Twitter Data Mining using Tweepy and R with JSONLite

- Robert Zane Spalding

1. Setting up in Windows:

Items you will need:

1. **R** : Either RGui from https://cran.r-project.org/mirrors.html or RStudio, I used the RGUI as it is lighter
2. **Python** : I used version 3.5.1, changing the version may require changing the syntax of the script some.
3. **Tweepy** : A library to stream tweets. You can download a ZIP from https:github.com/tweepy/tweepy and install it by running the setup.py script in Python from the command prompt.
4. **JSONLite** : Can be retrieved using the RGui:
    a. Packages -> Install package(s) -> *select closest Mirror* -> JSONLite

2. Setting up in Ubuntu Linux:

1. **R** : Ubuntu: `sudo apt-get install r-base`
2. **Python** : Should already be included with Ubuntu
3. **Tweepy** : Can be installed using pip:
    a. `sudo apt-get install python-pip`
    b. `sudo pip install tweepy`
4. **JSONLite** : Can be installed using R in terminal from source file, the mirror is a bit buggy:
    a. `wget "https://cran.r-project.org/src/contrib/jsonlite_0.9.19.tar.gz"`
    b. `R`
    c. `install.packages("jsonlite", repos = NULL, type = "source")`

3. Instructions for getting Twitter Data:

Source for twitter_streaming script/setup: Adil Moujahid http:adilmoujahid.com/posts/2014/07/twitter-analytics/

In order to access Twitter Streaming API, we need to get 4 pieces of information from Twitter: API key, API secret, Access token and Access token secret. Follow the steps below to get all 4 elements:

- Create a twitter account if you do not already have one.
- Go to https:apps.twitter.com/ and log in with your twitter credentials.
- Click "Create New App"
- Fill out the form, agree to the terms, and click "Create your Twitter application"
- In the next page, click on "API keys" tab, and copy your "API key" and "API secret".

- Scroll down and click "Create my access token", and copy your "Access token" and "Access token secret".

Python Script to stream Tweet data containing desired key words into JSON format

- Usage:
  - Windows: install python -> open command prompt -> navigate to directory containing python script called twitter_streaming.py -> "py twitter_streaming.py > twitter_data.txt"
  - Linux: same thing, but use “python” not “py”
- I would recommend running the script and piping into several different files so that the operations have a more manageable time, I usually let them run for a couple of days before switching files

-- Begin Twitter_streaming.py --

```
#Import the necessary methods from tweepy library
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream

#Variables that contains the user credentials to access Twitter API
access_token = "ENTER YOUR'S"
access_token_secret = "ENTER YOUR'S"
consumer_key = "ENTER YOUR'S"
consumer_secret = "ENTER YOUR'S"


#This is a basic listener that just prints received tweets to stdout.
class StdOutListener(StreamListener):

    def on_data(self, data):
        print(data)
        return True

    def on_error(self, status):
        print(status)


if __name__ == '__main__':

    #This handles Twitter authetification and the connection to
Twitter Streaming API
    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth, l)

    #This line filter Twitter Streams to capture data by the keywords,
replace them with your own keywords
    stream.filter(track=['computer science', 'mechanical engineering',
'college major', 'political science', 'economics', 'accounting',
```

```
'civil engineering', 'criminal justice', 'nursing', 'business
administration', 'future major', 'major'])
```
-- End twitter_streaming.py --

4. Instructions for parsing Twitter Data from Tweepy:

Source: Robert Spalding

- Doing tweet parse operations in R
- Download R, recommend using R GUI environment as well on Windows
- Variable names and file names are hard-coded into the script, so if you change them you will have to change the script as well
- Blue annotates R Console Commands
- Green annotates saved Scripts

Import needed library, you have to download this as well which can be done through the R environment
```
library(jsonlite)
```

Reads in data from "twitter_data.json" which was renamed from "twitter_data.txt"
```
dat = readLines("twitter_data.json")
```

This script parses the data into the correct format to be entered into a R data frame
Write all scripts in seperate .r files

Write script
-- Begin write.r --
```
sink("outfile.json")
cat("[")
for (i in 1:(length(dat)/2))
{
      if(i%%2 == 0)
      {
            next
      }
      cat(dat[i])
      if(i == (length(dat)/2) || i == (length(dat)/2+1))
      {
            cat("]")
            next
      }
      cat(",")
}
sink()
```
-- End write script –

-- Run write.r script --
```
source('write.r')
```

-- Read parsed data in --

```
tweet_data = readLines("outfile.json")
```

-- Load parsed data into data frame using jsonlite library function --

```
mydf <- fromJSON(tweet_data)
```

- From here, you have a data frame which contains all the data you streamed
- In this document I'm only taking the 'text' field of the Tweets and operating on it, but you could do the same with any of the JSON fields you received
- If you wanted to take something else, such as the 'Place' of the Tweet, you would replace the '4' in mydf[i,4] below with the index of another column

Parse text to remove 'Text' field from the JSON objects

-- Being write_text.r --

```
sink("parsed_text.txt")
for (i in 1:(length(dat)/4-1))
{
      cat(mydf[i,4])
      cat("\n")
}
sink()
```

← Replace 4 with the index of the column you want to parse

-- end write_text.r --

-- Run write_text.r script --

```
source('write_text.r')
```

Do the above steps for all of the tweet_data files you created with twitter_stream.py

Combine all text files together to parse as one -- EmEditor program is best for this

```
library(jsonlite)
```

parsed_text_all.txt is the name of my combined parsed text files

-- Read in all tweets --

```
dat = readLines("parsed_text_all.txt")
```

Write tweets to a csv format (Excel) for easier reading

-- Begin write_csv.r --

```
sink("data_sheet.csv")
for (i in 1:(length(dat)))
{
      if (nchar(dat[i], type = "chars", allowNA = FALSE, keepNA = NA) >
0) {
            cat(dat[i])
      }
      else {
            next
      }
      if(i == floor(length(dat)))
```

```
	{
		break
	}
	cat(",\n")
}
sink()
```
-- End write_csv.r --
```
source("write_csv.r")
```

-- Read in cleaned up data in csv format --
```
dat = readLines("data_sheet.csv")
```

This script parses all the tweets and looks for the word "major", if "major" is not in the text it will not write it to the new file

-- Begin parse_grep.r --
```
sink("major.csv")
count = 0
for(i in 1:length(dat))
{
	if(grepl("major", dat[i], perl=TRUE)){
		count = count + 1
		cat(dat[i])
		cat("\n")
	}
}
sink()
cat(count)
```
-- End parse_grep.r --
```
source("parse_grep.r")
```

-- Read in new data --
```
major = readlines("major.csv")
```

You can also parse it further by using this script, in this case I first parsed by "major" and then by different majors.

If you want to look for more than one keyword in the tweet, you can add "`||grepl("psychology", major[i], perl=TRUE)`" to the if() statement to include as many keywords as you desire

Parse_major.r calls other scripts to parse individual majors in a folder off the root Directory called Major

-- Begin parse_major.r --
```
source('Major/parse_psychology.r')
source('Major/parse_teacher.r')
source('Major/parse_accounting.r')
source('Major/parse_biology.r')
source('Major/parse_business.r')
source('Major/parse_computer_science.r')
```

```
source('Major/parse_criminal_justice.r')
source('Major/parse_english.r')
source('Major/parse_history.r')
source('Major/parse_liberal_arts.r')
source('Major/parse_nursing.r')
```
-- End parse_major.csv --
```
source("parse_major.r")
```

-- Individual Major Scripts --

-- Begin parse_english.r --
```
sink("english_major.csv")
count = 0
for(i in 1:length(major))
{
      if(grepl("nglish Maj", major[i], perl=TRUE) || grepl("liter",
major[i], perl=TRUE) || grepl("nglish maj", major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("English: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_criminal_justice.r --
```
sink("criminal_justice_major.csv")
count = 0
for(i in 1:length(major))
{
      if(grepl("crim", major[i], perl=TRUE) || grepl("justice",
major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("Criminal Justice: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_computer_science.r --
```
sink("computer_science_major.csv")
count = 0
for(i in 1:length(major))
{
```

```
      if(grepl("omp sci", major[i], perl=TRUE) || grepl("CS ",
major[i], perl=TRUE) || grepl("omputer", major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("Computer Science: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_business.r --
```
sink("business_major.csv")
count = 0
for(i in 1:length(major))
{
      if(grepl("business", major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("Business: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_biology.r --
```
sink("biology_major.csv")
count = 0
for(i in 1:length(major))
{
      if(grepl("bio", major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("Biology: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_accounting.r --
```
sink("accounting_major.csv")
count = 0
for(i in 1:length(major))
```

```
{
      if(grepl("account", major[i], perl=TRUE) || grepl("finance",
major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("Accounting: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_teacher.r --
```
sink("teacher_major.csv")
count = 0
for(i in 1:length(major))
{
      if(grepl("teach", major[i], perl=TRUE) || grepl("educat",
major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("Teacher: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_psychology.r --
```
sink("psychology_major.csv")
count = 0
for(i in 1:length(major))
{
      if(grepl("psych", major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("Psychology: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_nursing.r --
```
sink("nursing_major.csv")
```

```
count = 0
for(i in 1:length(major))
{
      if(grepl("nurs", major[i], perl=TRUE) || grepl("R.N. ", major[i],
perl=TRUE) || grepl("RN ", major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("Nursing: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_liberal_arts.r --
```
sink("liberal_arts_major.csv")
count = 0
for(i in 1:length(major))
{
      if(grepl("liberal", major[i], perl=TRUE) || grepl("arts",
major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("Liberal Arts: ")
cat(count)
cat("\n")
```
-- End --

-- Begin parse_history.r --
```
sink("history_major.csv")
count = 0
for(i in 1:length(major))
{
      if(grepl("istory ", major[i], perl=TRUE)){
            count = count + 1
            cat(major[i])
            cat("\n")
      }
}
sink()
cat("History: ")
cat(count)
cat("\n")
```
-- End --