Problem 3

Extracting twitter feeds through API and performing Text analysis on feeds

Vinod

July 8, 2018

Goal:

- To extract the social media feeds of customers
- To find insight on the topic/news currently trending with the customers with respect to Business growth

Framework:

1. Decide on the list of social media sites, say Twitter and Facebook, from which to gather information
2. Extract the public feeds, posts and comments of user ids for the list of search terms. Since I already have customer details, I can use the same for extracting feeds. Feeds can be extracted through R or Python using respective API.
3. Cleanse the data employing appropriate text mining methods/packages
4. Apply 'Latent Dirichlet Allocation' algorithm for topic modelling to find out underlying latent theme or factor for the most tweeted or commented words and to group them logically
5. Infer from the topic what interests the user or topic most talked about.

Technical Solution:

I would like to illustrate this by connecting to twitter account and extracting 'public' feeds related to 'business' hashtag and analyzing the feeds to find the most frequent tweeted words

I am using below packages in Python and R.

Python: tweepy; **R:** tm and wordcloud

This approach consists of 2 steps

a. Extract twitter feeds using Python
b. Process the feeds to identify most frequent words using R.

a. Extracting Twitter feeds using Python

I am using ‘**tweepy**’ package for connecting to twitter through API

```
## Extracting data from twitter
import tweepy
```

I need consumer key, consumer secret key, Access token and Access token secret key for accessing to twitter.

I generated this information from my twitter developer account. I am hiding credentials since it belongs to my personal twitter account.

```
## Define the API credentials in order to connect to twitter and extract twee
ts
API_Key = 'xxxxx'
API_Secret = 'xxxx'
Access_Token = 'xxxx'
Access_Token_Secret = 'xxxx'
## Call authentication handler to verify the API credentials
access = tweepy.OAuthHandler(API_Key,API_Secret)
## set access token for API requests
access.set_access_token(Access_Token,Access_Token_Secret)
## call instance of API class to access various functions
tweet_API = tweepy.API(access)
```

I created only two keywords(hashtags) based on the problem. There can be numerous

```
## create a list of keywords to search for the specific terms
search_terms = ['#business','growth']
```

There is provision to search for specific user public tweets as well. Below function can be used for that. Since I don’t have specific customer name/id, I am searching for tweets in entire twitter and not based on user id

```
## extract public tweets from specific user timeline

# user = tweet_API.user_timeline(user_id = '@abc')
```

Time to extract the tweets from twitter.

I specified to do search operation with selected keywords in tweets written since 1st June 2018 and extract full tweet message if tweet length >300. I don’t want to consider very short tweets, so I specified tweet length to be greater than 300. Also I extract tweets that are written only in English.

```
a=[]
for record in tweepy.Cursor(tweet_API.search,
                            q=search_terms,
                            tweet_mode='extended',
                            lang='en',
                            since='2018-06-01').items(limit=1000):
    if (len(record.full_text)>=300):
        a.append(record.full_text)
```

I am loading the extracted tweets to text file to be processed further in R

```
print('Number of Tweets extracted = ',len(a))

## Number of Tweets extracted =  44

tweets = open('C:/Users/Universe/Desktop/tweets.txt','w')
for element in a:
    tweets.write('%s\n'% element.encode('utf8'))
```

b. Extracting the tweets in R to process further to know the frequent words customers are talking related to 'Business' and 'Growth' hastags.

It involves following steps,

1. creating a corpus of tweets

2. Data cleansing - remove punctuations, stopwords, uder defined stopwords, white spaces and numbers and to convert all characters to lower case

3. Stemming - to extract the root word. Eg) work is the root word of working, worked, works

4. create a term document matrix which is matrix of frequency of each word in each doucument

5. calculate the frequency of each word in the entire tweets 6. plot the words to understand most talked about topic

```
tweets = read.csv('C:/Users/Universe/Desktop//tweets.txt',header=FALSE,col.na
mes='Tweets',sep='\n')
head(tweets)

##
Tweets
## 1        b'Relearning the #Secrets of #Economics &gt;&gt; https://t.co/eP2
P0AlxaR\\n\\n#planning #Startups #success #growth #business #SmallBusiness #E
ntrepreneurship #entrepreneur #WomenEmpowerment #womenentrepreneurs #selfempl
```

```
oyed #inspiration #Leadership #Coaching #wealth #DigitalMarketing https://t.c
o/5ItahBn8Xt'
## 2        b'Join an amazing FB group for lash artists &amp; any1 becoming q
ualified LASH LOVE! https://t.co/WacDsTJNme\\n#lash #lashextensions #facebook
#group #business #tips #growth #marketing #workout #workfromhome  #eyelashext
enions #eyelash #eyebrows #lashesonfleek #lashlove #lashsupplies https://t.co
/JBJlpFgcSx'
## 3  b'A great startup takes time. So, be #patient &gt;&gt; https://t.co/eP2
P0AlxaR\\n\\n#planning #Startups #success #growth #business #SmallBusiness #E
ntrepreneurship #entrepreneur #WomenEmpowerment #womenentrepreneurs #selfempl
oyed #inspiration #Leadership #Coaching #wealth #DigitalMarketing https://t.c
o/x2Af0SaNlV'
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(tm)
```

```
## Loading required package: NLP
```

```
file<-'C:/Users/Universe/Desktop/tweets.txt'
text = file(file,open="r")
text.decomposition =readLines(text)
tweet_corpus <- Corpus(VectorSource(text.decomposition))
tweet_corpus
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 42
```

```
# Data Clearning
# stopwords("english") # to know stopwords in english

tweet_corpus <-tm_map(tweet_corpus,tolower)
```

```
## Warning in tm_map.SimpleCorpus(tweet_corpus, tolower): transformation drop
s
## documents
```

```
tweet_corpus <-tm_map(tweet_corpus,removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(tweet_corpus, removeNumbers): transformatio
n
## drops documents
```

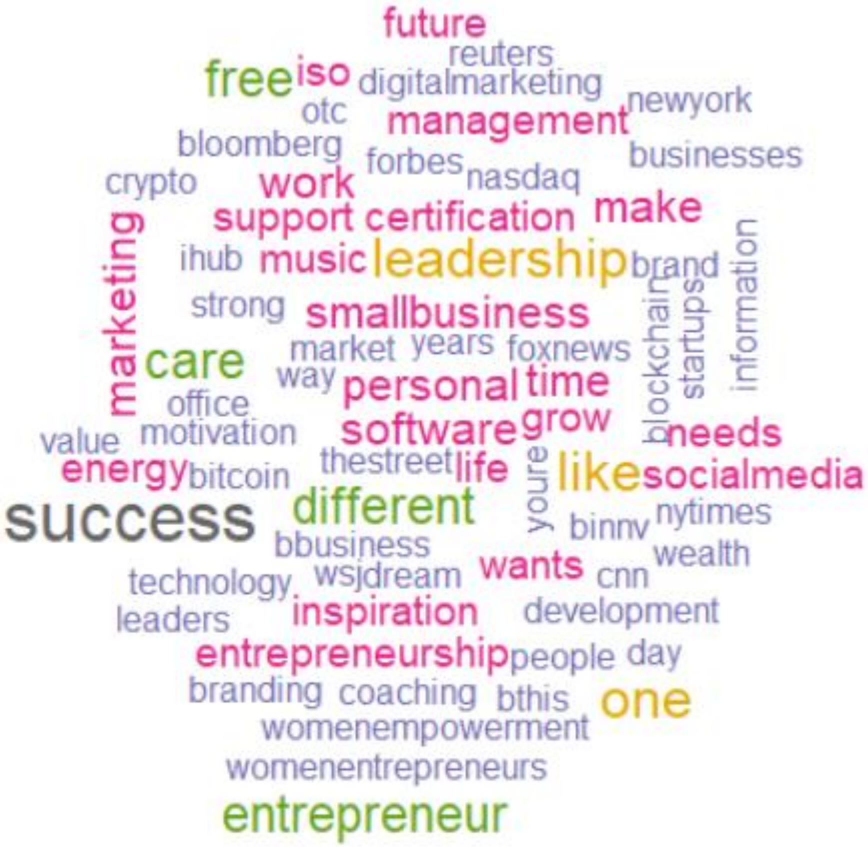
```
tweet_corpus <-tm_map(tweet_corpus,removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(tweet_corpus, removePunctuation):
## transformation drops documents
```

```
# to remove additional irrelevant words from tweets
selfstopwords <- c("b'",'business','growth','the','now','amp','get','want','c
an','bwe','gtgt','find')
tweet_corpus <- tm_map(tweet_corpus,removeWords, c(stopwords("english"),selfs
topwords))

## Warning in tm_map.SimpleCorpus(tweet_corpus, removeWords,
## c(stopwords("english"), : transformation drops documents

tweet_corpus <-tm_map(tweet_corpus,stripWhitespace)

## Warning in tm_map.SimpleCorpus(tweet_corpus, stripWhitespace):
## transformation drops documents

tweet_corpus <-tm_map(tweet_corpus,stemDocument)

## Warning in tm_map.SimpleCorpus(tweet_corpus, stemDocument): transformation
## drops documents

tweet_corpus_tdm <-TermDocumentMatrix(tweet_corpus)

word_freq <-rowSums(as.matrix(tweet_corpus_tdm))
word_freq <-sort(word_freq,decreasing=TRUE)
```

Let’s Visualize the top words peolpe are taking with respect to ‘business’ and ‘growth’ hashtags

```
wordcloud(names(word_freq)[1:200],word_freq[1:200],scale = c(2, 0.6),colors=b
rewer.pal(8,"Dark2"))
```



In the above word cloud, size of the words is directly proportional to frequency of those in the tweets. Customer are more talking about entrepreneurship especially about women entrepreneurs, leadership and success. Also, we can see mentions of business news channels like wsj. Nytimes, cnn, foxnews.

This exploratory analysis can be further taken forward using text mining techniques like **Latent Dirichlet Allocation** to further find out latent factors behind the tweets or posts.