## twitter2

Paine (HE Peiwei, 54471952) 2017/2/13

## Get Twitter followers' network

First, set up for the API.

## [1] "Using direct authentication"

Second, get the user's followers and add the followers into a dataframe.

```
u = twitteR::getUser("nruigrok")
followers = u$getFollowers()
followers.df = plyr::ldply(u$getFollowers(), as.data.frame)
knitr::kable(head(followers.df))

friends = u$getFriendIDs()
followers.df = subset(followers.df, id %in% friends)
followers.df = plyr::arrange(followers.df, -followersCount)
ids = head(followers.df$id, 10)
```

Third, define a new function of getting the followers. And save the followers data in rds format.

```
get_followers = function(u) {
   message(u$screenName)
   f = as.numeric(u$getFollowerIDs())
   data.frame(leader=as.numeric(u$id), follower=f)
}
users = c(u, followers[ids])
users
```

Fourth, to solve the Twitter rate limit, use a if else loop, if the follower is in the rds file, then skip, if it is not in the rds file, add and write in the rds.

```
if (file.exists("followers.rds")) {
  followers=readRDS("followers.rds")
} else {
  message("Getting follower information")
  followers="the followers"
  saveRDS(followers,"followers.rds")
}
```

Fifth, get the connections and put the list into a dataframe.

saveRDS(followers, file = "followers.rds")

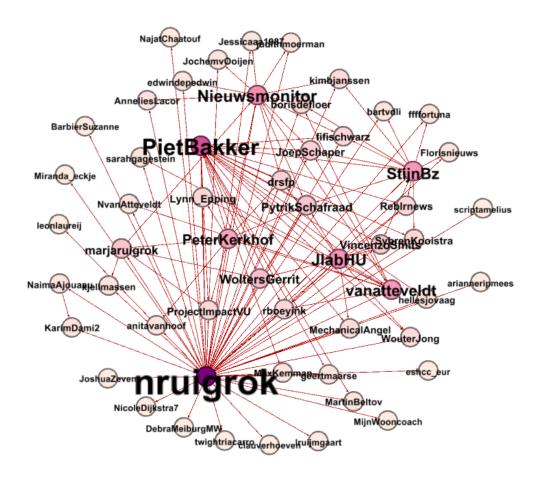
```
connections = plyr::ldply(users, get_followers, .id=NA)
followed_by_me = connections$follower[connections$leader == u$id]
connections = subset(connections, follower %in% followed_by_me)
connections = subset(connections, follower != leader)
nrow(connections)

userids = unique(c(connections$leader, connections$follower))
users = lookupUsers(userids)
users = plyr::ldply(users, as.data.frame)[c("id", "screenName")]
```

```
connections = merge(connections, users, by.x="leader", by.y="id")
connections = merge(connections, users, by.x="follower", by.y="id")
connections = connections[c("screenName.x", "screenName.y")]
head(connections)
```

Finally, I save the connections as csv file, and generate a new graph in Gephi as below. The label size is according to the degree distribution, and the node color is according to the clustering coefficient.

```
write.csv(connections, file = "D:\\Documents\\connections.csv", row.names = FALSE)
knitr::include_graphics("D:/Documents/network.png")
```



## Wordcloud

The user @nruigrok only has 9 tweets in total, so I change another user in the graph, who has the second highest degree. The user name is @PietBakker. I ask the api for 100 recent tweets, but only returns 69. By delete some stopwords, I generate a wordcloud, which seems to be relative to Journalism study.

```
u_tweets = plyr::ldply(searchTwitter("PietBakker", n=100), as.data.frame)

## Warning in doRppAPICall("search/tweets", n, params = params,
## retryOnRateLimit = retryOnRateLimit, : 100 tweets were requested but the
## API can only return 69
```

```
save(u_tweets, file="tweets.rda")

library(quanteda)
u_corpus = corpus(u_tweets$text)
u_dfm = dfm(u_corpus)
u_dfm

## Document-feature matrix of: 69 documents, 289 features (93.8% sparse).

library(RColorBrewer)
stopwords = c(stopwords("english"), 'a','"', ',','&','the',"?","-","[","]","(",")","coa","https", "rt",
u_dfm = dfm_select(u_dfm, stopwords,selection=c("remove"),valuetype=c("fixed"))
textplot_wordcloud(u_dfm, max.words = 70, colors = brewer.pal(9, "Reds")[5:9], scale = c(4, .5))
```

