Tweet Scraping in R (and Python)

Loren Collingwood

To harvest tweets, the best package in R is rtweet. However the twitter api R uses makes it difficult to harvest tweets (say from a particular hashtag) past 6-9 days. Therefore we will use both the rtweet package and the twint python package.

Step 1

Install and load rtweet. Look up the rtweet github account to get a better handle on the package's offerings.

```r
#install.packages("rtweet")
library(rtweet)
```

Step 2

Harvest the tweets using the search_tweets() function. There are some limitations here so you will need to investigate whether this option works for you.

```r
# searches hash tags last 6-9 days or so...
rstats <- search_tweets(q="Clicks4Kass"); dim(rstats)
```

```
## [1] 0 0
```

```r
# Search Kassra's timeline #
ko_time <- get_timeline("kassrao", n =3000)

# Dimensions #
dim(ko_time)
```

```
## [1] 1492   90
```

```r
# convert text to lower
tex <- data.frame(kass_text = tolower(ko_time$text))

# Create dummy indicator
tex$clicks <- ifelse(grepl("clicks4kass", tex$kass_text)==TRUE, 1, 0)

# Print to Console #
tex$kass_text[tex$clicks==1]
```

```
##  [1] "if you run out of toilet paper, trump is sending every house this propaganda with your tax mone
##  [2] "an accessible overview of research in w/ @nazitalajevardi @kassrao @karamdana highlighting syst
##  [3] "i got to talk about my new @cup_polisci book on @thedamagereport with @johniadarola yesterday!
##  [4] "new paper, \"estimating candidate support in voting rights act cases: comparing iterative ei an
##  [5] "new work by @kassrao, @nazitalajevardi, &amp; @lorenc2 finds that mass movements that successfu
##  [6] "@realmabarreto @lorenc2 @skdreier24 for every click matt will donate $1 to the #clicks4kass fou
##  [7] "#cmps2020 conference @hlw_phd presents on how punitive interactions with criminal justice syst
##  [8] "great crowd here in westwood for the #irn v #esp game! we are cheering for iran #clicks4kass @
##  [9] "you jelly!! @realmabarreto #fomo #thefutureisfemale #womenalsoknowstuff #pocalsoknowstuff #cli
## [10] "#clicks4kass #msupriec @wearepriec https://t.co/cdnhnygh5r"
## [11] "@realmabarreto @karthickr @quicopedraza @wearepriec priec was my first conference as a grad st
## [12] "@karthickr @quicopedraza @wearepriec folks such as @kassrao #clicks4kass attended a priec whil
## [13] "@angelaxocampo @chinbo_chong looking forward to seeing your t-shirts at msu @wearepriec. #clic
## [14] "@angelaxocampo @chinbo_chong @umich @wearepriec if not, both of you have to order and wear #cl
## [15] "@chinbo_chong are you doubting #clicks4kass? some help here, @realmabarreto??"
## [16] "looking ahead to #mpsa18 this paper by alexandra mayorga looks interesting and builds on recen
## [17] "@realmabarreto @angelaxocampo @thewpsa @mpsanet #clicks4kass casino coming to atlantic city in
## [18] "@realmabarreto @angelaxocampo @thewpsa @mpsanet you got to learn from the best! he created a b
## [19] "@realmabarreto @angelaxocampo @thewpsa @mpsanet @realmabarreto i will sue you for copy rights
## [20] "@realmabarreto @hlw_phd @sergiogarciars @lseusablog i think i now need to run for office with
```

Step 3

Do you want tweets and hashtags that go back a long ways? Then you will want to use the twint python package. I have written an R function wrapper that works on my Mac OSx computer but has not been tested on Windows and other OSx versions.

The user will need to:

1. Download and install python3
2. Install the pip and twint python packages, ideally using pip installation
3. There is a possibility that python3 cannot be called from the R system() call due to path issues. I have attempted to solve this with the py_tweet() function though.

```
###################################################
# Running with Python to get full hashtag history #
###################################################

# User needs to:
# Install python3
# install package twint and pip
# Variation may exist for windows vs. mac

# Set Directory to where py_tweet.R is located #
# Note paths are called differently on Windows machines...

setwd("~/Dropbox/collingwood_research/posc_fall_20/POSC-207/lecture"); list.files()
```

```
## [1] "Clicks.csv"      "py_tweet.R"      "week1_tweet.pdf" "week1_tweet.rmd"
## [5] "week1_twitter.R" "week1.pdf"       "week1.rmd"
```

```
# Now source the function I wrote that creates a python file
source("py_tweet.R")

# Execute Function #
py_tweet(
    search = "'#Clicks4Kass'",
    until = "'2020-09-09'",
    since = "'2007-01-01'",
    limit = 10000000,
    output = "'Clicks.csv'",
    pfile = "twitter_hist.py",
    remove=TRUE
)
```

The code should print out all the tweets quickly to the screen, then store them along with metadata to a .csv file.

Now read that data in and do some quick checking.

```
# Read back in #
clicks_all <- read.csv("Clicks.csv", header=T)

# Check Dimensions #
dim(clicks_all)
```

```
## [1] 957  34
```

```
# Convert Text to Lower #
clicks_all$tweet <- tolower(clicks_all$tweet)

# Validate that the scraper mostly worked #
table(grepl("clicks4kass", clicks_all$tweet))
```

```
##
## FALSE  TRUE
##     9   948
```

```
# Look at the 'FALSE' tweets #
clicks_all[!grepl("clicks4kass", clicks_all$tweet), "tweet"]
```

```
## [1] "#unlocktheopen with @chase and make more of that energy felt at the @usopen even as you watch fi
## [2] "fall 2020 is full of uncertainty and stress. supporting your deaf students doesn't have to be. 
## [3] "lenovo legion x @playapex\nnow is your chance to rise above the rest. \nunmatched performance, 
## [4] "they were once the top nba players, so find out what happened to them after they retired."
## [5] "#unlocktheopen with @chase and make more of that energy felt at the @usopen even as you watch fi
## [6] "lenovo legion x @playapex\nnow is your chance to rise above the rest. \nunmatched performance, 
## [7] "fall 2020 is full of uncertainty and stress. supporting your deaf students doesn't have to be. 
## [8] "lenovo legion x @playapex\nnow is your chance to rise above the rest. \nunmatched performance, 
## [9] "lenovo legion x @playapex\nnow is your chance to rise above the rest. \nunmatched performance, 
```

```
# Subset out the 'FALSE' tweets #
clicks_all <- clicks_all[grepl("clicks4kass", clicks_all$tweet),]
dim(clicks_all)
```

```
## [1] 948  34
```

```
# Further Subset Columns

clicks_all <- dplyr::select(clicks_all, date, username, name,
                            place, tweet, photos, replies_count,
                            retweets_count, likes_count)

# Clean the Text #
clean_string <- function(string){

    # Lowercase
    temp <- tolower(string)
```

```r
    # Remove everything that is not a number or letter (may want to keep more
    # stuff in your actual analyses).
    temp <- stringr::str_replace_all(temp,"[^a-zA-Z\\s]", " ")

    # Shrink down to just one white space
    temp <- stringr::str_replace_all(temp,"[\\s]+", " ")

    # Clean front/back whitespace
    temp <- stringr::str_trim(temp)

    return(temp)
}

# Apply across the character tweet vector #
clicks_all$tweets_c <- sapply(clicks_all$tweet,
                              FUN = clean_string,
                              USE.NAMES = F)

head(clicks_all$tweets_c, 3)
```

```
## [1] "clicks kass all day baby don t miss this one lorenc hlw phd"
## [2] "clicks kass"
## [3] "and apologies to johnholbein for getting wrapped into this clicks kass debate over terrible ref
```

```r
# Some Basic Analysis #

#summary ( lm(likes_count ~ factor(username), data = clicks_all) )

#summary ( lm(retweets_count ~ factor(username), data = clicks_all) )
```