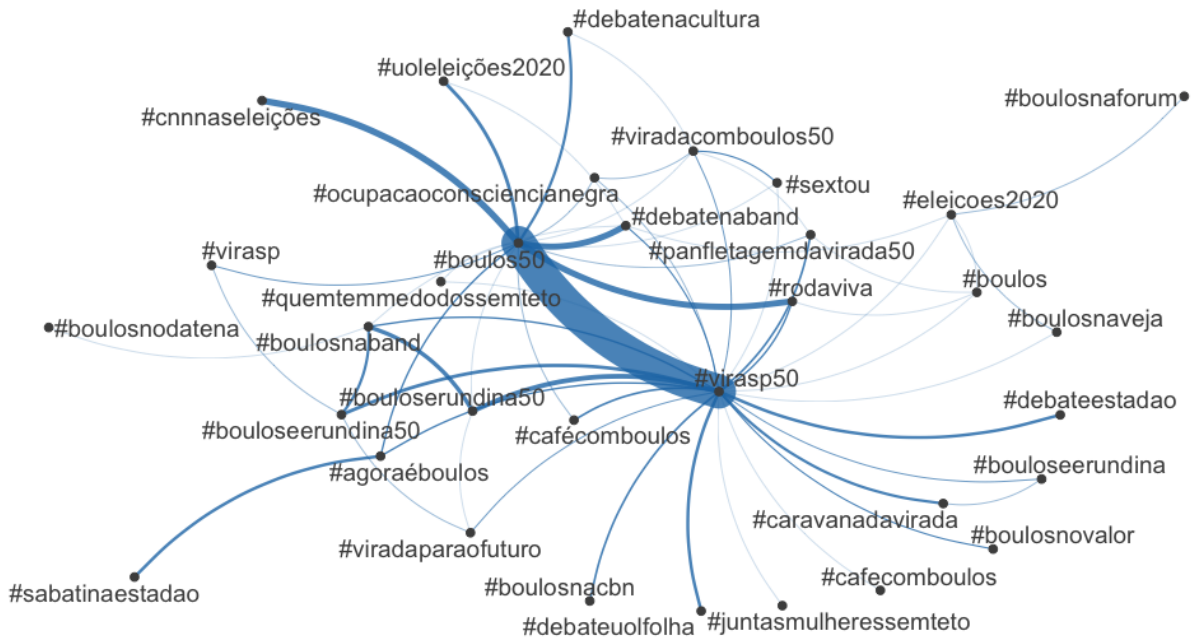


Figure 1: Bloulo's network of hashtags

Now let us see how it works for Covas, all in a single batch of commands:

```
tag.dfm.covas <- dfm_select(covas.dfm, pattern = ("#*"))
toptag.covas <- names(topfeatures(tag.dfm.covas, 50))
tag_fcm.covas <- fcm(tag.dfm.covas)
topgat_fcm.covas <- fcm_select(tag_fcm.covas, pattern = toptag.covas )
textplot_network(topgat_fcm.covas,
                 min_freq = 0.1,
                 edge_alpha = 0.8,
```

```
                edge_size = 5,
                edge_color = "orange")
```

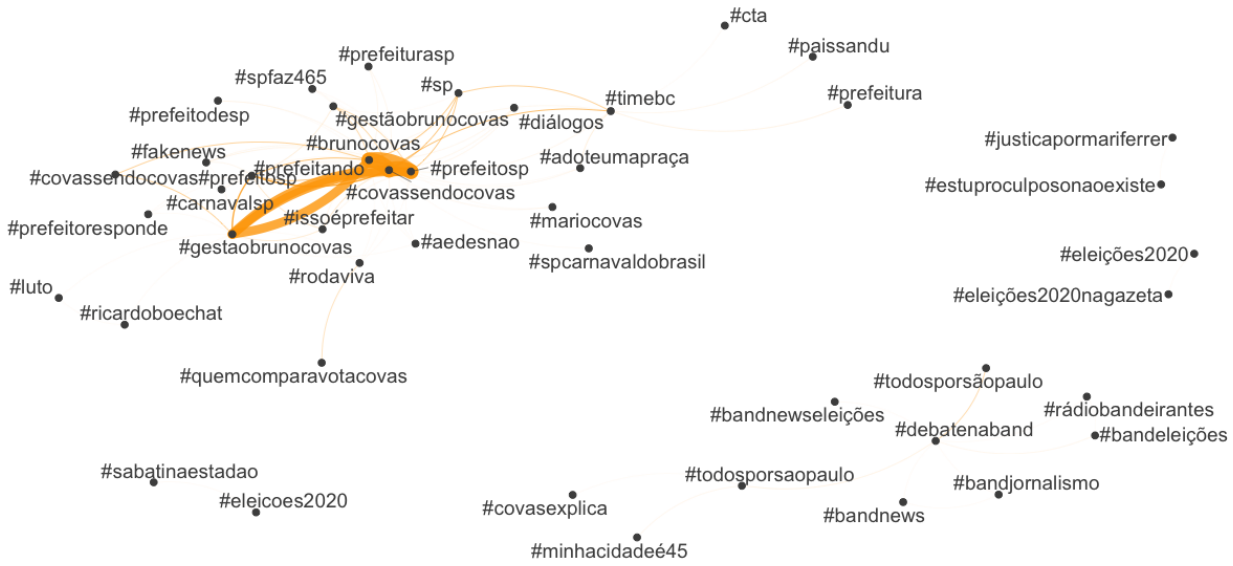


Figure 2: Covas' network of hashtags

Now let us do the two together. Again in a single script:

```
tag.dfm.all <- dfm_select(all.dfm, pattern = ("#*"))
toptag.all <- names(topfeatures(tag.dfm.all, 50))
tag_fcm.all <- fcm(tag.dfm.all)
topgat_fcm.all <- fcm_select(tag_fcm.all, pattern = toptag.all )
textplot_network(topgat_fcm.all, min_freq = 0.1,
                 edge_alpha = 1,
                 edge_size = 10,
                 edge_color = "green")
```

And the result is bellow

3.4 New ways to compare

Now let us make a new graphic. Here we are counting the importance of each hashtag.

```
tstat_freq <- textstat_frequency(tag.dfm.all, n = 15, groups = "screen_name")
```

Then we do some coding using `ggplot2`, so we can see how it looks like:

```
library(ggplot2)
tag.dfm.all %>%
  textstat_frequency(n = 15) %>%
  ggplot(aes(x = reorder(feature, frequency), y = frequency)) +
  geom_point() +
  coord_flip() +
  labs(x = NULL, y = "Frequency") +
  theme_minimal()
```

The expected result would be something similar to this:

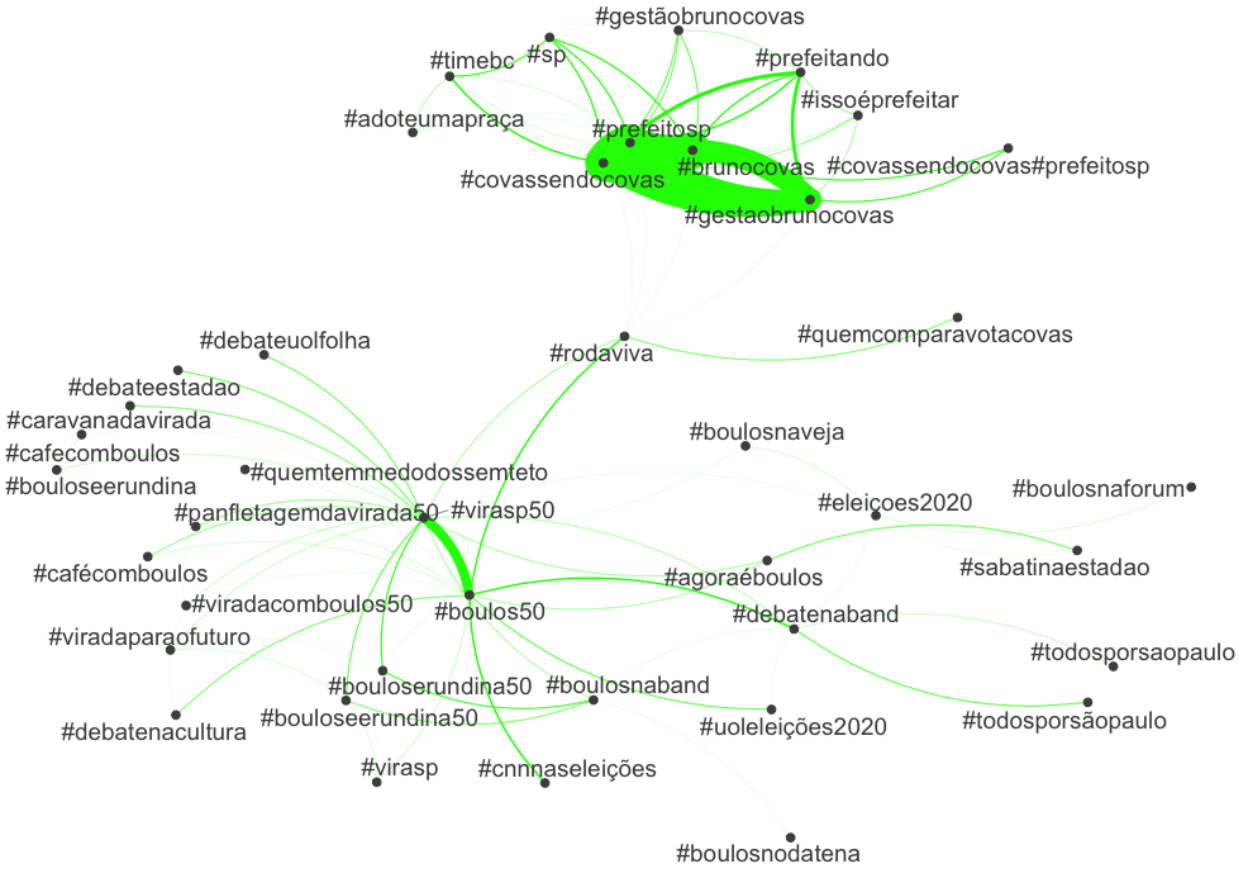


Figure 3: Network of hashtags for two candidates

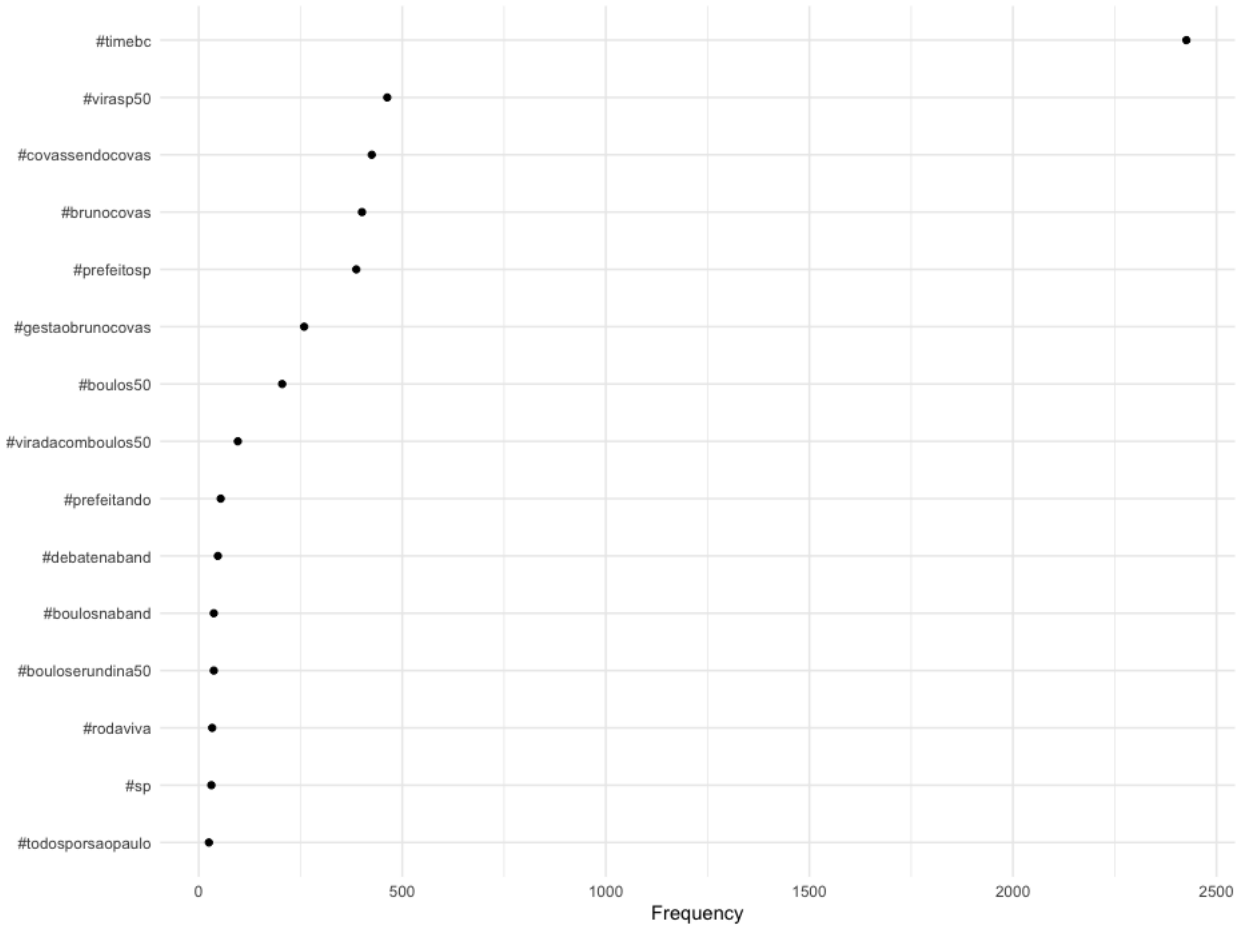


Figure 4: Hashtag plotting

Now, let us make a general cloud of hashtags:

```
set.seed(132)
textplot_wordcloud(tag.dfm.all, max_words = 100)
```

And the result should be somesthing similar to this:

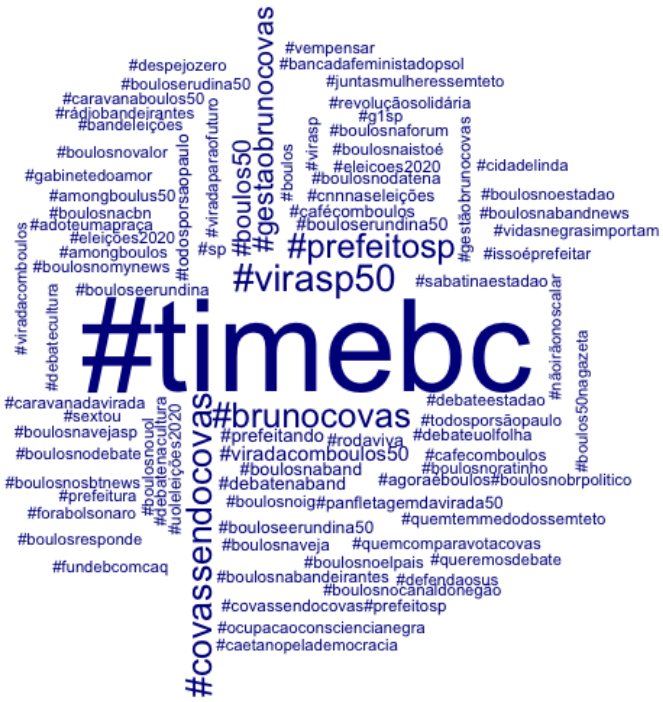


Figure 5: It is a cloud!

This code will make some comparison:

```
dfm.hash.all <- dfm(all.corpora, select = "#*", groups = "screen_name")
```

Now we plot it:

```
textplot_wordcloud(dfm.hash.all,
                   comparison = TRUE,
                   max_words = 200,
                   color = c("darkblue", "darkred"))
```

Figure 6: It is a cloud!

3.5 Analysing user interactions

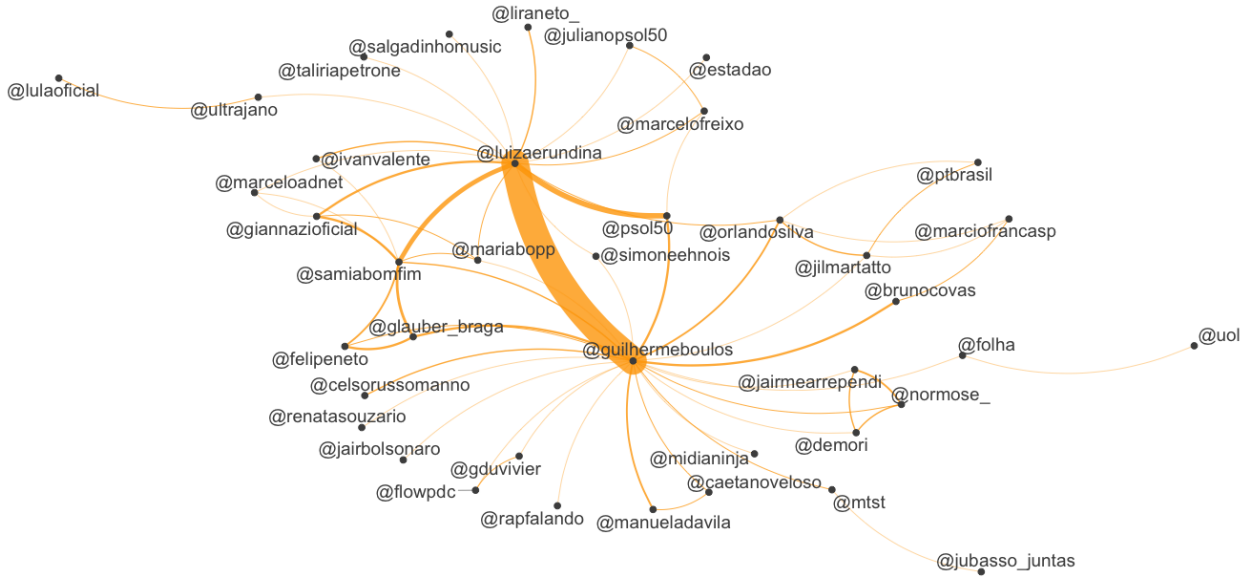
We can use the same methodology to study users interaction. The difference we are going to change the search for `*#` to `*@`. Let us start by Guilherme Boulos, but in a single command:

```
boulos.user.dfm <- dfm_select(boulos.dfm, pattern = "@*")
topuser.boulos <- names(topfeatures(boulos.user.dfm, 50))
View(topuser.boulos)

boulos.user.fcm <- fcm(boulos.user.dfm)
View(boulos.user.fcm)

boulos.user.plot <- fcm_select(boulos.user.fcm, pattern = topuser.boulos)
textplot_network(boulos.user.plot, min_freq = 0.1, edge_color = "orange", edge_alpha = 0.8, edge_size =
```

The result would be something similar to it:



Now let us do the same for Bruno Covas

```
covas.user.dfm <- dfm_select(covas.dfm, pattern = "@*")
topuser.covas <- names(topfeatures(covas.user.dfm, 50))
head(topuser.covas)

covas.user.fcm <- fcm(covas.user.dfm)
View(covas.user.fcm)

covas.user.plot <- fcm_select(covas.user.fcm, pattern = topuser.covas)
textplot_network(covas.user.plot,
                 min_freq = 0.1,
                 edge_color = "pink3",
                 edge_alpha = 0.8,
                 edge_size = 5)
```

The result should be something similar to:

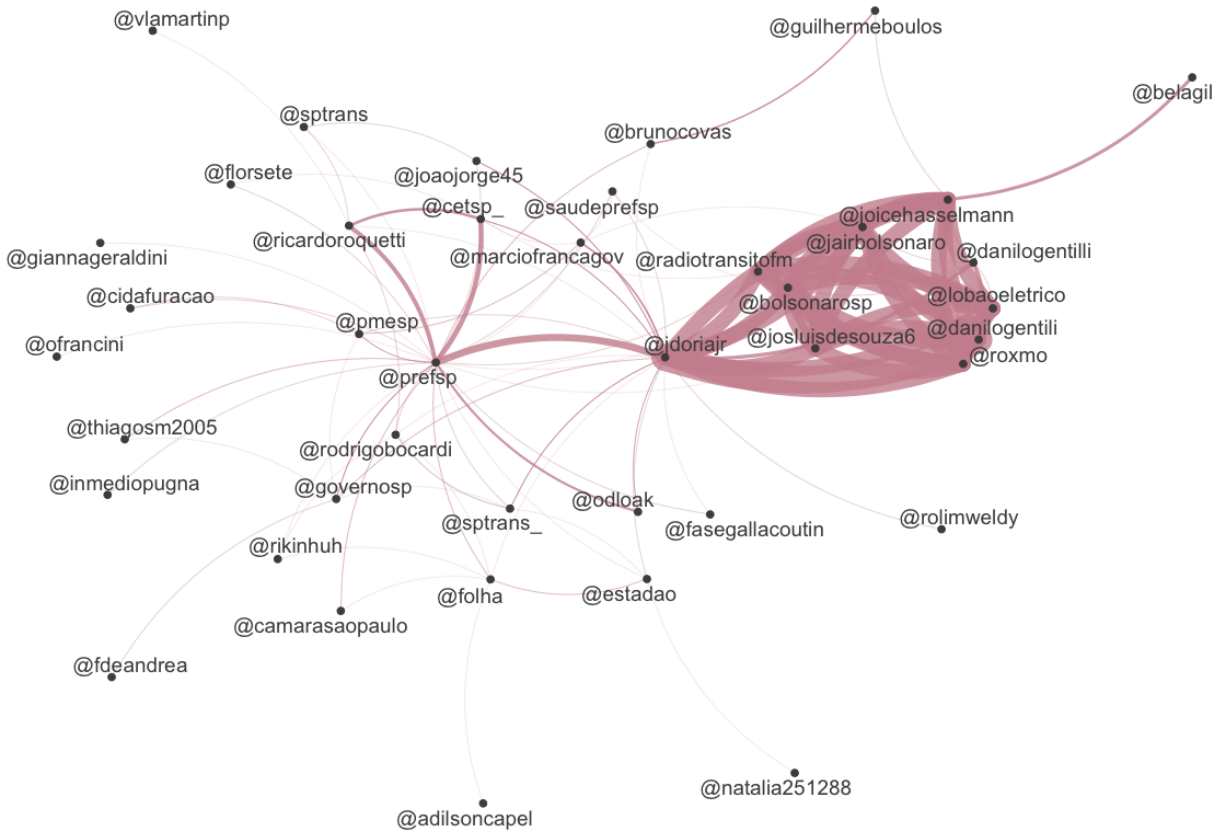


Figure 7: Covas' users