Facebook Lab

Paine (HE Peiwei, 54471952)

Research Question: How people react to an illegal migrant issue on FB.

First, authorize by using the Facebook token.

```
library(Rfacebook)
library(rmarkdown)
```

```
fb_app_id = "767570183395387"
fb_app_secret = "539a32d8923e44edfd3c7aa750dc14d6"
token = fbOAuth(fb_app_id, fb_app_secret)
```

```
token = readRDS("token.rds")
```

Second, get 20 posts on the page of CNN.And create a subset of the posts dataframe, including the ids, time, likes, messages, and the numbers of likes, comments and shares. By checking the id number to select the post I want.

```
posts = getPage("cnn", token, n=20)
```

```
## 20 posts
```

```
posts = subset(posts, select=c("id", "created_time", "likes_count", "comments_count
", "shares_count", "message"))
```

Third, the post I choose is about illegally crossing the border into Canada. To get this post, just check the post id, which is [6] in the dataframe. After I get this post, I collect the reactions and comments data, and create dataframes seperately. Present the reactions on this post by pie chart, using plotly.

```
post = getPost(posts$id[7], token, reactions = T, comments=F)
reactions = post$reactions
r_counts=table(reactions$from_type)

post = getPost(posts$id[7], token, comments = T, likes = F)
comments = post$comments
save(comments, file="fbcomments.rda")
```

```
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```
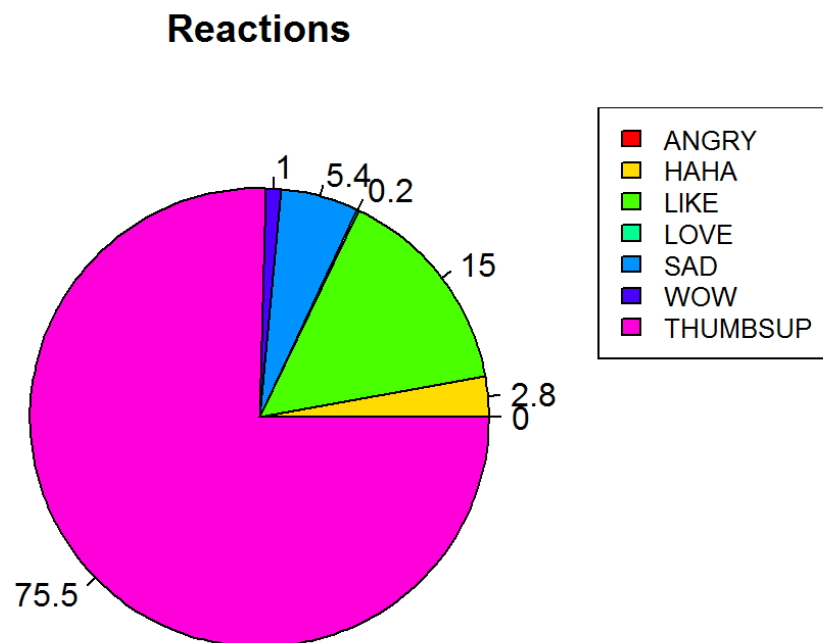
```
## The following object is masked from 'package:ggplot2':
##
##      last_plot
```

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

```
## The following object is masked from 'package:stats':
##
##      filter
```

```
## The following object is masked from 'package:graphics':
##
##      layout
```

```
x = c(1, 58, 305, 4,111,21,1537)
labels = c("ANGRY", "HAHA", "LIKE", "LOVE","SAD","WOW","THUMBSUP")
piepercent = round(100*x/sum(x), 1)
pie(x, labels = piepercent, main = "Reactions",col = rainbow(length(x)))
legend("topright",c("ANGRY", "HAHA", "LIKE", "LOVE","SAD","WOW","THUMBSUP"), cex =
0.8,
   fill = rainbow(length(x)))
```


Fifth, create a wordcloud of all the comments on this event. As the cloud shows, it mainly concerns about Canadian people.

```r
library(quanteda)
```

```
## quanteda version 0.9.9.3
```

```
##
## Attaching package: 'quanteda'
```

```
## The following object is masked from 'package:utils':
##
##     View
```

```
## The following object is masked from 'package:base':
##
##     sample
```

```r
u_corpus = corpus(comments$message)
u_dfm = dfm(u_corpus)
u_dfm
```

```
## Document-feature matrix of: 77 documents, 622 features (97.6% sparse).
```

```r
library(RColorBrewer)
stopwords = c(stopwords("english"), 'a','"', ',','&','the',"?","-","[","]","(",")",
"cnn","https", "rt", "news","who","you","this","too","for","in","by","http","will"
,"and","has","to","don't","/",":",".","bye","!","let",",",'"',"=")
u_dfm = dfm_select(u_dfm, stopwords,selection=c("remove"),valuetype=c("fixed"))
```

```
## removed 106 features, from 196 supplied (fixed) feature types
```

```r
textplot_wordcloud(u_dfm, max.words = 50, colors = brewer.pal(9, "Blues")[5:9], sc
ale = c(9, .2))
```

Sixth, get the replies to the comments, so as to construct a connection between the users. And get the second layer of the connection, by catching people who reply to the comments.

```
post = getPost("5550296508_10156108813016509", token, likes=F)
comments = post$comments
replies = list()
for (comment in comments$id[comments$comments_count > 0]) {
creplies = getCommentReplies(comment, token)$replies
if (nrow(creplies) > 0) {
creplies$comment_id = comment
replies = c(replies, list(creplies))
}
}
replies = plyr::rbind.fill(replies)
```

```
replies2 = replies[c("from_name", "comment_id")]
comments2 = comments[c("id", "from_name")]
colnames(comments2) = c("comment_id", "to_name")
replies2 = merge(replies2, comments2)
replies2 = aggregate(replies2$comment_id, replies2[c("from_name", "to_name")], leng
th)
replies2 = subset(replies2, from_name != to_name)
```

Finally, draw a graph according to the connections.But the nodes on the graph are too crowded, so I only keep the people with over 2 comments on the connection graph. There are 5 main colors on the graph, which mean there are 5 main debates among the comments under this post.

```r
library(igraph)
```

```
## 
## Attaching package: 'igraph'
```

```
## The following object is masked from 'package:quanteda':
## 
##     similarity
```

```
## The following objects are masked from 'package:plotly':
## 
##     %>%, groups
```

```
## The following objects are masked from 'package:stats':
## 
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
## 
##     union
```

```r
g = graph_from_data_frame(replies2, directed=F)
E(g)$weight = E(g)$x
V(g)$size = 0.5 + (.5 * degree(g))
clusters = edge.betweenness.community(g)$membership
pal = substr(rainbow(length(unique(clusters)), start=0.33, end=1, v=0.4), 1, 7)
V(g)$color = pal[match(clusters, unique(clusters))]
plot(g, vertex.label=NA,edge.arrow.size=1)
```
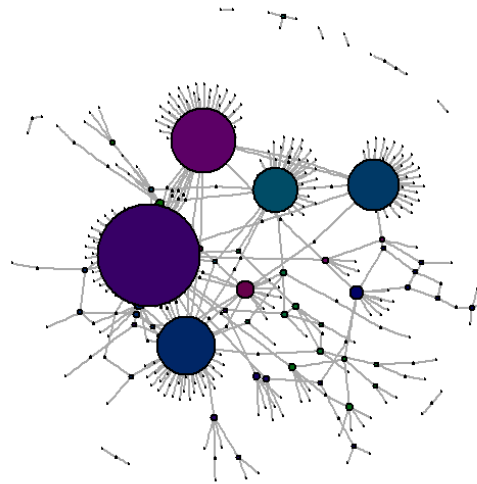
```r
# Keep only people with >2 reply in largest component
g2 = igraph::decompose(g, min.vertices = 50)[[1]]
g2 =induced_subgraph(g2, degree(g2, V(g2), "in")>2)
# Label size based on betweenness centrality
centrality = betweenness(g2)
V(g2)$label.cex = 0.5 + 0.5 * centrality / max(centrality)
# color labels based on clustering
clusters = edge.betweenness.community(g2)$membership
pal = substr(rainbow(length(unique(clusters)), start=0.33, end=1, v=0.5), 1, 7)
V(g2)$label.color = pal[match(clusters, unique(clusters))]
layout = layout.reingold.tilford(g2, circular=T)
plot(g2, vertex.shape = "none", layout=layout, edge.arrow.size=4, edge.curved=TRUE
)
```

Loading [Contrib]/a11y/accessibility-menu.js