Twitter Sentiment Analysis with R

——————————————————————-

Goal: With the 2020 Election approaching, this project aims to perform sentiment analysis to have an overview of people's attitude toward the election. The lexical analysis that we utilize can be used to predict the sentiment of tweets and subsequently express the opinion graphically through data visualization.

Flow of Data Analysis:

1. Twitter Authentication
2.
    - Create Twitter application
    - Used twitterR to provide an interface to the Twitter web API
    - Create twitter authenticated credential object, It is done using consumer key and consumer secret, access token, access secret.

3. Text Extraction from Twitter
    - Select tweets that we want to research by using the keyword "2020 election."

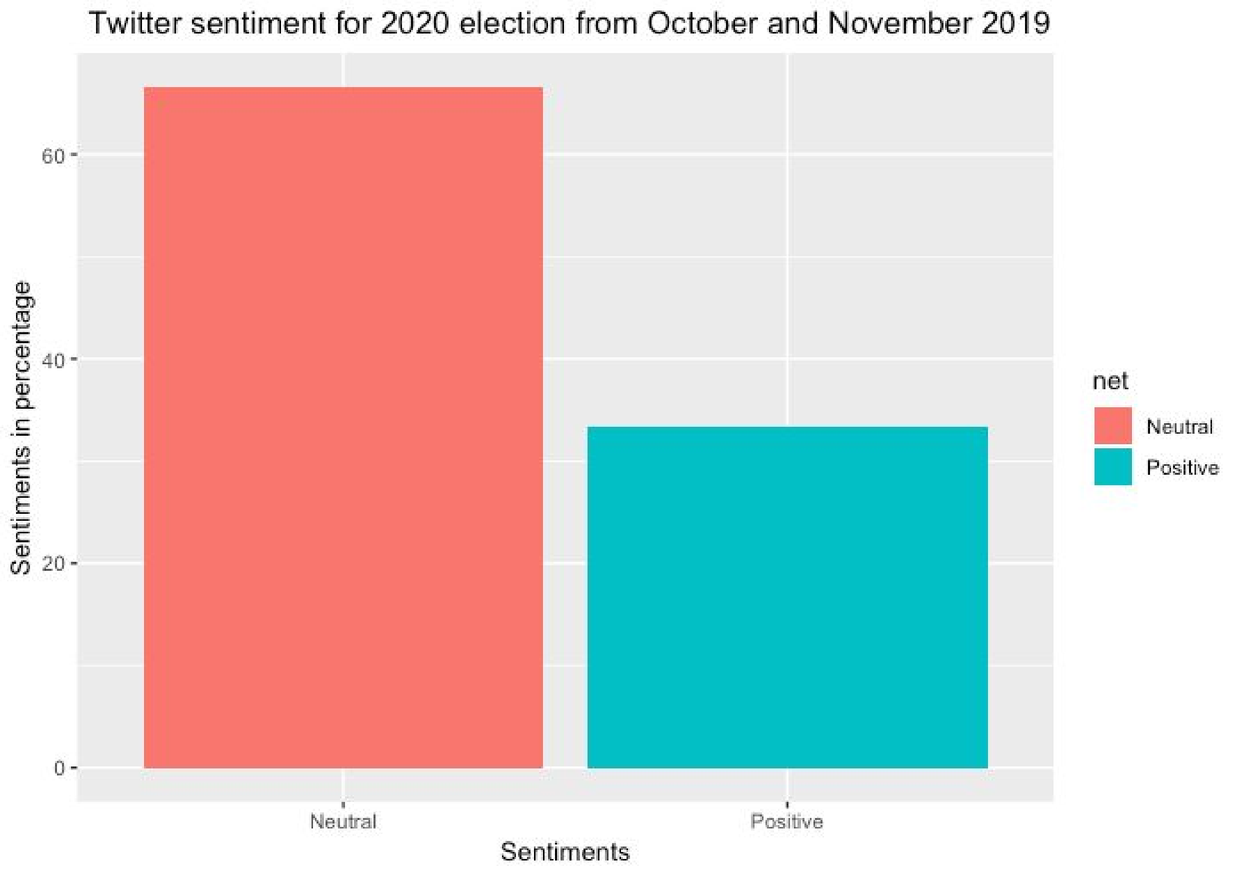
4. Data Cleaning
    - Remove emotions, URLs, punctuations, stop words and extra white spaces
    - Convert tweets to lower case

5. Database Loading
    - Load the database containing positive and negative words into R.This is used for Lexical Analysis, where the words in the tweets are compared with the words in the database and the sentiment is predicted.
    - Add extra words into the database based on the context we are studying.

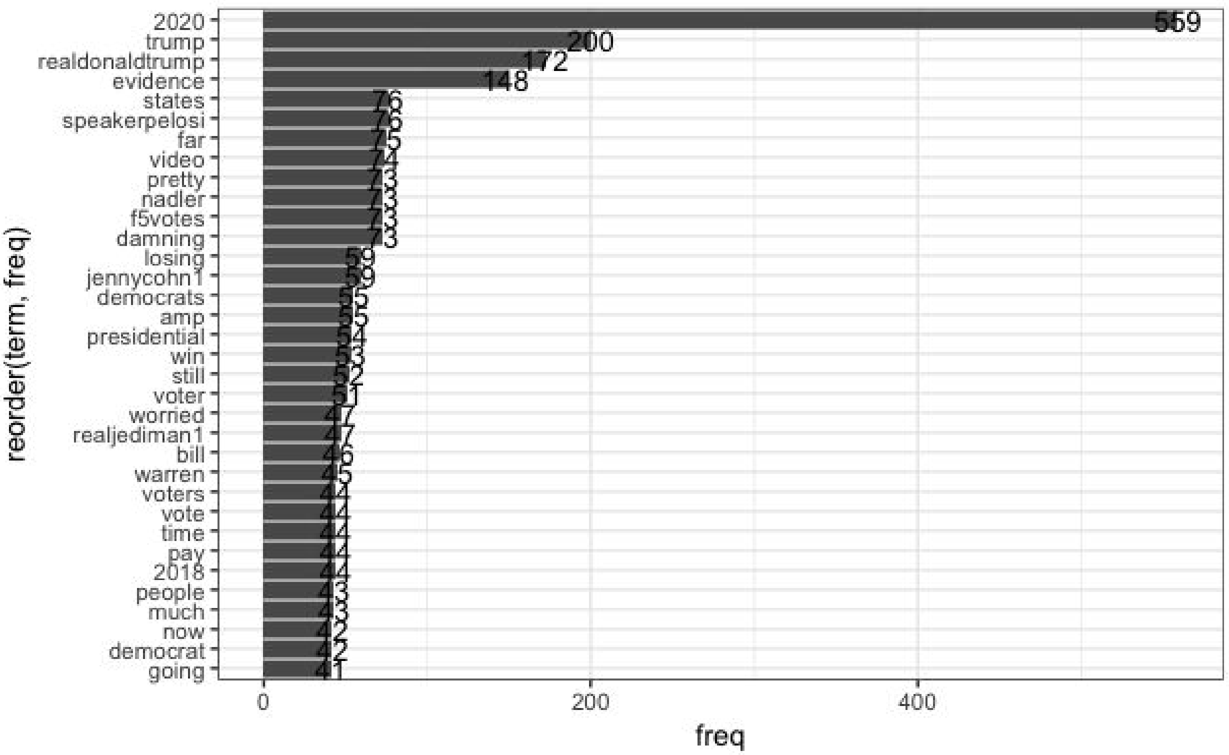
6. Lexical Analysis
    - By comparing uni-grams to the pre-loaded word database, tweets are assigned sentiment scores - positive, negative or neutral and overall score is calculated.
    - Calculate sentiment scores
    - Display the proportion of various sentiments (positive, negative and neutral)

7. Data Visualization

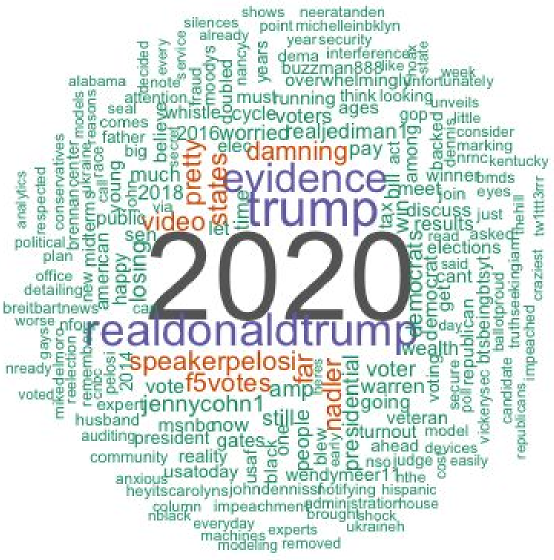
Visualization Excerpts



Observation: It’s shown in the graph that negative sentiment is missing. Meanwhile, Neutral: Positive = 2: 1. We imply that most people on Twitter do not hold negative attitude toward 2020 election.



Observation: This graph shows what people tweets most on the topic of 2020 election. Besides "2020", the top three most frequently words appear in tweets are "trump", "realdonaldtrump" and "evidence." We also observe that a number of names/users such as Jenny Cohn, speakerpelosi, f5vites and RealJediman. To have further understanding, it's necessary to research on these people.



Observation: A wordcloud allows a quick visualization of most frequent words and in our case to confirm that the frequent words of Twitter user are tweeting. Apart from the most frequent words we mention above, we can see people are concerned about tax, democracy, security and wealth. Some keywords for the identity or race are also worthy of attention, such as veteran, black and hispanic.

Reference:

https://github.com/vedantnarayan/Twitter-Sentiment-Analysis-Using-R/blob/master/twitter_Auth.R

https://www.quora.com/How-can-I-read-Twitter-data-directly-in-R

https://github.com/Twitter-Sentiment-Analysis/R

Appendix:

~~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Twitter Authentication
```{r}
library(twitteR)
library(ROAuth)
consumerKey = "5pdz5NUS6JC6mbQROBeof12bD"
consumerSecret = "ZuiaHYqQmwKjRFfjqNo3IPnrvQTBZDAPhkW4CJrOFtC6atLDQU"
accessToken = "181626111-0P6CUE2h4hgdGUIH9pgdAUDvJ14ubtOZ83wPouYA"
accessSecret = "OnZ1VlUqhpuCy6UYg2ZEaiSs14sD2xJfA7Av6w5rbLZ7X"

##download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem") #downloads
the certificate

setup_twitter_oauth(consumerKey, consumerSecret, accessToken, accessSecret)

cred <- OAuthFactory$new(consumerKey=consumerKey,
          consumerSecret=consumerSecret,
          requestURL='https://api.twitter.com/oauth/request_token',
          accessURL='https://api.twitter.com/oauth/access_token',
          authURL='https://api.twitter.com/oauth/authorize')


```

Select tweets to research
```{r}
library(tm)
tweets <- searchTwitter('2020 election',n =1000,since = "2019-10-01",until = "2019-11-10",lang
= "en",resultType = "recent")
data<-do.call("rbind",lapply(tweets,as.data.frame))
df_tweets<-data$text
df_Corpus<-Corpus(VectorSource(df_tweets))
```

Tweets Cleaning
```{r}
#Remove emotions
emotions <- content_transformer(function(x) iconv(x, "latin1", "ASCII", sub=""))
tweets_clean <- tm_map(df_Corpus,emotions)
~~~~

~~~
##Remove Urls
URL <- content_transformer(function(x) gsub("https?:\\/\\/(.*?|\\/)(?=\\s|$)\\s?", "", x,
perl=T))
tweets_clean <- tm_map(tweets_clean,URL)

## Remove Punctuations
punctuations <-  content_transformer(function(x) gsub("[[:punct:][:blank:]]+", " ",
tweets_clean))
tweets_clean <- tm_map(tweets_clean,punctuations)

## Convert tweets to lower case
tweets_clean <- tm_map(tweets_clean,content_transformer(tolower))

## Remove the stopwords of English
tweets_clean <- tm_map(tweets_clean,removeWords,stopwords("English"))

## Remove any space
tweets_clean <- tm_map(tweets_clean,stripWhitespace)

## Further Cleaning
tweets_clean <-
tm_map(tweets_clean,removeWords,c("https","election","will","says","is","are"))
```
~~~

Sentiment Analysis

~~~
```{r}
#Convert the content of corpus into a dataframe
df_cleanedTweets <-  data.frame(text=get("content", tweets_clean),
   stringsAsFactors=F)

list_cleanedTweets <- as.list(df_cleanedTweets$text)


library(stringr)
#Eliminate extra spaces
list_cleanedTweets <- lapply(list_cleanedTweets,function(x) gsub(pattern = "\\s+","
",str_trim(x)))

#Splite sentence into words and make it a list of characters
list_cleanedTweets <- lapply(list_cleanedTweets,function(x) strsplit(x,split = " "))
unlist_CleanedTweets <- sapply(list_cleanedTweets,unlist)

#Name change
voteTweets <- unlist_CleanedTweets

#Load word database
~~~

~~~
positive <- scan("~/desktop/positive-words.txt",what = "character",comment.char = ";")
negative <-scan("~/desktop/negative-words.txt",what = "character",comment.char = ";")
negative = c(negative, 'wtf', 'behind','feels', 'ugly', 'back','worse' , 'shitty', 'bad',
'no','freaking','sucks','horrible','die')


#Calculate the positive score and negative scores
p_scores <- lapply(voteTweets,function(x){sum(!is.na(match(x,positive)))})
n_scores <- lapply(voteTweets,function(x){sum(!is.na(match(x,negative)))})

#Calculate the net sentiment score
net_scores <-
lapply(voteTweets,function(x){sum(!is.na(match(x,positive)))-sum(!is.na(match(x,negative)))})

#Unlist the socres and return vectors of integers
pos = unlist(p_scores)
neg = unlist(n_scores)
net = unlist(net_scores)

#Classfied all tweets as positive, negative or neutral
net[net>0]="Positive"
net[net<0]="Negative"
net = ifelse(net=="0","Neutral",net)

#convert net into a factor variable
net = as.factor(net)

#Display the percentage of each sentiment
prop.table(table(net))
```
~~~

Data Visualization

~~~
```{r}
library(ggplot2)

plot <- ggplot(data.frame(net),aes(x=net)) + geom_bar(aes(fill=
net,y=((..count..)/sum(..count..))*100))

plot <- plot+ labs(title =" Twitter sentiment for 2020 election from October and November
2019", x = "Sentiments", y = "Sentiments in percentage") + theme(plot.title =
element_text(hjust = 0.5))

red.bold.italic.text <- element_text(face = "bold.italic", color = "blue")

plot + theme(title = red.bold.italic.text, axis.title = red.bold.italic.text)
~~~

~~~
```
```{r}
tdm_tweet<-TermDocumentMatrix(tweets_clean)
TDM1<-as.matrix(tdm_tweet)
freq.terms <- findFreqTerms(tdm_tweet, lowfreq = 50)
term.freq <- rowSums(TDM1)
term.freq <- subset(term.freq, term.freq > 40)
df <- data.frame(term = names(term.freq), freq= term.freq)
ggplot(df, aes(reorder(term, freq),freq)) + theme_bw() + geom_bar(stat = "identity")  +
coord_flip() +labs(list(title="Term Frequency Chart", x="Terms", y="Term Counts")) +
geom_text(aes(label=freq))
```


```{r}
library(wordcloud)
library(RColorBrewer)

tdm_tweet<-TermDocumentMatrix(tweets_clean)
TDM1<-as.matrix(tdm_tweet)
v = sort(rowSums(TDM1), decreasing = TRUE)
summary(v)

wordcloud(tweets_clean, scale=c(5,0.5), max.words=200, random.order=FALSE, rot.per=0.35,
use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
~~~