1.125 Arch & Engineering Software Systems
Term Project Assignment

Deliverable #2
Due Date:
Thursday, Nov 17th, 2016

Team:
Aswini Narayana Prasad and Shweta Jindal

Code Artifacts: Version2

A. Code for Twitter data Extraction & Establishing Training Data:

Below code is written in node.js. Here the update includes streaming of data as well as analyzing the sentiment using preliminary tools to generate the required test data. Currently we have extracted nearly 1200 tweets of which 70% will be used for testing. Also unlike last time, the required data extracting is programmed in node itself using additional NPM package – emotional.

Node.js

```
var Twit = require('twit');
var emotional = require("emotional");
//var arr=[];
var T = new Twit({

  consumer_key: 'CbcYhV2o2uzboJbzdEFdGnGW3',
  consumer_secret: 'PdNhNIK86m8dD8Q5mLPVPFDfodzXnj0yASsyg5oevmLL1j6dJu',
  access_token: '566483662-SnAgqKt0eslMceX2YVRLHdlzBLB7RLdjNBObxDxS',
  access_token_secret: '4tNBVYf1jCet9v6pKmDQmIczRGazPtf3HHrPD3ZQWA0LQ',

});
 var aa;
var fs = require('fs');
var util=require('util');
var logFile=fs.createWriteStream('log.json',{flags:'a'});
var logStdout=process.stdout;
```

```
var arr=[];
var i;
var r3=[];
var r4=[];
var jj=0;
var r2=[];
//T.get('search/tweets', { q: '#Audi since:1990-01-10',count:200},function(err, data, response) {
T.get('search/tweets', { q: '@apple since:2016-11-14 until:2016-11-15',count:100}, function(err,
data, response) {
var tweets=data.statuses, i=tweets.length;
for (j=0;j<i;j++)
{
arr[j]=tweets[j].text;

};

});



T.get('search/tweets', { q: '@apple since:2016-11-12 until:2016-11-13',count:100}, function(err,
data, response) {
var tweets=data.statuses, i=tweets.length;
for (j=0;j<i;j++)
{
arr[j+100]=tweets[j].text;

//console.log1(aa+","+tweets[j].text+";");
};

});

T.get('search/tweets', { q: '@apple since:2016-11-10 until:2016-11-11',count:100}, function(err,
data, response) {
var tweets=data.statuses, i=tweets.length;
for (j=0;j<i;j++)
{
arr[j+200]=tweets[j].text;

//console.log1(aa+","+tweets[j].text+";");
};
```

```
});

T.get('search/tweets', { q: '@apple since:2016-11-08 until:2016-11-09',count:100}, function(err,
data, response) {
var tweets=data.statuses, i=tweets.length;
for (j=0;j<i;j++)
{
arr[j+300]=tweets[j].text;
//console.log1(aa+","+tweets[j].text+";");
};
for (j=0;j<400;j++)
{
aa=j+1;
console.log1(aa+"       "+arr[j]+"\n");
emo(arr[j]);
}
});



 console.log1=function(){

        logFile.write(util.format.apply(null, arguments));
 //     logStdout.write(util.format.apply(null, arguments)+"    ");
 }

function emo(sen){
        emotional.load(function () {
                ab=emotional.get(sen).polarity ;
console.log1(ab+"\n");
console.log(ab);
});

}
```

OUTPUT

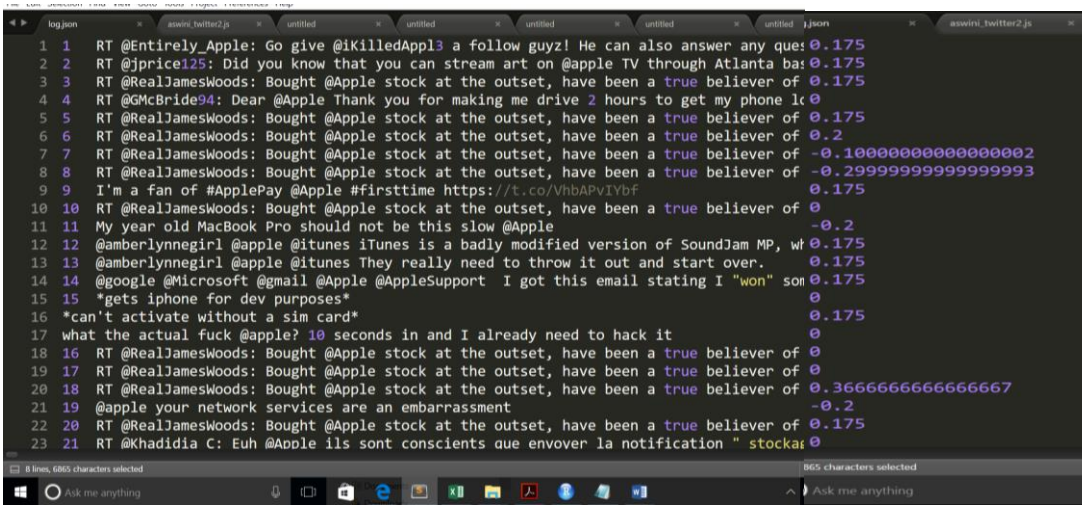

Fig 1: Node output of Tweets extracted & emotional scores allocated using NPM – tweet & emotional

B. Code for Reading & Cleaning the Twitter data in R

Updates include

- 5 steps data cleaning,
  - **A-** Lower case conversion
  - **B-** Removing punctuations
  - **C-** removing Stopwords (Pronouns, articles, etc.)
  - **D-** Stemming (root word extraction)
  - **E-** TM mapping (Mapping into a document matrix)

```
# R code to clean up the tweets

# for sentiment analysis for a company such as Apple

# read the .csv file into a dataframe

tweets = read.csv("tweets.csv", stringAsFactor = FALSE)
```

```
# display the structure of the dataframe tweets

str(tweets)
summary(tweets)

#filter the negative tweets with an averge score < -1

tweets$Negative = as.factor(tweets$Avg < = -1)

#diplay the table

table(tweets$Negative)

#install tm,SnowballC packages

install.packages("tm")
install.packages("SnowballC")

#load package

load(tm)
load(SnowballC)

#create corpus

corpus = Corpus(VectorSource(tweets$Tweet))

#use tm_map function is tm library to convert all the tweets to lowercase - STEP1 in cleaning
corpus = tm_map(corpus, tolower)

#use this to display the first tweet to verify if it is coverted to lowercase
corpus[[1]]

#list the stopwords usually in english

stopwords("english") [1:10]

#this command is removing words - apple, stopwords such as i, your, me, my, myself etc. if you
want some special words removed - add along with apple below - STEP2 in cleaning

corpus = tm_map(corpus, removeWords, c("apple",stopwords("english")))
```

#use this to display the first tweet to verify if stopwords are removed
corpus[[1]]

#this command performs stemming function as in - argued, arguing, argue - all is convereted to argue

corpus = tm_map(corpus, stemDocument)

#use this to display the first tweet to verify if stemming worked
corpus[[1]]

#data is clean to some extend

Output of R file

> tweets = read.csv("tweets.csv")

> str(tweets)
'data.frame': 1181 obs. of 2 variables:
$ Tweet: Factor w/ 1133 levels ":-) \"Turns out that 'c' in Apple's iPhone 5c doesn't stand for 'cheaper\" @apple #iphone http://lnkd.in/bMWsyRR",..: 687 768 829 987 20 123 890 1000 455 801 ...
$ Avg : num 2 2 1.8 1.8 1.8 1.8 1.8 1.6 1.6 1.6 ...

> tweets = read.csv("tweets.csv", stringsAsFactor = FALSE)
> str(tweets)
'data.frame': 1181 obs. of 2 variables:
$ Tweet: chr "I have to say, Apple has by far the best customer care service I have ever received! @Apple @AppStore" "iOS 7 is so fricking smooth & beautiful!! #ThanxApple @Apple" "LOVE U @APPLE" "Thank you @apple, loving my new iPhone 5S!!!!! #apple #iphone5S pic.twitter.com/XmHJCU4pcb" ...
$ Avg : num 2 2 1.8 1.8 1.8 1.8 1.8 1.6 1.6 1.6 ...
> tweets$Negative = as.factor(tweets$Avg <= -1)
> table(tweets$Negative)

FALSE TRUE
999 182
> install.packages("tm")
> library(tm)

Loading required package: NLP
> install.packages("SnowballC")
> library(SnowballC)
> corpus = Corpus(VectorSource(tweets$Tweet))
> corpus[[1]]
“I have to say, Apple has by far the Best Customer care service I have ever received! @apple @appstore"

> corpus = tm_map(corpus, tolower)
> corpus[[1]]
[1] "i have to say, apple has by far the best customer care service i have ever received! @apple @appstore"
> summary(tweets)
```
   Tweet              Avg           Negative
 Length:1181        Min.   :-2.0000   FALSE:999
 Class :character   1st Qu.:-0.6000   TRUE :182
 Mode  :character   Median : 0.0000
                    Mean   :-0.1931
                    3rd Qu.: 0.2000
                    Max.   : 2.0000
```

> stopwords("english") [1:10]
[1] "i" "me" "my" "myself" "we" "our" "ours" "ourselves" "you" [10] "your"
> corpus[[1]]
[1] "i have to say, apple has by far the best customer care service i have ever received! @apple @appstore"
> corpus = tm_map(corpus, removeWords, c("apple",stopwords("english")))
> corpus[[2]]
[1] "ios 7 fricking smooth & beautiful!! #thanxapple @appl"
> corpus = tm_map(corpus, stemDocument)
> corpus[[1]]
[1] " say, far best custom care service ever receiv! @ @appstor"

C. Code for Machine Learning – Training the model (Work in progress- not completely functional codes)

#Next step - to work with the cleaned data to generate classification MODELs. Work in Progress

```
# inspect the frequencies
findFreqTerms(frequencies lowFreq=20)

#create a sparse matrix
sparse = removeSparseTerms(frequencies, 0.995)

#install the related modeling packages
install.packages("rpart")
install.packages("rpart.plot")

#load the libraries
library(rpart)
library(rpart.plot)
```