Analysis of Brexit Using Twitter Data

Thejas Raju

Sumanth Sajjan

Importing required packages

In [1]:

```
import pandas as pd
import numpy as np
import csv
from textblob import TextBlob

#For creating Twitter API
import tweepy

# For plotting and visualization:
from IPython.display import display
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

Establishing connection to Twitter API to mine data

In [2]:

```
consumer_key = ""
consumer_key_secret = ""
access_token = ""
access_token_secret = ""

auth = tweepy.OAuthHandler(consumer_key,consumer_key_secret)
auth.set_access_token(access_token,access_token_secret)
api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)
```

Mining tweets with search term, language and number of tweets specified and repeating the process to download enough data for analysis

In [3]:

```
tweets=[];
for i in range(0,20):
    tweets = tweets + api.search("Brexit", "en", count=100)

csvFile = open('tweets.csv','w',newline='',encoding='utf-8')

#Use csv writer
csvWriter = csv.writer(csvFile)
csvWriter.writerow(['Tweet','Length','Date','Source','Likes','Retweet_Count'])
for tweet in tweets:
    csvWriter.writerow([tweet.text.encode('utf-8'), len(tweet.text), tweet.created_at,
 \
                        tweet.source, tweet.favorite_count, tweet.retweet_count])

csvFile.close()
```

Read the CSV dump

In [16]:

```
tweets_data = pd.read_csv('tweets.csv')
tweets_data.head()
```

Out[16]:

	Tweet	Length	Date	Source	Likes	Retweet_Count
0	b'RT @AmyMek: Free Speech Is Dead in Britain!\...	140	2019-03-20 09:11:56	Twitter Web Client	0	4705
1	b'RT @VBOFEB: What are the challenges for the ...	144	2019-03-20 09:11:56	Twitter for iPhone	0	1
2	b'RT @RCorbettMEP: #Brexit is no longer the wi...	88	2019-03-20 09:11:56	Twitter for iPhone	0	301
3	b'@MadameGPWales following the lines of brexit...	52	2019-03-20 09:11:56	Twitter Web Client	0	0
4	b'RT @RCorbettMEP: #Brexit is no longer the wi...	88	2019-03-20 09:11:55	Twitter for Android	0	301

Data Preprocessing for Sentiment Analysis

In [17]:

```
import re, unicodedata
from nltk import word_tokenize
from nltk.stem import WordNetLemmatizer


def remove_non_ascii(words):
    """Remove non-ASCII characters from list of tokenized words"""
    new_words = []
    for word in words:
        new_word = unicodedata.normalize('NFKD', word).encode('ascii', 'ignore').decode
('utf-8', 'ignore')
        new_words.append(new_word)
    return new_words

def lowercase(words):
    """Convert all characters to lowercase from list of tokenized words"""
    new_words = []
    for word in words:
        new_word = word.lower()
        new_words.append(new_word)
    return new_words

def remove_punctuation(words):
    """Remove punctuation from list of tokenized words"""
    new_words = []
    for word in words:
        new_word = re.sub(r'[^\w\s]', '', word)
        if new_word != '':
            new_words.append(new_word)
    return new_words

def normalize(words):
    words = remove_non_ascii(words)
    words = lowercase(words)
    words = remove_punctuation(words)
    return words

def lemmatize(words):
    """Lemmatize verbs in list of tokenized words"""
    lemmatizer = WordNetLemmatizer()
    lemmas = []
    for word in words:
        lemma = lemmatizer.lemmatize(word)
        lemmas.append(lemma)
    return lemmas

# Stopwords are not removed as it results in removal of most of the words in tweets and
it does not affect the tweet's sentiment
```

In [18]:

```
# Cleaning tweet for sentiment analysis
tweets_data['Cleaned Tweet']= [lemmatize(normalize(word_tokenize(tweet))) for tweet in
tweets_data['Tweet']]
tweets_data['Sentiment']= [TextBlob(str(tweet)).sentiment[0] for tweet in list(tweets_d
ata['Cleaned Tweet'])]
```

In [19]:

```
# Lebaling the sentiment as Positive,Negetive or Neutral based on the value returned by
TextBlob
i=0;
tweets_data['Sentiment']='';
for tweet in tweets_data['Cleaned Tweet']:
    j = TextBlob(str(tweet)).sentiment[0]
    if(j > 0):
        tweets_data['Sentiment'][i]='Positive'
        i=i+1
    elif(j < 0):
        tweets_data['Sentiment'][i]='Negative'
        i=i+1
    else:
        tweets_data['Sentiment'][i]='Neutral'
        i=i+1
```

In [20]:

```
# Data ready for analysis
tweets_data.head()
```

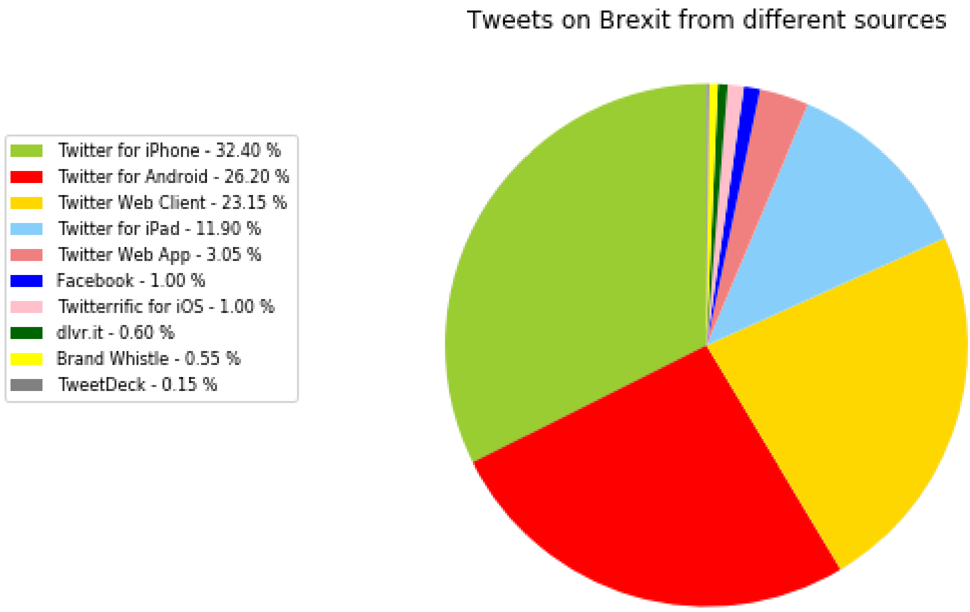
Out[20]:

	Tweet	Length	Date	Source	Likes	Retweet_Count	Cleaned T
0	b'RT @AmyMek: Free Speech Is Dead in Britain!\...	140	2019-03-20 09:11:56	Twitter Web Client	0	4705	[brt, amymek free, speech, dead, in, brit.
1	b'RT @VBOFEB: What are the challenges for the ...	144	2019-03-20 09:11:56	Twitter for iPhone	0	1	[brt, vbofeb, what, are, the challenge, fo
2	b'RT @RCorbettMEP: #Brexit is no longer the wi...	88	2019-03-20 09:11:56	Twitter for iPhone	0	301	[brt, rcorbettr brexit, is, no, longer, the...
3	b'@MadameGPWales following the lines of brexit...	52	2019-03-20 09:11:56	Twitter Web Client	0	0	[b, madamegpw following, the line, of, b...
4	b'RT @RCorbettMEP: #Brexit is no longer the wi...	88	2019-03-20 09:11:55	Twitter for Android	0	301	[brt, rcorbettr brexit, is, no, longer, the...

Plotting and Analysis

In [21]:

```
# Analysis on source of tweets
Source = tweets_data.Source.value_counts(normalize=True)

# Ploting
fig, ax = plt.subplots()
ax.axis('equal')

colors = ['yellowgreen','red','gold','lightskyblue','lightcoral','blue','pink', 'darkgr
een','yellow','grey']
percent = 100*Source

patches, texts = plt.pie(Source, colors=colors, startangle=90, radius=1.5)
labels = ['{0} - {1:1.2f} %'.format(i,j) for i,j in zip(Source.index, percent)]

plt.legend(patches, labels, bbox_to_anchor=(-0.1, 1.), fontsize=8)

plt.title('Tweets on Brexit from different sources', y=1.2)
plt.show()
```

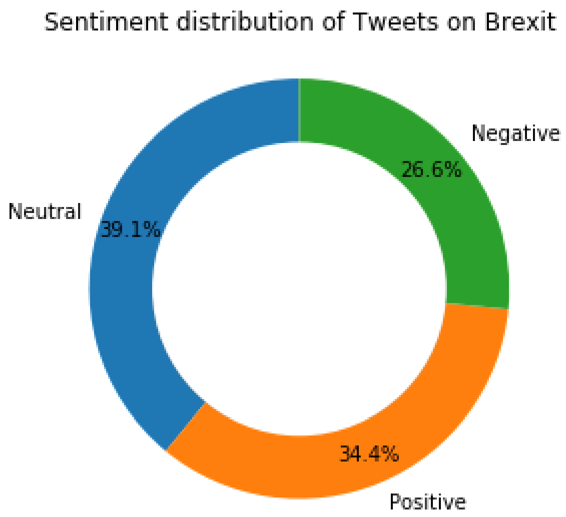


Analysis on source of tweets

The above pie chart shows that Mobile phones are the source for more than 70 percent of the tweets on brexit and around 25 percent is from twitter web. Rest of the sources combined accounts to less than 20 percent of tweets

In [22]:

```
# Analysis on sentiment distribution of tweets
Sentiment = tweets_data.Sentiment.value_counts(normalize=True)

# Ploting
fig, ax = plt.subplots()
ax.axis('equal')

plt.pie(Sentiment,labels=Sentiment.index, autopct='%1.1f%%', startangle=90, pctdistance
=0.85)

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.title('Sentiment distribution of Tweets on Brexit', y=1)
plt.tight_layout()
plt.show()
```

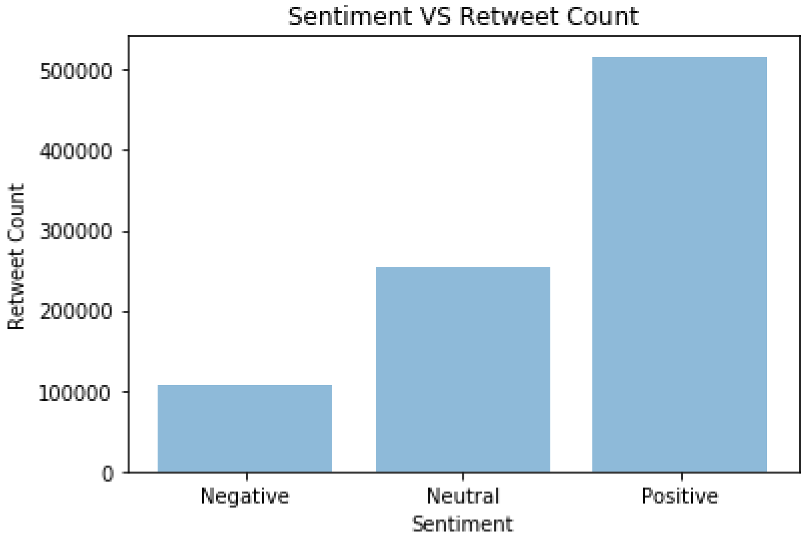


Analysis on sentiment distribution of tweets

Nearly 75 percent of the tweets are either Positive or Neutral tweets, where as Negative tweets accounts to nearly 26 percent

In [23]:

```
# Analysis on sentiment V/S retweet count
retweet = tweets_data.groupby('Sentiment').sum().Retweet_Count

# Plotting
y_pos = np.arange(len(retweet))

plt.bar(y_pos, retweet, align='center', alpha=0.5)
plt.xticks(y_pos, retweet.index)
plt.xlabel('Sentiment')
plt.ylabel('Retweet Count')

plt.title('Sentiment VS Retweet Count')

plt.show()
```

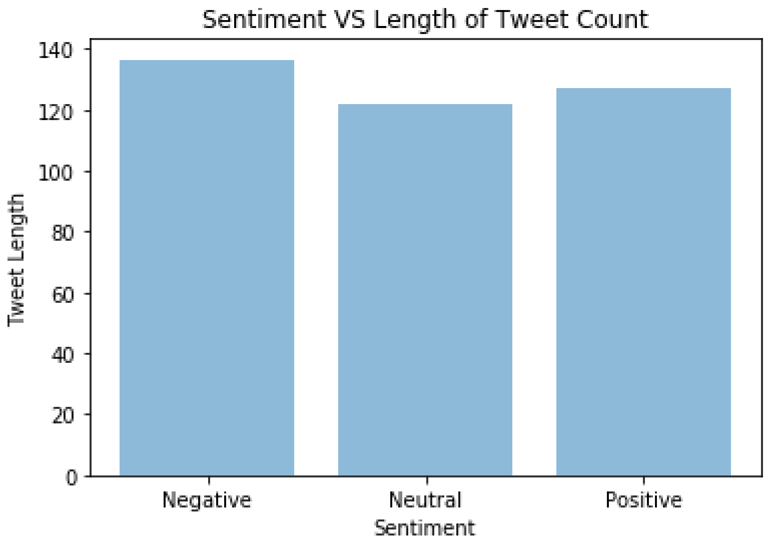


Analysis on sentiment V/S retweet count

There is a clear difference in the retweet count based on sentiment. As the above barchart shows, the most retweeted tweets on brexit are the Positive tweets, next are the neutral tweets and the least retweeted are the Negative tweets

In [24]:

```
# Analysis on Sentiment V/S length of the tweet
length = tweets_data.groupby('Sentiment').mean().Length

# Ploting
y_pos = np.arange(len(length))

plt.bar(y_pos, length, align='center', alpha=0.5)
plt.xticks(y_pos, length.index)
plt.xlabel('Sentiment')
plt.ylabel('Tweet Length')

plt.title('Sentiment VS Length of Tweet Count')

plt.show()
```

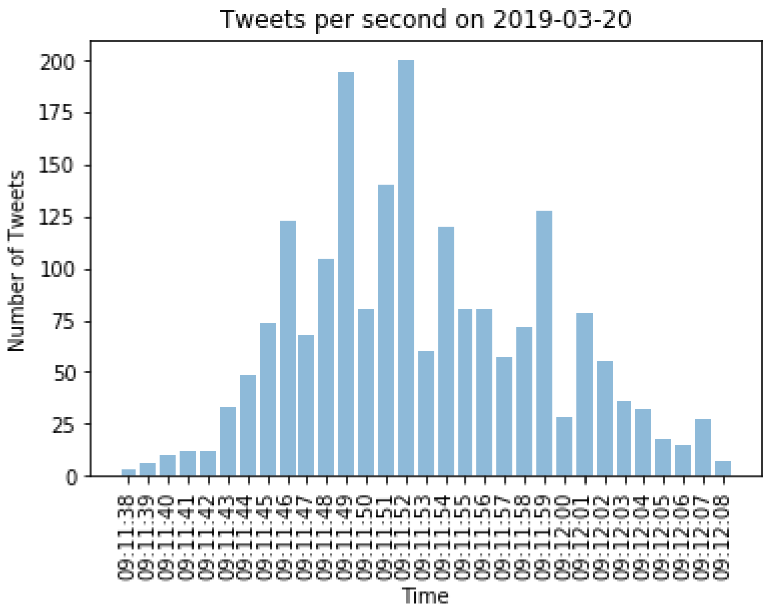


Analysis on Sentiment V/S length of the tweet

As described by the above bar chart Negative tweets tends to be lengthier than the positive or neutral tweets

In [48]:

```
# Analysis on number of tweets per second
second = tweets_data.Date.value_counts().sort_index()

# Plotting
y_pos = np.arange(len(second))

plt.bar(y_pos, second, align='center', alpha=0.5)
plt.xticks(y_pos, [w[11:] for w in second.index], rotation=90)
plt.xlabel('Time')
plt.ylabel('Number of Tweets')

plt.title('Tweets per second on ' + second.index[0][:10])

plt.show()
```



Analysis on number of tweets per second

The above bar chart shows number of tweets per second, the sudden rise in the number of tweets could be guessed as the reaction to news reports or official announcements