DATA 602: INTRODUCTION TO MACHINE LEARNING SENTIMENT ANALYSIS FOR TWITTER DATA (USA PRESIDENTIAL ELECTION) PROJECT GROUP:SENTIMENT ANALYSIS -2

In [63]:
```
from sklearn.metrics import classification_report, f1_score, accuracy_score, c
onfusion_matrix
from sklearn.metrics import  accuracy_score, confusion_matrix, classification_
report
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer,
TfidfTransformer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from nltk.stem.lancaster import LancasterStemmer
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
from spacy.lang.en.stop_words import STOP_WORDS
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC, LinearSVC
from sklearn.pipeline import Pipeline
from gensim.models import Word2Vec
from nltk.corpus import stopwords
from textblob import TextBlob
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from wordcloud import STOPWORDS
from pandas import Series
import networkx as nx
from PIL import Image
import seaborn as sns
import datetime as dt
import nltk, string
import pandas as pd
import numpy as np
import nltk as nlp
import datetime
import warnings
import calendar
import sys
import os
import re
import pandas as pd


import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from sklearn.utils import resample
from sklearn.utils import shuffle
```

```
from sklearn.metrics import confusion_matrix,classification_report
import re
```

```
Using TensorFlow backend.
```

In [2]:
```
biden_data=pd.read_csv(r"C:\Users\aslam\Downloads\602_project\biden_tweets.csv")
biden_data['handle']=np.nan
biden_data['handle'].fillna(value='JoeBiden',inplace=True)
biden_data.head()
```

Out[2]:

	id	created_at	text	retweet_count	favorite_count	hand
0	1316189479544279041	2020-10-14 01:30:00	Thank you, Florida! https://t.co/liQYRmvY7J	8556	67603	JoeBide
1	1316178155510652929	2020-10-14 00:45:00	I'll be a president for all Americans. Not jus...	17627	151476	JoeBide
2	1316163055760232449	2020-10-13 23:45:00	Folks, it's hard to believe, but tomorrow nigh...	2077	7397	JoeBide
3	1316155575511966721	2020-10-13 23:16:00	Here's something that will be very different i...	11944	72533	JoeBide
4	1316147956689575936	2020-10-13 22:45:00	We all know President Trump has a tendency to ...	7897	23308	JoeBide

In [3]:
```
x=biden_data['id']
type(x)
```

Out[3]:
```
pandas.core.series.Series
```

In [4]:
```
biden_data.drop_duplicates(inplace=True)
biden_data.dropna(inplace=True)
biden_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3077 entries, 0 to 3076
Data columns (total 6 columns):
id                3077 non-null int64
created_at        3077 non-null object
text              3077 non-null object
retweet_count     3077 non-null int64
favorite_count    3077 non-null int64
handle            3077 non-null object
dtypes: int64(3), object(3)
memory usage: 168.3+ KB
```

In [5]:
```
biden_data.drop(biden_data.index[1058:3078],0,inplace=True)
biden_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1058 entries, 0 to 1057
Data columns (total 6 columns):
id                1058 non-null int64
created_at        1058 non-null object
text              1058 non-null object
retweet_count     1058 non-null int64
favorite_count    1058 non-null int64
handle            1058 non-null object
dtypes: int64(3), object(3)
memory usage: 57.9+ KB
```

In [6]:
```
trump_data=pd.read_csv(r"C:\Users\aslam\Downloads\602_project\trump_tweets.csv")
trump_data['handle']=np.nan
trump_data['handle'].fillna(value='realDonaldTrump',inplace=True)
trump_data.head()
```

Out[6]:

	id	created_at	text	retweet_count	favorite_count	
0	1316221805133279232	2020-10-14 03:38:27	https://t.co/75J7mUX0Iy	1155	3110	realDo
1	1316214253527986178	2020-10-14 03:08:26	I will never let you down! #MAGA https://t.co/...	6712	27177	realDo
2	1316194625405751296	2020-10-14 01:50:27	https://t.co/wJN4zv0y8O	46280	207008	realDo
3	1316188335862247424	2020-10-14 01:25:27	For years you had a President who apologized f...	13196	49412	realDo
4	1316187932961587200	2020-10-14 01:23:51	Proud citizens like you helped build this Coun...	9567	35601	realDo

In [7]:
```
trump_data.drop_duplicates(inplace=True)
trump_data.dropna(inplace=True)
trump_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1058 entries, 0 to 1057
Data columns (total 6 columns):
id                1058 non-null int64
created_at        1058 non-null object
text              1058 non-null object
retweet_count     1058 non-null int64
favorite_count    1058 non-null int64
handle            1058 non-null object
dtypes: int64(3), object(3)
memory usage: 57.9+ KB
```

In [8]:
```
trump_data['time_decoded'] = pd.to_datetime(trump_data.created_at)
trump_data['time_decoded'] = trump_data.time_decoded.map(lambda x: x.strftime(
'%Y-%m-%d'))
trump_data[['created_at', 'time_decoded']].head()
```

Out[8]:

	created_at	time_decoded
0	2020-10-14 03:38:27	2020-10-14
1	2020-10-14 03:08:26	2020-10-14
2	2020-10-14 01:50:27	2020-10-14
3	2020-10-14 01:25:27	2020-10-14
4	2020-10-14 01:23:51	2020-10-14

In [9]:
```
biden_data['time_decoded'] = pd.to_datetime(biden_data.created_at)
biden_data['time_decoded'] = biden_data.time_decoded.map(lambda x: x.strftime(
'%Y-%m-%d'))
biden_data[['created_at', 'time_decoded']].head()
```

Out[9]:

	created_at	time_decoded
0	2020-10-14 01:30:00	2020-10-14
1	2020-10-14 00:45:00	2020-10-14
2	2020-10-13 23:45:00	2020-10-13
3	2020-10-13 23:16:00	2020-10-13
4	2020-10-13 22:45:00	2020-10-13

In [10]:
```
data=pd.concat([biden_data,trump_data])
data.rename(columns={'created_at':'time'},inplace=True)
data.head()
```

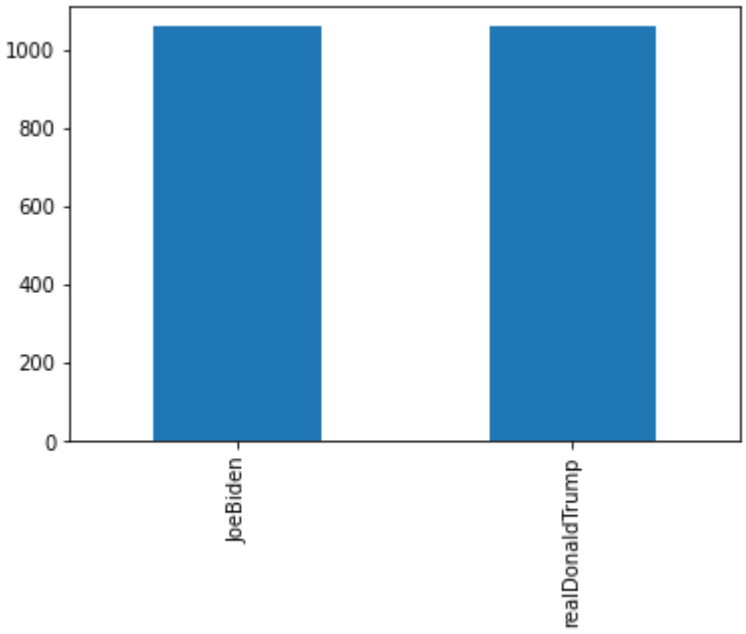
Out[10]:

	id	time	text	retweet_count	favorite_count	handle
0	1316189479544279041	2020-10-14 01:30:00	Thank you, Florida! https://t.co/liQYRmvY7J	8556	67603	JoeBiden
1	1316178155510652929	2020-10-14 00:45:00	I'll be a president for all Americans. Not jus...	17627	151476	JoeBiden
2	1316163055760232449	2020-10-13 23:45:00	Folks, it's hard to believe, but tomorrow nigh...	2077	7397	JoeBiden
3	1316155575751196672 1	2020-10-13 23:16:00	Here's something that will be very different i...	11944	72533	JoeBiden
4	1316147956689575936	2020-10-13 22:45:00	We all know President Trump has a tendency to ...	7897	23308	JoeBiden

In [11]:
```
# Let's check if the data is balanced
print(data.handle.value_counts())
data.handle.value_counts().plot(kind='bar')
```

```
JoeBiden          1058
realDonaldTrump   1058
Name: handle, dtype: int64
```

Out[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1eff5e138c8>
```



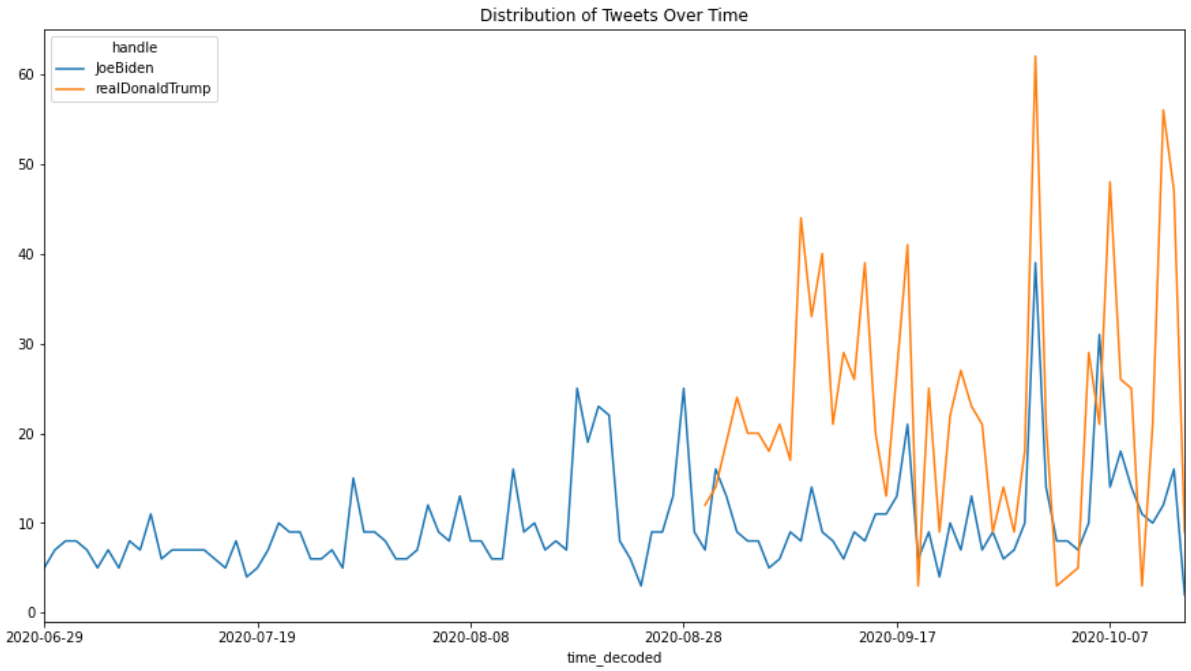
In [12]:
```
data['count']=1
grouped = data.groupby(['time_decoded', 'handle'])
grouped = grouped['count'].sum().reset_index()
grouped.tail()
```

Out[12]:

	time_decoded	handle	count
149	2020-10-12	realDonaldTrump	56
150	2020-10-13	JoeBiden	16
151	2020-10-13	realDonaldTrump	47
152	2020-10-14	JoeBiden	2
153	2020-10-14	realDonaldTrump	9

In [13]:
```
# Let's look at tweets over time
grouped.pivot(index='time_decoded', columns='handle', values='count').plot(fig
size=(15, 8), title='Distribution of Tweets Over Time')
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1eff622b5c8>

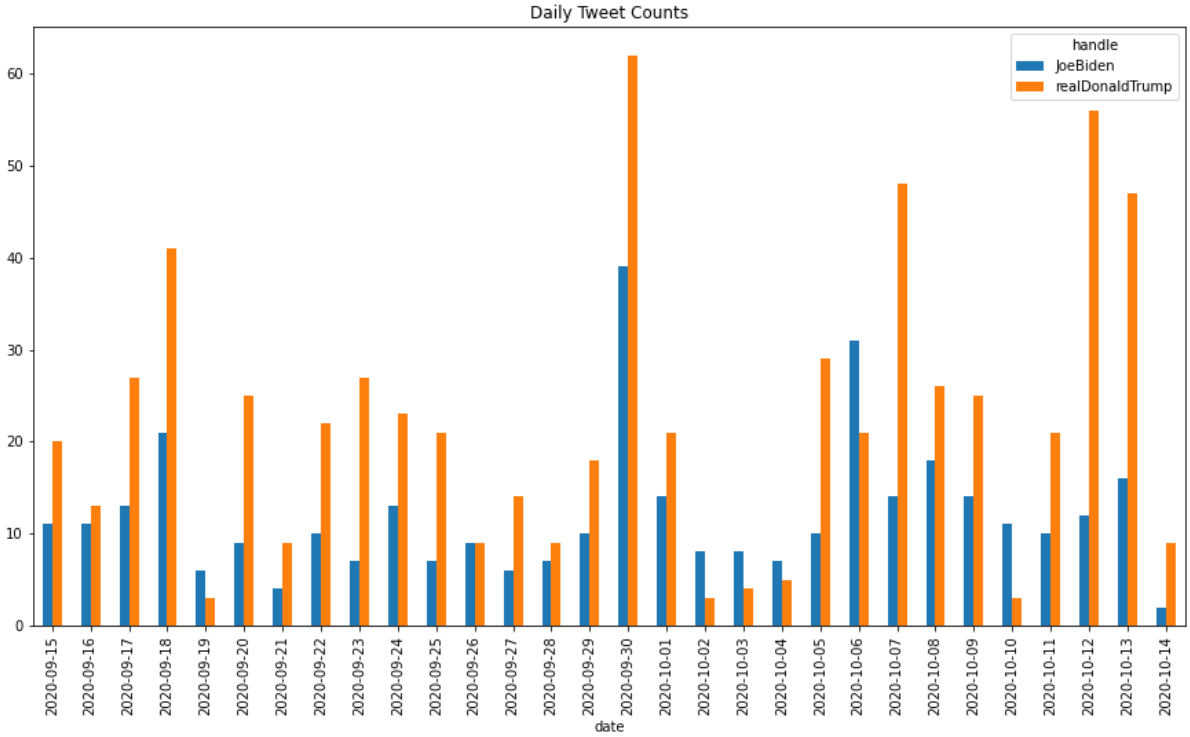


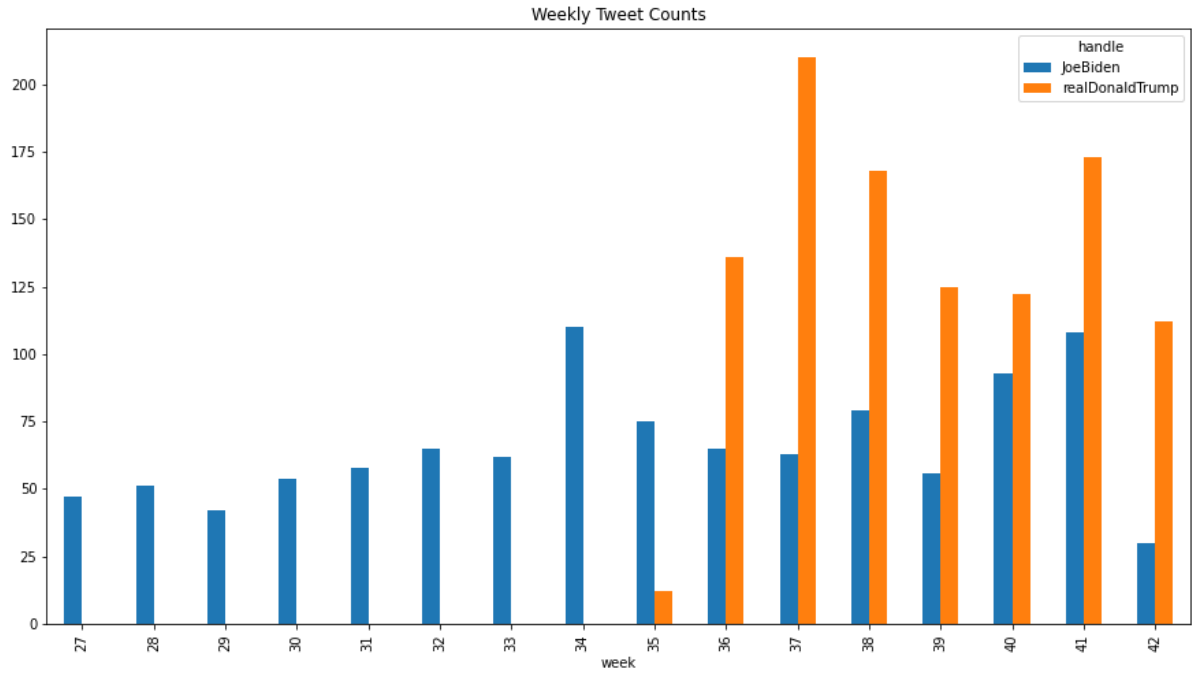
In [14]:
```
data['time'] = pd.to_datetime(data.time)
data['date']= data.time.apply(lambda x: x.date())
data['week']= data.time.apply(lambda x: x.isocalendar()[1])
data['tweet_hour'] = data.time.apply(lambda x: (x).hour)

# filter retweets and dates after 4/18/2016
data_ex_rt_daily = data.groupby(['date', 'handle']).size().unstack()
data_ex_rt_weekly = data.groupby(['week', 'handle']).size().unstack()
```

In [15]:
```
data_ex_rt_daily.tail(30).plot(kind='bar', title='Daily Tweet Counts', figsize
=(15, 8) )
data_ex_rt_weekly.plot(kind='bar', title='Weekly Tweet Counts', figsize=(15, 8
))

data_ex_rt_daily.describe()
```

Out[15]:

handle	JoeBiden	realDonaldTrump
count	108.000000	46.000000
mean	9.796296	23.000000
std	5.614999	13.790496
min	2.000000	3.000000
25%	7.000000	14.000000
50%	8.000000	21.000000
75%	11.000000	27.000000
max	39.000000	62.000000

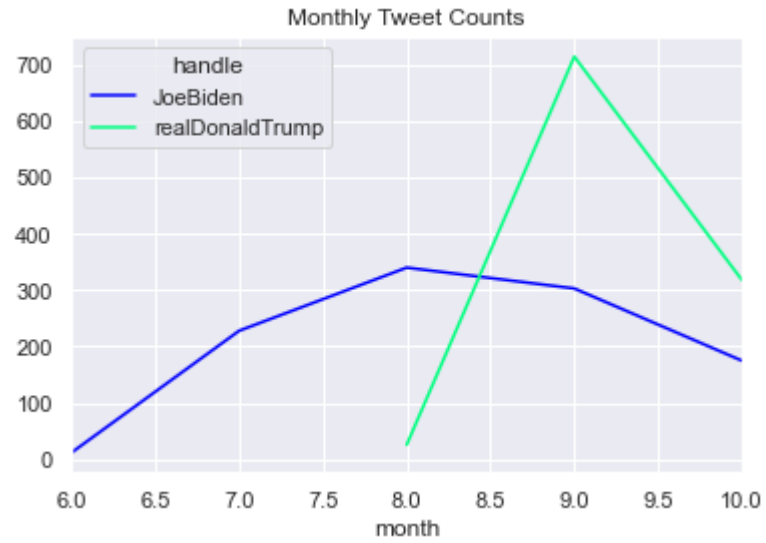




Number of tweets by the months

In [16]:
```
data['month'] = data['time'].apply(lambda x: x.month)
sns.set(font_scale=1)
monthly_tweets = data.groupby(['month', 'handle']).size().unstack()
monthly_tweets.plot(title='Monthly Tweet Counts', colormap='winter')
```

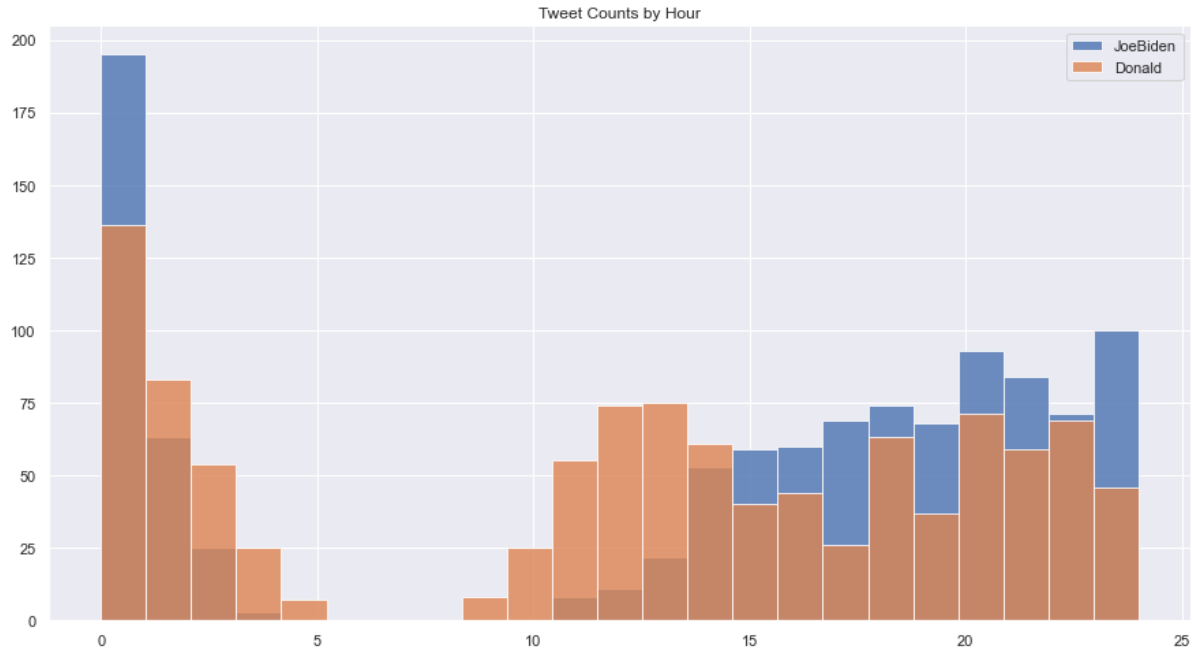
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1eff6c1fe48>



Biden has tweeted 1058 tweets in 4 months where as trump did it in just months

When do candidates tweet?

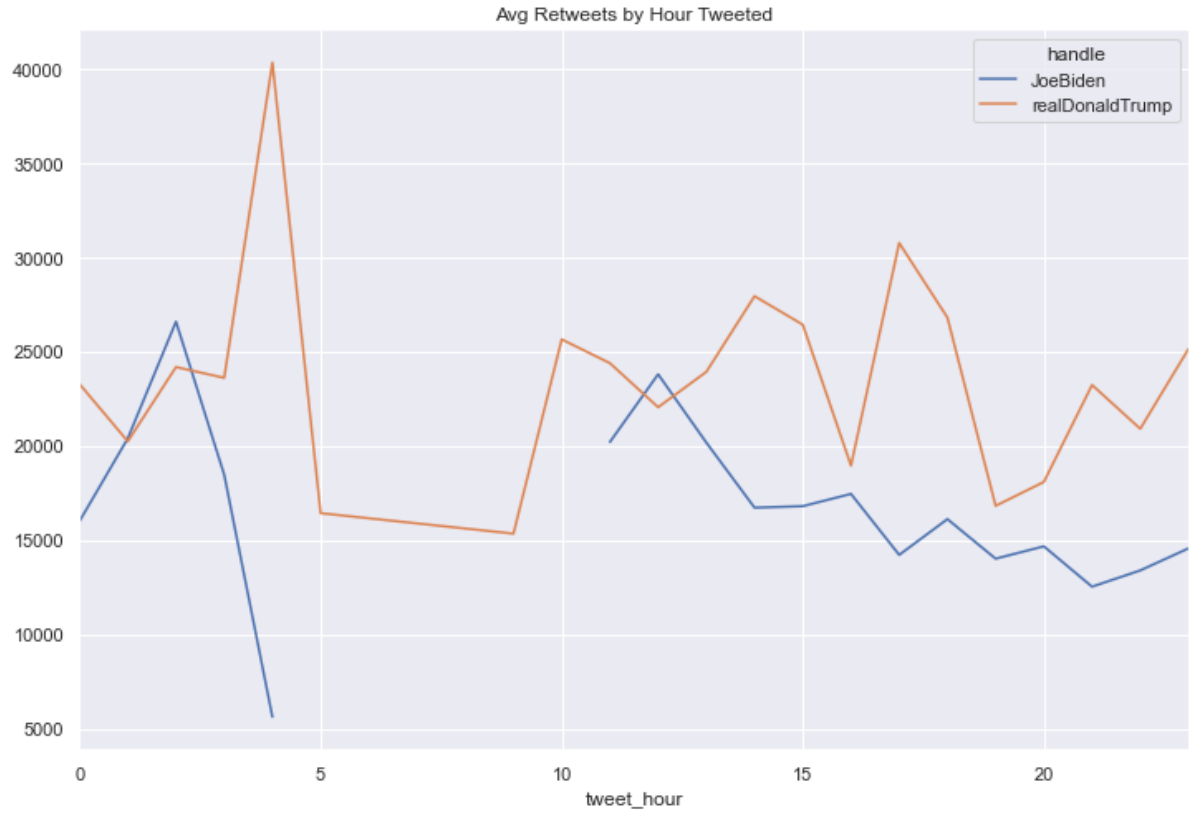
In [17]:
```
bins = np.linspace(0, 24, 24)
plt.figure(figsize=(15, 8))
plt.hist(data['tweet_hour'][data['handle']=='JoeBiden'], bins, alpha=0.8, label="JoeBiden")
plt.hist(data['tweet_hour'][data['handle']=='realDonaldTrump'], bins, alpha=0.8, label="Donald")
plt.legend()
plt.title('Tweet Counts by Hour')
plt.show()
```



Does time of tweet correlate with likes ?

In [18]:
```
data_hour_retweets = data.groupby(['tweet_hour', 'handle']).apply(lambda x: np
.mean(x.retweet_count)).unstack()
data_hour_retweets.plot(title='Avg Retweets by Hour Tweeted', figsize=(12, 8))
```

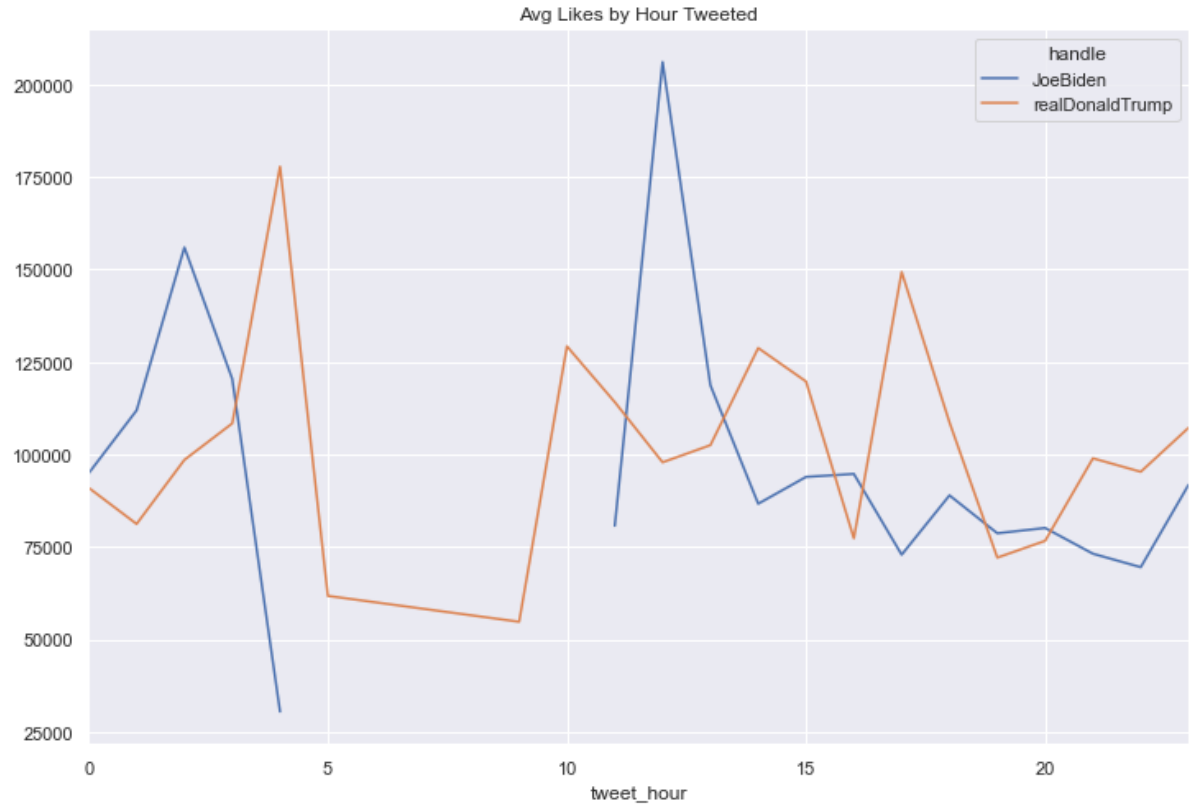
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1eff68032c8>



Does time of tweet correlate with likes ?

In [19]:
```
data_hour_likes = data.groupby(['tweet_hour', 'handle']).apply(lambda x: np.mean(x.favorite_count)).unstack()
data_hour_likes.plot(title='Avg Likes by Hour Tweeted', figsize=(12, 8))
```

Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x1eff6b48c48>



In [ ]:
```

```

In [ ]:
```

```

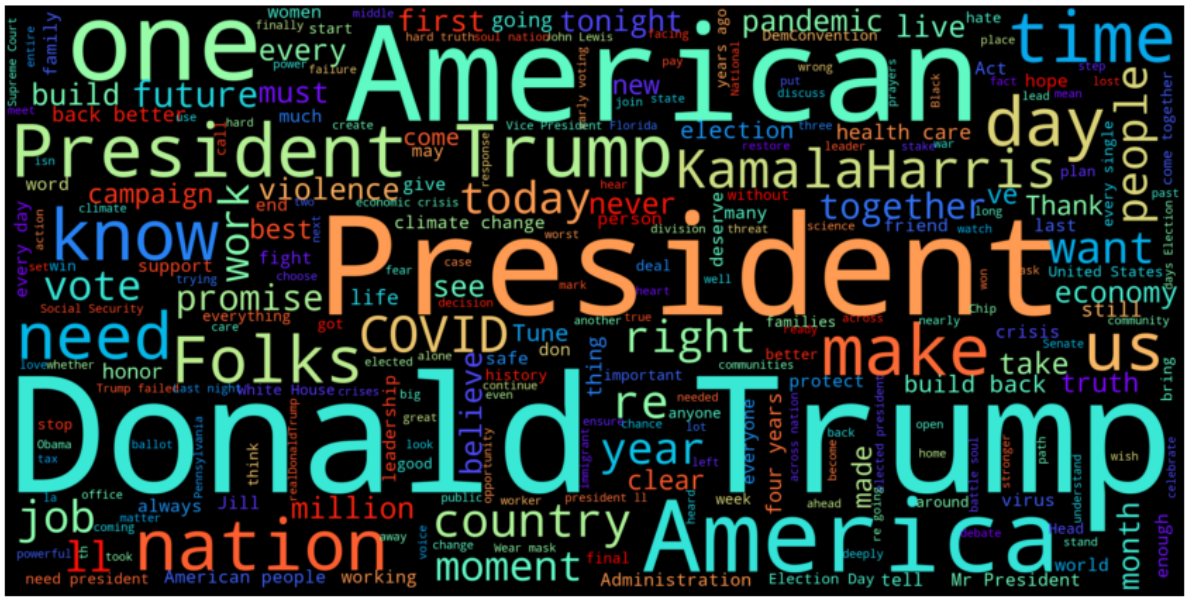
In [20]:
```
trump_tweets = [' '.join(t.strip().split()) for t in trump_data.text.tolist()]
biden_tweets = [' '.join(t.strip().split()) for t in biden_data.text.tolist()]
```

In [21]:
```
stopwords = set(STOPWORDS)
stopwords.add("http")
stopwords.add("https")
stopwords.add("amp")
stopwords.add("CO")
stopwords.add("will")
stopwords.add("say")
stopwords.add("said")
stopwords.add("let")
stopwords.add("now")
stopwords.add("go")
```

Word Cloud :

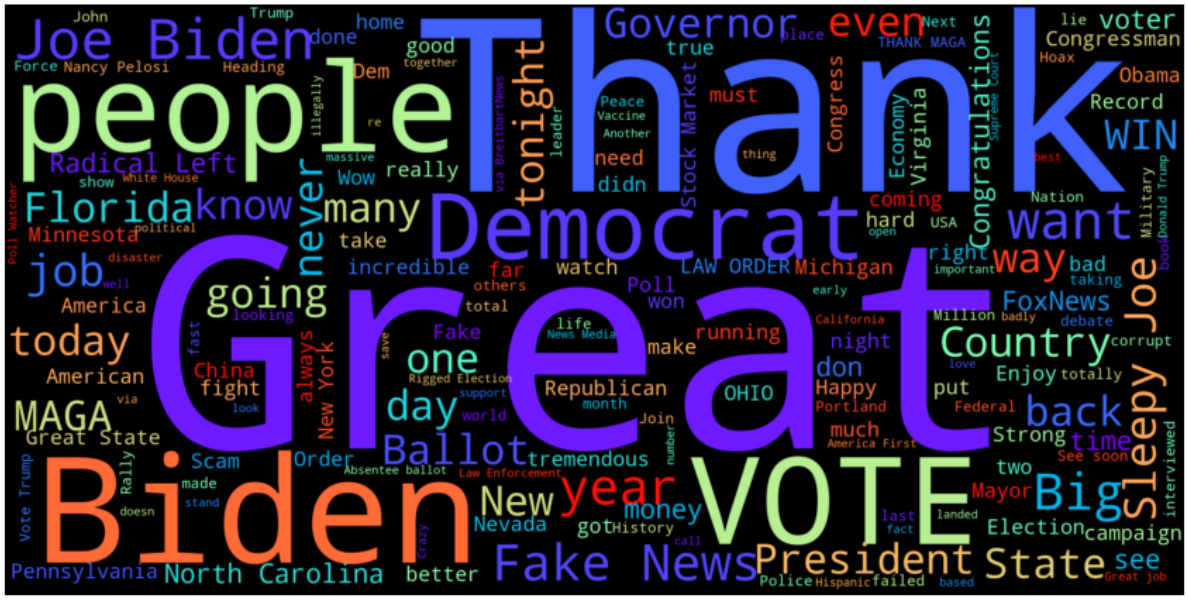
Joe Biden's Word Cloud :

In [22]:
```
wordcloud_hc = WordCloud(max_font_size=100, max_words=1000, scale=6, relative_
scaling=.6,stopwords=stopwords, background_color="black", colormap = "rainbow"
).generate(str(biden_tweets))
plt.figure(figsize=(15,10))
plt.imshow(wordcloud_hc,interpolation="bilinear")
plt.axis("off")
plt.show()
```



Donald Trump's Word Cloud :

In [23]:
```
wordcloud_hc = WordCloud(max_font_size=100, max_words=1000, scale=6, relative_
scaling=.6,stopwords=stopwords, background_color="black", colormap = "rainbow"
).generate(str(trump_tweets))
plt.figure(figsize=(15,10))
plt.imshow(wordcloud_hc,interpolation="bilinear")
plt.axis("off")
plt.show()
```



Mentions :

Extract mentions from tweets

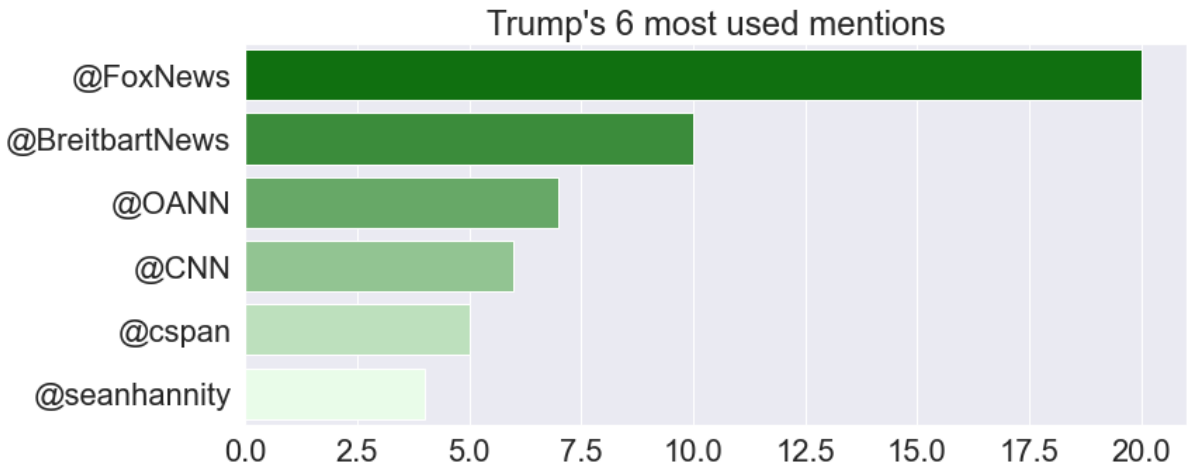
In [24]:
```
import re
def get_mentions(txt):
    mention = re.findall('(\@[A-Za-z_]+)', txt)
    if mention:
        return mention
    else:
        return ""
```

In [25]:
```
mention_list_trump = []
trump_data['top_mentions'] = trump_data['text'].apply(lambda x: get_mentions(x
))
for n in range(len(trump_data['top_mentions'])):
    mention_list_trump += trump_data['top_mentions'][n]
```

In [26]:
```
mention_list_biden = []
biden_data['top_mentions'] = biden_data['text'].apply(lambda x: get_mentions(x
))
for i in range(len(biden_data['top_mentions'])):
    mention_list_biden += biden_data['top_mentions'][i]
```

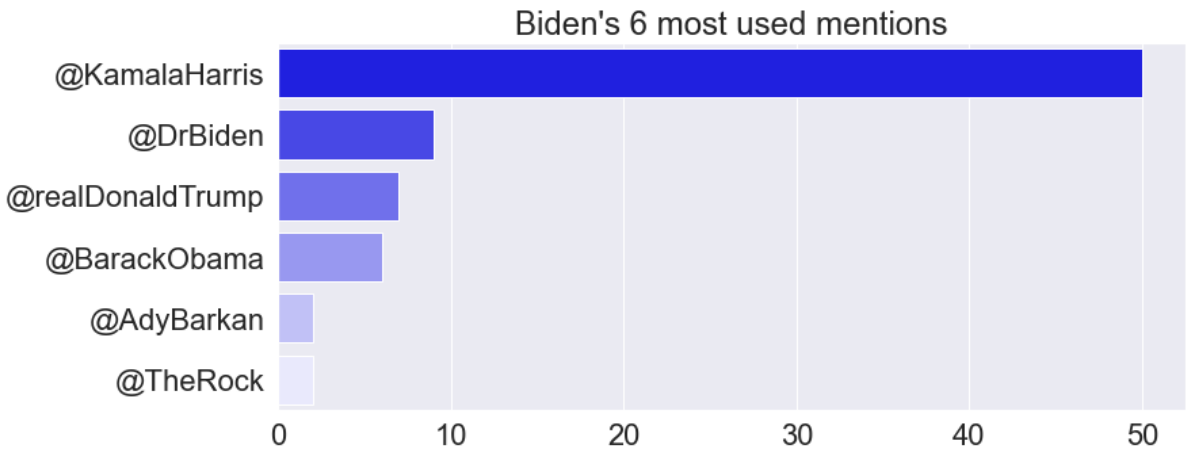
In [27]:
```
data1 = Series(mention_list_trump).value_counts().head(n=6)
sns.set_style("white")
sns.set(font_scale=2)
plt.figure(figsize=(12, 5))
sns.barplot(x=data1, y=data1.index, orient='h', palette=sns.light_palette("gre
en",reverse=True)).set_title("Trump's 6 most used mentions")
```

Out[27]:
```
Text(0.5, 1.0, "Trump's 6 most used mentions")
```



In [28]:
```
data2 = Series(mention_list_biden).value_counts().head(n=6)
sns.set_style("white")
sns.set(font_scale=2)
plt.figure(figsize=(12, 5))
sns.barplot(x=data2, y=data2.index, orient='h', palette=sns.light_palette("Blu
e",reverse=True)).set_title("Biden's 6 most used mentions")
```

Out[28]:
```
Text(0.5, 1.0, "Biden's 6 most used mentions")
```



In [ ]:
```
```

In [29]:
```
def get_hashtags(txt):
    hashtag = re.findall('(\#[A-Za-z_]+)', txt)
    if hashtag:
        return hashtag
    else:
        return ""
```

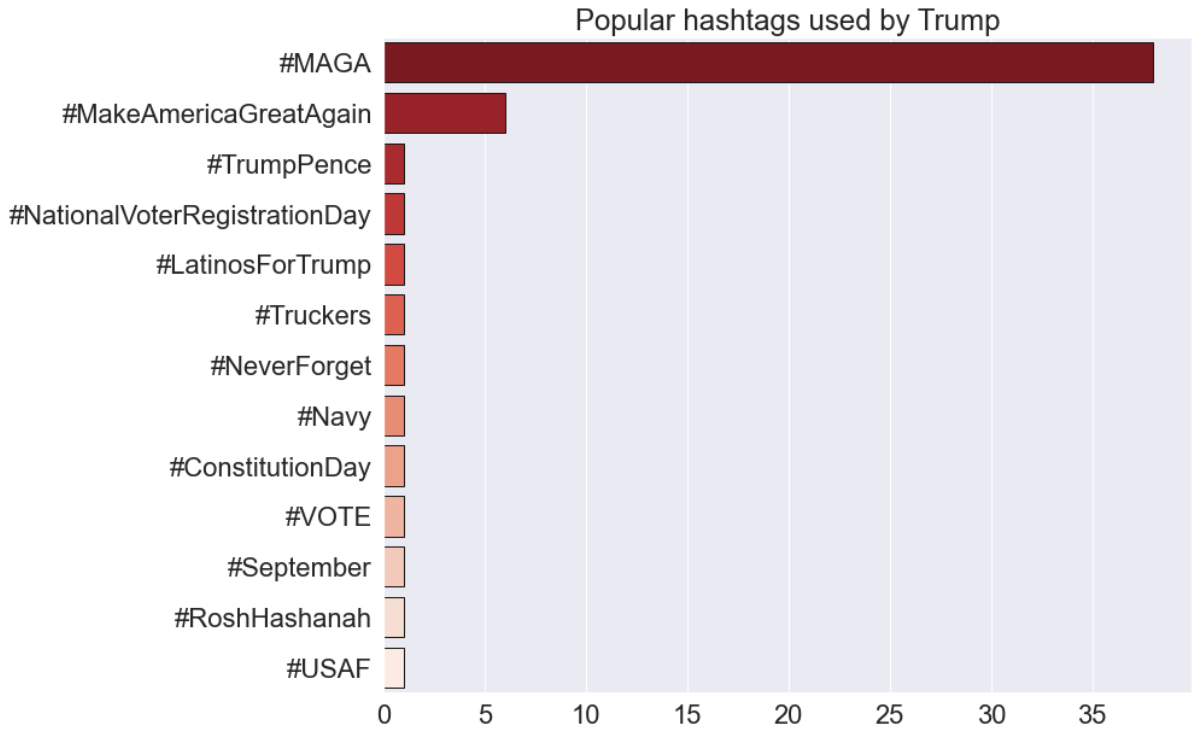
In [30]:
```
hashtag_list_trump = []
trump_data['top_hashtags'] = trump_data['text'].apply(lambda x: get_hashtags(x
))
for n in range(len(trump_data['top_hashtags'])):
    hashtag_list_trump += trump_data['top_hashtags'][n]
```

In [31]:
```
hashtag_list_biden = []
biden_data['top_hashtags'] = biden_data['text'].apply(lambda x: get_hashtags(x
))
for i in range(len(biden_data['top_hashtags'])):
    hashtag_list_biden += biden_data['top_hashtags'][i]
```

In [32]:
```
data4 = Series(hashtag_list_trump).value_counts().head(n=15)
sns.set_style("white")
sns.set(font_scale=2)
plt.figure(figsize=(12,10))
sns.barplot(x=data4, y=data4.index, orient='h', palette="Reds_r",linewidth = 1
, edgecolor = "k"* 25).set_title("Popular hashtags used by Trump")
```

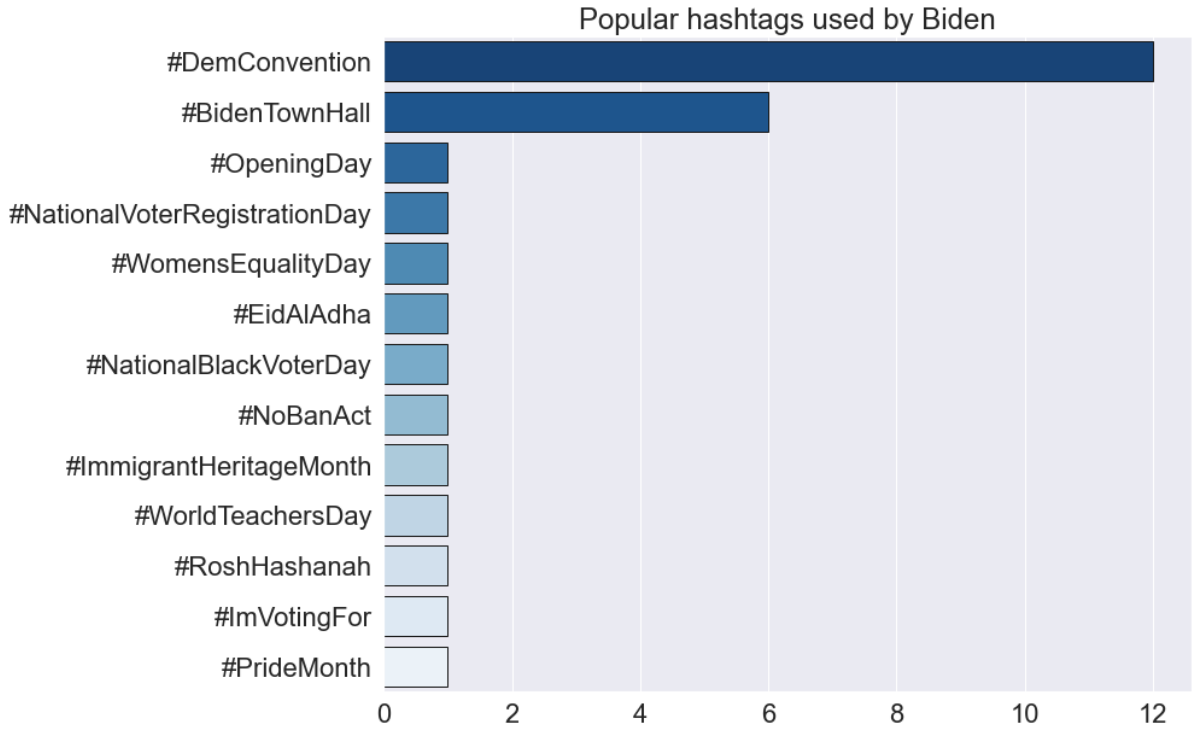
```
C:\Users\aslam\AppData\Local\Continuum\anaconda3\lib\site-packages\seaborn\ca
tegorical.py:1627: MatplotlibDeprecationWarning: Using a string of single cha
racter colors as a color sequence is deprecated. Use an explicit list instea
d.
  color=self.colors, align="center", **kws)
```

Out[32]:
```
Text(0.5, 1.0, 'Popular hashtags used by Trump')
```



In [33]:
```
data5 = Series(hashtag_list_biden).value_counts().head(n=15)
sns.set_style("white")
sns.set(font_scale=2)
plt.figure(figsize=(12,10))
sns.barplot(x=data5, y=data5.index, orient='h', palette="Blues_r",linewidth =
1 , edgecolor = "k"* 25).set_title("Popular hashtags used by Biden")
```

Out[33]:
```
Text(0.5, 1.0, 'Popular hashtags used by Biden')
```



In [34]:
```
B=nx.Graph() #create an empty graph

data6 = Series(hashtag_list_trump).value_counts().head(n=40)
data7 = Series(hashtag_list_biden).value_counts().head(n=40)

for hashtag1 in data6.index: #for each user loop over the hashtags they use
    B.add_edge('D.Trump',hashtag1) #add the edge Candidate<->hashtag

for hashtag2 in data7.index: #for each user loop over the hashtags they use
    B.add_edge('J.Biden',hashtag2) #add the edge Candidate<->hashtag
```

In [35]:
```
B.number_of_nodes()
```

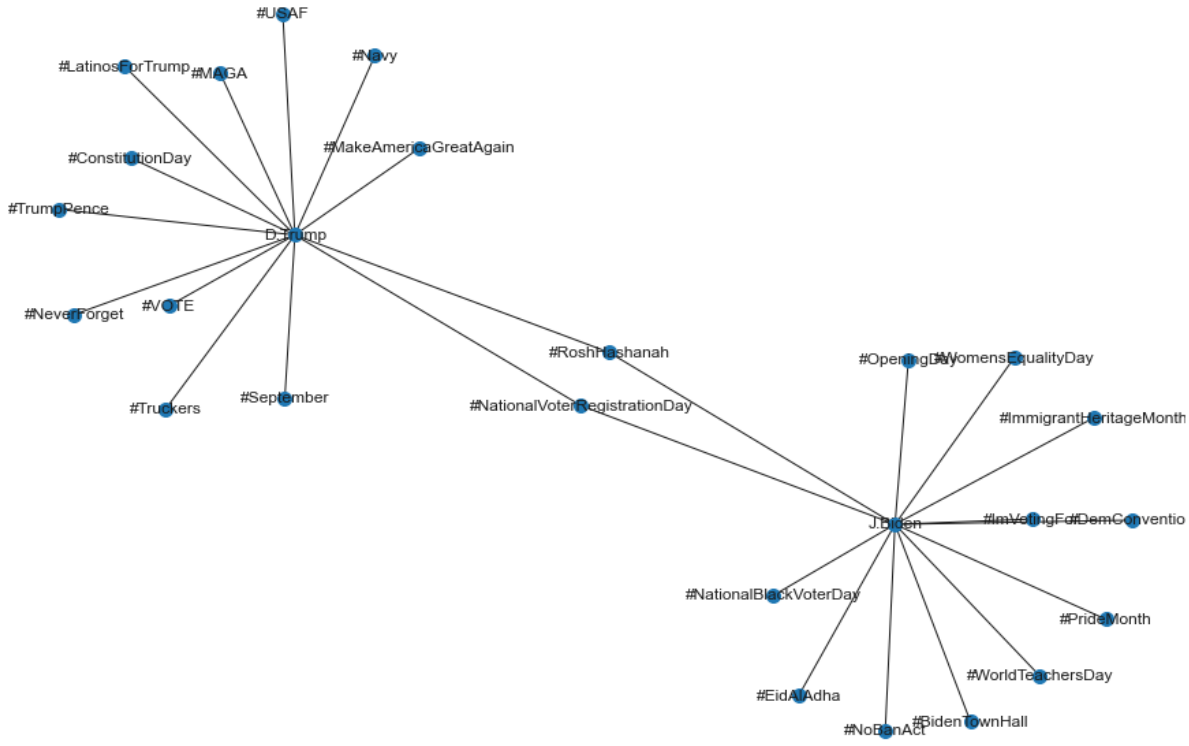
Out[35]:
```
26
```

In [36]:
```
B.number_of_edges()
```

Out[36]:
```
26
```

In [37]:
```
plt.figure(figsize=(12,8))
nx.draw_spring(B, with_labels=True, node_size= 100, font_size=12)
plt.show()
```

```
C:\Users\aslam\AppData\Local\Continuum\anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:579: MatplotlibDeprecationWarning:
The iterable function was deprecated in Matplotlib 3.1 and will be removed in 3.3. Use np.iterable instead.
  if not cb.iterable(width):
```

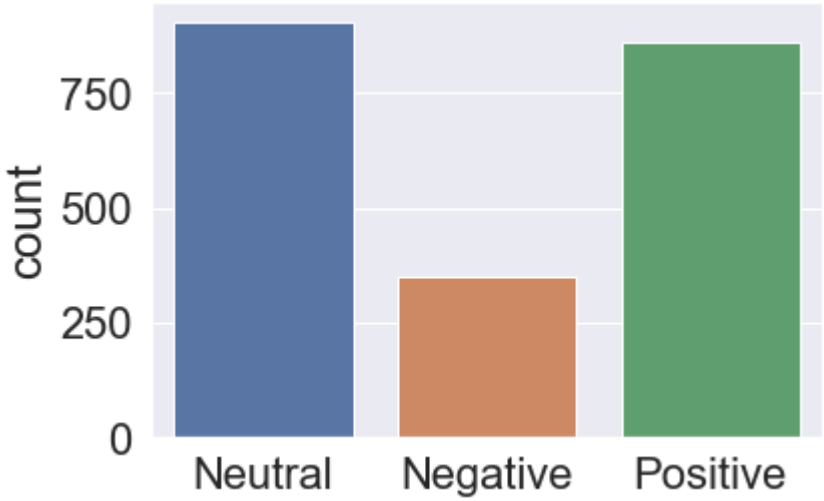


In [38]:
```
from textblob import TextBlob, Word, Blobber
from textblob.classifiers import NaiveBayesClassifier
from textblob.taggers import NLTKTagger
from collections import Mapping
bloblist_desc = []

df_tweet_descr_str=data['text'].astype(str)
for row in df_tweet_descr_str:
    blob = TextBlob(row)
    bloblist_desc.append((row,blob.sentiment.polarity, blob.sentiment.subjecti
vity))
    df_tweet_polarity_desc = pd.DataFrame(bloblist_desc, columns = ['sentence'
,'sentiment','polarity'])
d=df_tweet_polarity_desc['sentiment']
def f(d):
    if d > 0:
        val = "Positive"
    elif d == 0:
        val = "Neutral"
    else:
        val = "Negative"
    return val
```

```
C:\Users\aslam\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_
launcher.py:4: DeprecationWarning: Using or importing the ABCs from 'collecti
ons' instead of from 'collections.abc' is deprecated, and in 3.8 it will stop
working
  after removing the cwd from sys.path.
```

In [39]:
```
polarity=[]
for value in d:
    polarity.append(f(value))
```

In [40]:
```
import seaborn as sns
ax = sns.countplot(x=polarity, data=df_tweet_polarity_desc)
```

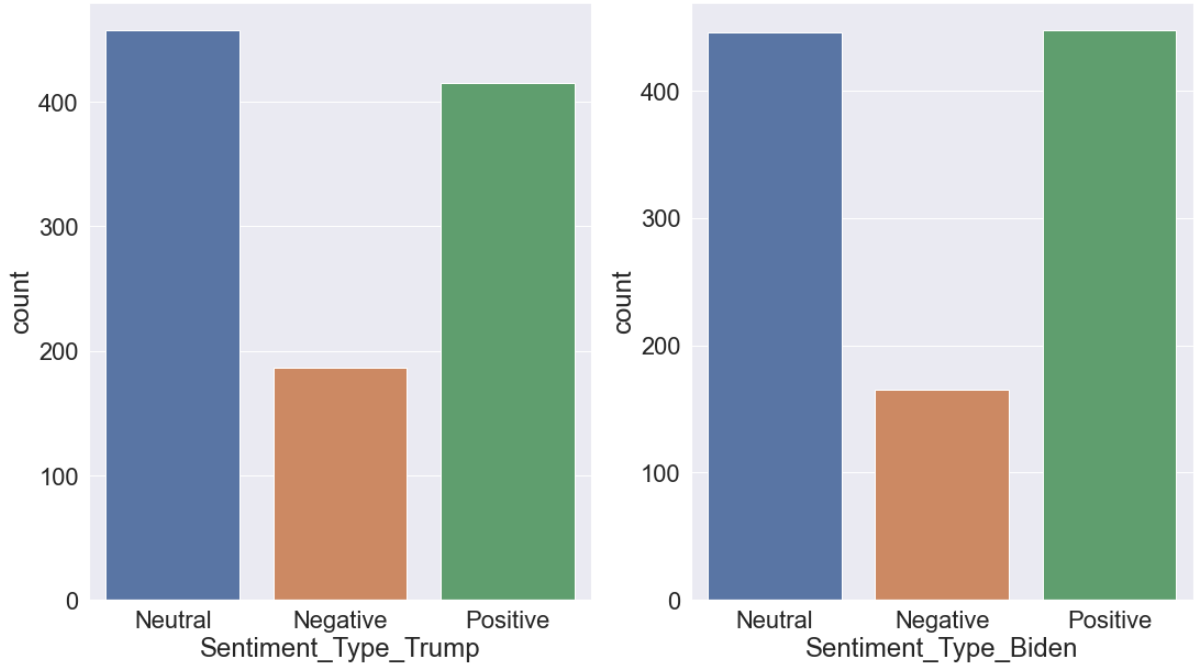


In [41]:
```
tweet_biden = data.loc[(data['handle']=='JoeBiden'), ['text']]
bloblist_desc = list()

df_tweet_biden_str=tweet_biden['text'].astype(str)
for row in df_tweet_biden_str:
    blob = TextBlob(row)
    bloblist_desc.append((row,blob.sentiment.polarity, blob.sentiment.subjecti
vity))
    df_tweet_biden_polarity_desc = pd.DataFrame(bloblist_desc, columns = ['sen
tence','sentiment','polarity'])


def f(df_tweet_biden_polarity_desc):
    if df_tweet_biden_polarity_desc['sentiment'] > 0:
        val = "Positive"
    elif df_tweet_biden_polarity_desc['sentiment'] == 0:
        val = "Neutral"
    else:
        val = "Negative"
    return val

df_tweet_biden_polarity_desc['Sentiment_Type_Biden'] = df_tweet_biden_polarity
_desc.apply(f, axis=1)
```

In [42]:
```
tweet_trump = data.loc[(data['handle']=='realDonaldTrump'), ['text']]
bloblist_desc = list()

df_tweet_trump_str=tweet_trump['text'].astype(str)
for row in df_tweet_trump_str:
    blob = TextBlob(row)
    bloblist_desc.append((row,blob.sentiment.polarity, blob.sentiment.subjecti
vity))
    df_tweet_trump_polarity_desc = pd.DataFrame(bloblist_desc, columns = ['sen
tence','sentiment','polarity'])


def f(df_tweet_trump_polarity_desc):
    if df_tweet_trump_polarity_desc['sentiment'] > 0:
        val = "Positive"
    elif df_tweet_trump_polarity_desc['sentiment'] == 0:
        val = "Neutral"
    else:
        val = "Negative"
    return val

df_tweet_trump_polarity_desc['Sentiment_Type_Trump'] = df_tweet_trump_polarity
_desc.apply(f, axis=1)
```

In [43]:
```
fig, ax =plt.subplots(1,2,figsize=(18,10))
sns.set_style("whitegrid")
sns.set(font_scale=1.2)
sns.countplot(x="Sentiment_Type_Trump", data=df_tweet_trump_polarity_desc,  ax
=ax[0])
sns.countplot(x="Sentiment_Type_Biden", data=df_tweet_biden_polarity_desc, ax=
ax[1])
fig.show()
```

```
C:\Users\aslam\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_
launcher.py:6: UserWarning: Matplotlib is currently using module://ipykernel.
pylab.backend_inline, which is a non-GUI backend, so cannot show the figure.
```



In [44]:
```
import markovify as mk

tweets = data[['handle', 'text']]

def tweet(tweeter):
    doc = tweets[tweets.handle.str.contains(tweeter)].text.tolist()
    text_model = mk.Text(doc)
    print('\n', tweeter)
    for i in range(8):
        print(text_model.make_sentence(tries=100))

tweet('Biden')
tweet('Donald')
```

```
 Biden
I made some comments about diversity in the country, @KamalaHarris took on th
e passing of Justice Ruth Bader Ginsburg and the existential threat of climat
e change poses an existenti… https://t.co/fAHm6sME1n
We have to meet it together as a nation true to our highest ideals… https://
t.co/L9Y5w9AkyY
To everyone celebrating the Hindu festival of Ganesh Chaturthi in the path of
becoming angrier, less hopeful, and more economic pain and anxiety fo… http
s://t.co/zDYW2IWnwJ
Tune in as I deliver remarks on the ballot, and your voice — and we need a pr
esident who will choose science over fiction.
.@AndrewCuomo, thank you for all of us to get complacent.
We have to worry about my friend and running mate, @KamalaHarris. https://t.c
o/FYPXV5dZZG
The fact is, Donald Trump wants to ask her to be based on the disparities tha
t have plagued our c… https://t.co/qLG7QT9lgF
But I will re… https://t.co/qNgyOKHOlE

 Donald
Heading to the Prime Minister @BorisJohnson of the WORST governors in the US
A.
Governor Whitmer of Michigan jobs.
Didn’t we go through the Election.
We are bringing it back from China and foreign countri… https://t.co/AJOYO6Nh
5I
Jim, to late to take advantage of fools.
...Importantly, we also won on the SCAM? https://t.co/ynEAchAWPA
At no time before has there been a wacko for years, and hundreds of times a y
ear.
Just out: Some people in the Great States of America will NOT be cancelling i
ts contract with Catholic Priests who… https://t.co/18c9BE8E82
```

In [67]:
```
def subj_tweet(tweeter, subject):
    doc = tweets[tweets.handle.str.contains(tweeter)].text.tolist()
    text_model = mk.Text(doc)
    print('\n', tweeter)
    for i in range(8):
        print(text_model.make_sentence_with_start(subject, strict=False))

subj_tweet('Biden', 'Trump')
subj_tweet('Donald', 'Biden')
```

```
 Biden
Trump knew how deadly COVID-19 was and did nothing — or the conviction o… htt
ps://t.co/PKdJlyJPP4
Trump administration another four years in the land of the Build Back Better
Express Tour… https://t.co/2OPnM0YPL1
Trump put our nation unprepared and unprotected for the first confirmed case
of COVID-19 in the back by police.
Trump and First Lady Melania Trump for the little guy, and one… https://t.co/
gnzCegPMfl
Trump put our nation on the best partner I could have acted months ago to cur
b this pandemic—it's obvious he still hasn't le… https://t.co/Po0ghaQkyi
Trump says tonight you won't be safe in Joe Biden's America, look around and
ask him to do this: Listen to the way things were before these crises — we ha
ve to happen this way. https://t.co/27TL7PAhCi
Trump ignored the experts and leaders from… https://t.co/p8cLnYApjE
Trump Administration is in crisis.

 Donald
Biden wants to take advantage of the Great State of Texas.
Biden Accountable for his 47 years giving their jobs to China and other of he
r new book, "Firebrand: Dispatches from the beginning, way off in 2016.
Biden REFUSED to use the 10% Pumps for the American People!
Biden going to WIN.
Biden will also raise your taxes an… https://t.co/zjGoPf3dYV
Biden will also raise your Taxes, knock out your Second Amendment.
Biden is not on the massive number of Unsolicited &amp;amp; Solicited Ballots tha
t will be leaving office so… https://t.co/qYaWdsmgxv
Biden going to WIN.
```

In [46]:
```
#Create new dataframe for prediction model with candidate name (as label) and
 tweets (as message):
data['message'] = data['text'].apply(lambda x: x.lower().split('http')[0])
messages = data[['handle','message']]
```

In [47]:
```
print(messages[:5])
```

```
     handle                                            message
0  JoeBiden                                thank you, florida!
1  JoeBiden  i'll be a president for all americans. not jus...
2  JoeBiden  folks, it's hard to believe, but tomorrow nigh...
3  JoeBiden  here's something that will be very different i...
4  JoeBiden  we all know president trump has a tendency to ...
```

In [48]:
```
def split_into_tokens(message):
    message = message  # convert bytes into proper unicode
    return TextBlob(message).words
```

In [49]:
```
messages.message.head()
```

Out[49]:
```
0                                  thank you, florida!
1    i'll be a president for all americans. not jus...
2    folks, it's hard to believe, but tomorrow nigh...
3    here's something that will be very different i...
4    we all know president trump has a tendency to ...
Name: message, dtype: object
```

In [50]:
```
messages.message.head().apply(split_into_tokens)
```

Out[50]:
```
0                                [thank, you, florida]
1    [i, ', ll, be, a, president, for, all, america...
2    [folks, it, ', s, hard, to, believe, but, tomo...
3    [here, 's, something, that, will, be, very, di...
4    [we, all, know, president, trump, has, a, tend...
Name: message, dtype: object
```

In [51]:
```
import nltk
nltk.download('wordnet')
def split_into_lemmas(message):
    message = message.lower()
    words = TextBlob(message).words
    # for each word, take its "base form" = lemma
    return [word.lemma for word in words]

messages.message.head().apply(split_into_lemmas)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\aslam\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[51]:
```
0                                [thank, you, florida]
1    [i, ', ll, be, a, president, for, all, america...
2    [folk, it, ', s, hard, to, believe, but, tomor...
3    [here, 's, something, that, will, be, very, di...
4    [we, all, know, president, trump, ha, a, tende...
Name: message, dtype: object
```

In [52]:
```
bow_transformer = CountVectorizer(analyzer=split_into_lemmas).fit(messages['message'])
print(len(bow_transformer.vocabulary_))
print(bow_transformer.get_feature_names()[:10])
```

```
4551
["'d", "'em", "'ll", "'m", "'re", "'s", "'ve", '0.57', '1', '1,000']
```

In [53]:
```
messages_bow = bow_transformer.transform(messages['message'])
print('sparse matrix shape:', messages_bow.shape)
print('number of non-zeros:', messages_bow.nnz)
print('sparsity: %.2f%%' % (100.0 * messages_bow.nnz / (messages_bow.shape[0]
* messages_bow.shape[1])))
```

```
sparse matrix shape: (2116, 4551)
number of non-zeros: 31663
sparsity: 0.33%
```

In [54]:
```
tfidf_transformer = TfidfTransformer().fit(messages_bow)
print (tfidf_transformer.idf_[bow_transformer.vocabulary_['the']])
print (tfidf_transformer.idf_[bow_transformer.vocabulary_['help']])
```

```
1.7600503280062296
5.1920193683351386
```

In [55]:
```
messages_tfidf = tfidf_transformer.transform(messages_bow)
print(messages_tfidf.shape)
messages_tfidf
```

```
(2116, 4551)
```

Out[55]:
```
<2116x4551 sparse matrix of type '<class 'numpy.float64'>'
        with 31663 stored elements in Compressed Sparse Row format>
```

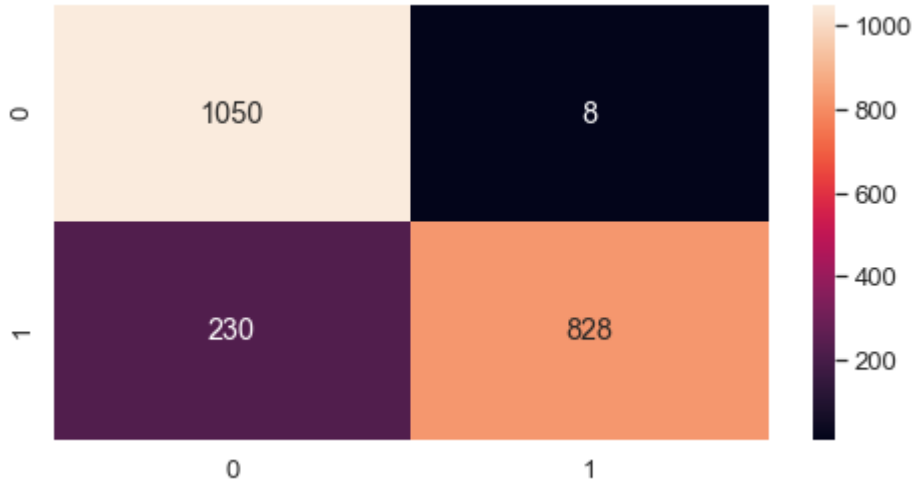
In [56]:
```
spam_detector = MultinomialNB().fit(messages_tfidf, messages['handle'])
all_predictions = spam_detector.predict(messages_tfidf)
tr_acc = accuracy_score(messages['handle'], all_predictions)
print("Accuracy on training set:  %.2f%%" % (100 * tr_acc))
```

```
Accuracy on training set:  88.75%
```

In [57]:
```
con_nai=confusion_matrix(messages['handle'], all_predictions)

print("Confusion Matrix \n ",con_nai)
```

```
Confusion Matrix
  [[1050    8]
 [ 230  828]]
```

In [58]:
```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(8,4))

sns.heatmap((con_nai), annot=True,fmt='d')
plt.show()
```



In [60]:
```
msg_train, msg_test, label_train, label_test = \
    train_test_split(messages['message'], messages['handle'], test_size=0.2, r
andom_state=1)
print(len(msg_train), len(msg_test), len(msg_train) + len(msg_test))
```

```
1692 424 2116
```

In [61]:
```
from sklearn.model_selection import cross_val_score
pipeline = Pipeline([('bow', CountVectorizer(analyzer=split_into_lemmas)),('tf
idf', TfidfTransformer()),('classifier', MultinomialNB())])
```

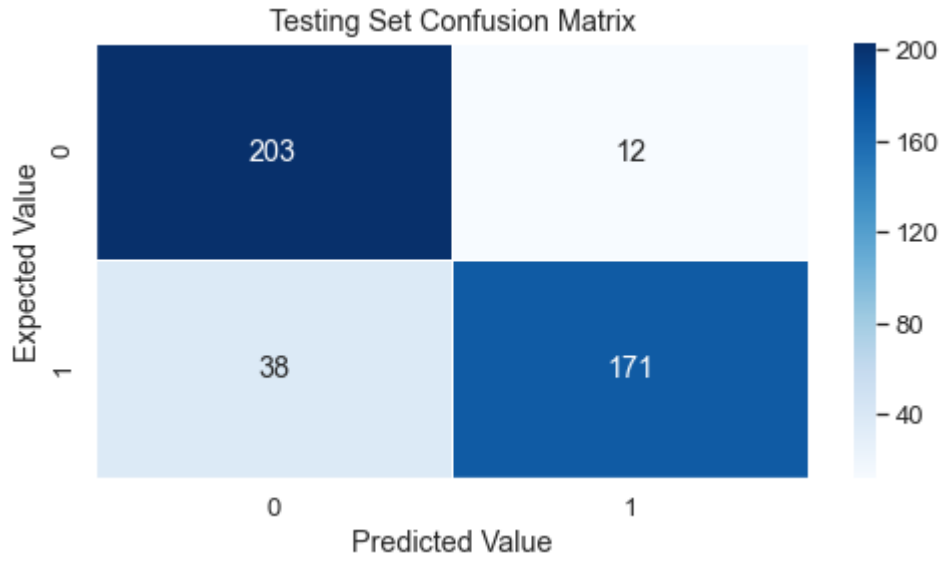
In [62]:
```
# steps to convert raw messages into models
# training data
# split data randomly into 10 parts: 9 for training, 1 for scoring
# which scoring metric?
# -1 = use all cores "faster"
scores = cross_val_score(pipeline,msg_train,label_train,cv=10,scoring='accurac
y',n_jobs=-1)
print(scores, '\n')
print('Mean score:', scores.mean())
print('Stdev:', scores.std())
```

```
[0.9        0.90588235 0.89411765 0.89940828 0.8816568  0.88757396
 0.87573964 0.89940828 0.90532544 0.89285714]

Mean score: 0.8941969568892645
Stdev: 0.009444448481210475
```

In [63]:
```
%time nb_detector = pipeline.fit(msg_train, label_train)
predictions = nb_detector.predict(msg_test)
```

```
Wall time: 994 ms
```

In [64]:
```
fig, ax = plt.subplots(figsize=(8,4))
sns.heatmap(confusion_matrix(label_test, predictions), annot=True, linewidths
=.5, ax=ax, cmap="Blues", fmt="d").set(xlabel='Predicted Value', ylabel='Expec
ted Value')
plt.title('Testing Set Confusion Matrix')
sns.despine(ax=None, top=True, right=True, left=False, bottom=False, offset=No
ne, trim=False)
```



In [65]:
```
my_1st_tweet = 'MAKE AMERICA GREAT AGAIN. VOTE!!!'
my_2nd_tweet = 'Now more than ever, we need a president who will choose compas
sion over cruelty.'
print("Tweet #1:", "'",my_1st_tweet, "'", ' \n \n', "I'm about %.0f%%" % (100
* max(nb_detector.predict_proba([my_1st_tweet])[0])), "sure this was tweeted b
y", nb_detector.predict([my_1st_tweet])[0])
print("Tweet #2:", "'",my_2nd_tweet, "'", ' \n \n', "I'm about %.0f%%" % (100
* max(nb_detector.predict_proba([my_2nd_tweet])[0])), "sure this was tweeted b
y", nb_detector.predict([my_2nd_tweet])[0])
```

```
Tweet #1: ' MAKE AMERICA GREAT AGAIN. VOTE!!! '

 I'm about 69% sure this was tweeted by realDonaldTrump
Tweet #2: ' Now more than ever, we need a president who will choose compassio
n over cruelty. '

 I'm about 86% sure this was tweeted by JoeBiden
```

In [66]:
```
your_tweet = 'This election is about so much more than policy. The character o
f our country is on the ballot.'

# ---- this part stays the same----
if your_tweet == 'ENTER YOUR TWEET HERE, INSIDE THE QUOTES':
    pass
else:
    print("Tweet #1:", "'",your_tweet, "'", ' \n \n', "I'm about %.0f%%" % (10
0 * max(nb_detector.predict_proba([your_tweet])[0])), "sure this was tweeted b
y", nb_detector.predict([your_tweet])[0])
```

```
Tweet #1: ' This election is about so much more than policy. The character of
our country is on the ballot. '

 I'm about 73% sure this was tweeted by JoeBiden
```

In [ ]:
```
we can use a Markov Chain Model to create synthetic tweets based on an existin
g library of actual tweets. Right now, its main use is for building Markov mod
els of large corpora of text and generating random sentences from that.
```

KEYWORDS DATASET

In [3]:
```
df=pd.read_csv('keywords_new.csv')
```

In [4]:
```
df.head()
```

Out[4]:

	_id	created_at	id	text	
0	5f84d7835785643da5d86bc8	Mon Oct 12 22:23:58 +0000 2020	1315780274890903560	RT @realDonaldTrump: “I’m running as a proud D...	href="http://twi
1	5f84d7835785643da5d86bca	Mon Oct 12 22:23:58 +0000 2020	1315780274752548866	RT @ProjectLincoln: You literally have covid a...	href="http://tw
2	5f84d7835785643da5d86bcc	Mon Oct 12 22:23:58 +0000 2020	1315780274848899072	RT @TheDemCoalition: "America has its own uniq...	href="http://tw
3	5f84d7835785643da5d86bce	Mon Oct 12 22:23:58 +0000 2020	1315780274677051399	@r_a_salvatore @JoeBiden Not the point. Biden ...	href="http://tw
4	5f84d7835785643da5d86bd0	Mon Oct 12 22:23:58 +0000 2020	1315780274731585536	RT @TeamTrump: Vice President @Mike_Pence: Whe...	href="http://tw

In [5]:
```
df=df[['id','created_at','text','user.location']]
df=df.rename(columns={"user.location":"location"})
```

In [6]:
```
import re
def clean_tweets(text):
  text = re.sub("RT @[\w]*:","",text)
  text = re.sub("@[\w]*","",text)
  text = re.sub("https?://[A-Za-z0-9./]*","",text)
  text = re.sub("\n","",text)
  return text
```

In [7]:
```
df['text'] = df['text'].apply(lambda x: clean_tweets(x))
```

In [8]:
```
df.head()
```

Out[8]:

	id	created_at	text	location
0	1315780274890903560	Mon Oct 12 22:23:58 +0000 2020	“I’m running as a proud Democrat, for the Sen...	Nigeria
1	1315780274752548866	Mon Oct 12 22:23:58 +0000 2020	You literally have covid and are going to Flo...	East Rutherford, NJ
2	1315780274848899072	Mon Oct 12 22:23:58 +0000 2020	"America has its own unique, dangerous form o...	Portland, Oregon
3	1315780274677051399	Mon Oct 12 22:23:58 +0000 2020	Not the point. Biden is telling voters they ...	Near the coffee ☕
4	1315780274731585536	Mon Oct 12 22:23:58 +0000 2020	Vice President : Where Joe Biden is talking a...	Tarheel State

In [9]:
```
df['subject']="x"
for i,k in enumerate(df['text']):
    if ("biden" or "joe") in k.lower():
        df['subject'][i]='Biden'
       # df['relevant'][i]=True
    if ("trump" or "donald" or 'president') in k.lower():
        df['subject'][i]='Trump'
        #df['relevant'][i]=True

    if ('dems' or "democrats" or "left" or"democratic") in k.lower():
        df['subject'][i]='Biden'
        #df['relevant'][i]=True

    if ("republicans" or "reps" or "right") in k.lower():
        df['subject'][i]='Trump'
        #df['relevant'][i]=True
```

```
C:\Users\SUMANA\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  after removing the cwd from sys.path.
C:\Users\SUMANA\Anaconda3\lib\site-packages\ipykernel_launcher.py:7: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
C:\Users\SUMANA\Anaconda3\lib\site-packages\ipykernel_launcher.py:15: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  from ipykernel import kernelapp as app
C:\Users\SUMANA\Anaconda3\lib\site-packages\ipykernel_launcher.py:11: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
```

In [10]:
```
df.head()
```

Out[10]:

	id	created_at	text	location	subject
0	1315780274890903560	Mon Oct 12 22:23:58 +0000 2020	“I’m running as a proud Democrat, for the Sen...	Nigeria	Biden
1	1315780274752548866	Mon Oct 12 22:23:58 +0000 2020	You literally have covid and are going to Flo...	East Rutherford, NJ	x
2	1315780274848899072	Mon Oct 12 22:23:58 +0000 2020	"America has its own unique, dangerous form o...	Portland, Oregon	x
3	1315780274677051399	Mon Oct 12 22:23:58 +0000 2020	Not the point. Biden is telling voters they ...	Near the coffee ☕	Biden
4	1315780274731585536	Mon Oct 12 22:23:58 +0000 2020	Vice President : Where Joe Biden is talking a...	Tarheel State	Biden

In [11]:
```
df.dtypes
df.isnull().sum()
```

Out[11]:
```
id              0
created_at      0
text            0
location     6625
subject         0
dtype: int64
```

In [12]:
```
df.size
```

Out[12]:
```
81500
```

In [13]:
```
df.drop_duplicates(inplace=True)
df.dropna(inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9675 entries, 0 to 16298
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   id          9675 non-null   int64 
 1   created_at  9675 non-null   object
 2   text        9675 non-null   object
 3   location    9675 non-null   object
 4   subject     9675 non-null   object
dtypes: int64(1), object(4)
memory usage: 453.5+ KB
```

In [14]:
```
df['time_created'] = pd.to_datetime(df.created_at)
df['time_created'] = df.time_created.map(lambda x: x.strftime('%Y-%m-%d'))
df[['created_at', 'time_created']].head()
```

Out[14]:

	created_at	time_created
0	Mon Oct 12 22:23:58 +0000 2020	2020-10-12
1	Mon Oct 12 22:23:58 +0000 2020	2020-10-12
2	Mon Oct 12 22:23:58 +0000 2020	2020-10-12
3	Mon Oct 12 22:23:58 +0000 2020	2020-10-12
4	Mon Oct 12 22:23:58 +0000 2020	2020-10-12

In [15]:
```
df=df[['id','text','location','subject','time_created']]
```

In [16]:
```
df.head()
```

Out[16]:

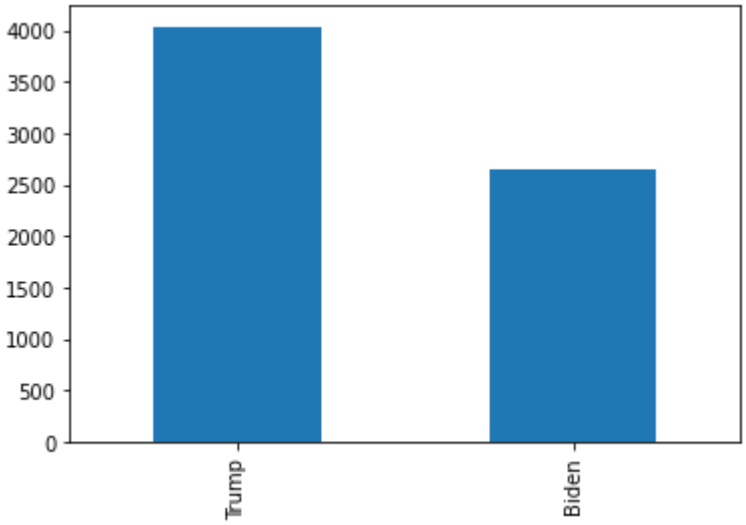
	id	text	location	subject	time_created
0	1315780274890903560	“I’m running as a proud Democrat, for the Sen...	Nigeria	Biden	2020-10-12
1	1315780274752548866	You literally have covid and are going to Flo...	East Rutherford, NJ	x	2020-10-12
2	1315780274848899072	"America has its own unique, dangerous form o...	Portland, Oregon	x	2020-10-12
3	1315780274677051399	Not the point. Biden is telling voters they ...	Near the coffee	Biden	2020-10-12
4	1315780274731585536	Vice President : Where Joe Biden is talking a...	Tarheel State	Biden	2020-10-12

In [17]:
```
df.drop(df.index[df['subject'] == 'x'], inplace = True)
```

In [18]:
```
print(df.subject.value_counts())
df.subject.value_counts().plot(kind='bar')
```

```
Trump    4035
Biden    2648
Name: subject, dtype: int64
```

Out[18]: <AxesSubplot:>



In [19]:
```
df['count']=1
grouped = df.groupby(['time_created', 'subject'])
grouped = grouped['count'].sum().reset_index()
grouped.tail()
```

Out[19]:

	time_created	subject	count
0	2020-10-12	Biden	2648
1	2020-10-12	Trump	4035

In [20]:
```
states = ['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California', 'Colorado'
, 'Connecticut', 'Delaware', 'Florida', 'Georgia', 'Hawaii',
          'Idaho', 'Illinois', 'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Louis
iana', 'Maine', 'Maryland', 'Massachusetts', 'Michigan', 'Minnesota',
          'Mississippi', 'Missouri', 'Montana', 'Nebraska', 'Nevada', 'New Ham
pshire', 'New Jersey', 'New York', 'New Mexico', 'North Carolina',
          'North Dakota', 'Ohio', 'Oklahoma', 'Oregon', 'Pennsylvania', 'Rhode
Island', 'South Carolina', 'South Dakota', 'Tennessee', 'Texas', 'Utah',
          'Vermont', 'Virginia', 'Washington', 'West Virginia', 'Wisconsin',
'Wyoming']
stateCodes = ['AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'FL', 'GA', 'HI'
, 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', 'MN',
              'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND'
, 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA',
              'WA', 'WV', 'WI', 'WY']
stateMapping = {'AL': 'Alabama', 'AK': 'Alaska', 'AZ': 'Arizona', 'AR': 'Arkan
sas', 'CA': 'California',
                'CO': 'Colorado', 'CT': 'Connecticut', 'DE': 'Delaware', 'F
L': 'Florida', 'GA': 'Georgia',
                'HI': 'Hawaii', 'ID': 'Idaho', 'IL': 'Illinois', 'IN': 'Indi
ana', 'IA': 'Iowa', 'KS': 'Kansas',
                'KY': 'Kentucky', 'LA': 'Louisiana', 'ME': 'Maine', 'MD': 'M
aryland', 'MA': 'Massachusetts',
                'MI': 'Michigan', 'MN': 'Minnesota', 'MS': 'Mississippi', 'M
O': 'Missouri', 'MT': 'Montana',
                'NE': 'Nebraska', 'NV': 'Nevada', 'NH': 'New Hampshire', 'N
J': 'New Jersey', 'NY': 'New York',
                'NM': 'New Mexico', 'NC': 'North Carolina', 'ND': 'North Dak
ota', 'OH': 'Ohio', 'OK': 'Oklahoma',
                'OR': 'Oregon', 'PA': 'Pennsylvania', 'RI': 'Rhode Island',
'SC': 'South Carolina',
                'SD': 'South Dakota', 'TN': 'Tennessee', 'TX': 'Texas', 'UT'
:  'Utah', 'VT': 'Vermont',
                'VA': 'Virginia', 'WA': 'Washington', 'WV':  'West Virginia'
, 'WI': 'Wisconsin', 'WY': 'Wyoming'}
```

In [21]:
```
df = df[df['location'].notnull()]
```

In [22]:
```
df.head()
```

Out[22]:

	id	text	location	subject	time_created	count
0	1315780274890903560	“I’m running as a proud Democrat, for the Sen...	Nigeria	Biden	2020-10-12	1
3	1315780274677051399	Not the point. Biden is telling voters they ...	Near the coffee ☕	Biden	2020-10-12	1
4	1315780274731585536	Vice President : Where Joe Biden is talking a...	Tarheel State	Biden	2020-10-12	1
5	1315780275004149761	Breaking: President Trump has tested negative...	Oklahoma	Trump	2020-10-12	1
8	1315780274995724288	. has tested positive... for Trump Derangemen...	Monaca PA	Trump	2020-10-12	1

In [23]:
```
tweet_copied_df = df
for index, row in df.iterrows():
  flag = 0
  if row.location:
    locationSplit = row.location.split(',')
    for word in locationSplit:
      word_stripped = word.strip()
      if word_stripped in states:
        flag = 1
        row['state'] = word_stripped
      elif word_stripped in stateCodes:
        flag = 1
        row['state'] = stateMapping[word_stripped]
  if flag == 0:
    tweet_copied_df = tweet_copied_df.drop(index=index)
  else:
    tweet_copied_df.loc[index, 'state'] = row['state']
```

In [24]:
```
tweet_copied_df=tweet_copied_df[['id','text','location','subject','time_create
d','state']]
```

In [25]:
```
tweet_copied_df
```

Out[25]:

	id	text	location	subject	time_created	state
5	1315780275004149761	Breaking: President Trump has tested negative...	Oklahoma	Trump	2020-10-12	Oklahoma
9	1315780274911801344	“I’m running as a proud Democrat, for the Sen...	Wyoming, USA	Biden	2020-10-12	Wyoming
21	1315780275343749121	Gov DeSantis just entered Trump’s Sanford hat...	Sammamish, WA	Trump	2020-10-12	Washington
22	1315780275260076034	Hello media, the broke a story that Donald T...	New York, USA	Trump	2020-10-12	New York
24	1315780275490549760	President Trump now totally over Coronavirus....	Minnesota	Trump	2020-10-12	Minnesota
...	...	...	...	...	...	...
16280	1315781636630499328	Absolutely Austin. Trump was always going to ...	Brooklyn, NY	Trump	2020-10-12	New York
16282	1315781636735275008	Tony Fauci will do an interview on CNN today,...	Ocean Bluff, MA	Biden	2020-10-12	Massachusetts
16286	1315781636890320897	That’s “President Trump” sir...	Iowa	Trump	2020-10-12	Iowa
16292	1315781637137989634	should be ashamed if they allow this. It’s...	Virginia	Trump	2020-10-12	Virginia
16293	1315781637158907905	So true I guess the Biden supporters missed...	Ohio, USA	Trump	2020-10-12	Ohio

3321 rows × 6 columns

In [ ]:
```
#SENTIMENT INTENSITY ANALYZER
```

In [26]:
```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\SUMANA\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

Out[26]: True

In [27]:
```
sid = SentimentIntensityAnalyzer()
```

In [28]:
```
tweets_election_df=tweet_copied_df
```

In [29]:
```
tweets_election_df.head()
```

Out[29]:

	id	text	location	subject	time_created	state
5	1315780275004149761	Breaking: President Trump has tested negative...	Oklahoma	Trump	2020-10-12	Oklahoma
9	1315780274911801344	“I’m running as a proud Democrat, for the Sen...	Wyoming, USA	Biden	2020-10-12	Wyoming
21	1315780275343749121	Gov DeSantis just entered Trump’s Sanford hat...	Sammamish, WA	Trump	2020-10-12	Washington
22	1315780275260076034	Hello media, the broke a story that Donald T...	New York, USA	Trump	2020-10-12	New York
24	1315780275490549760	President Trump now totally over Coronavirus....	Minnesota	Trump	2020-10-12	Minnesota

In [30]:
```
tweets_trump = tweets_election_df[tweets_election_df.subject == 'Trump']
tweets_trump.head()
```

Out[30]:

	id	text	location	subject	time_created	state
5	1315780275004149761	Breaking: President Trump has tested negative...	Oklahoma	Trump	2020-10-12	Oklahoma
21	1315780275343749121	Gov DeSantis just entered Trump’s Sanford hat...	Sammamish, WA	Trump	2020-10-12	Washington
22	1315780275260076034	Hello media, the broke a story that Donald T...	New York, USA	Trump	2020-10-12	New York
24	1315780275490549760	President Trump now totally over Coronavirus....	Minnesota	Trump	2020-10-12	Minnesota
38	1315780275817918465	Umpteenth reminder that in the closing days o...	Brooklyn, NY	Trump	2020-10-12	New York

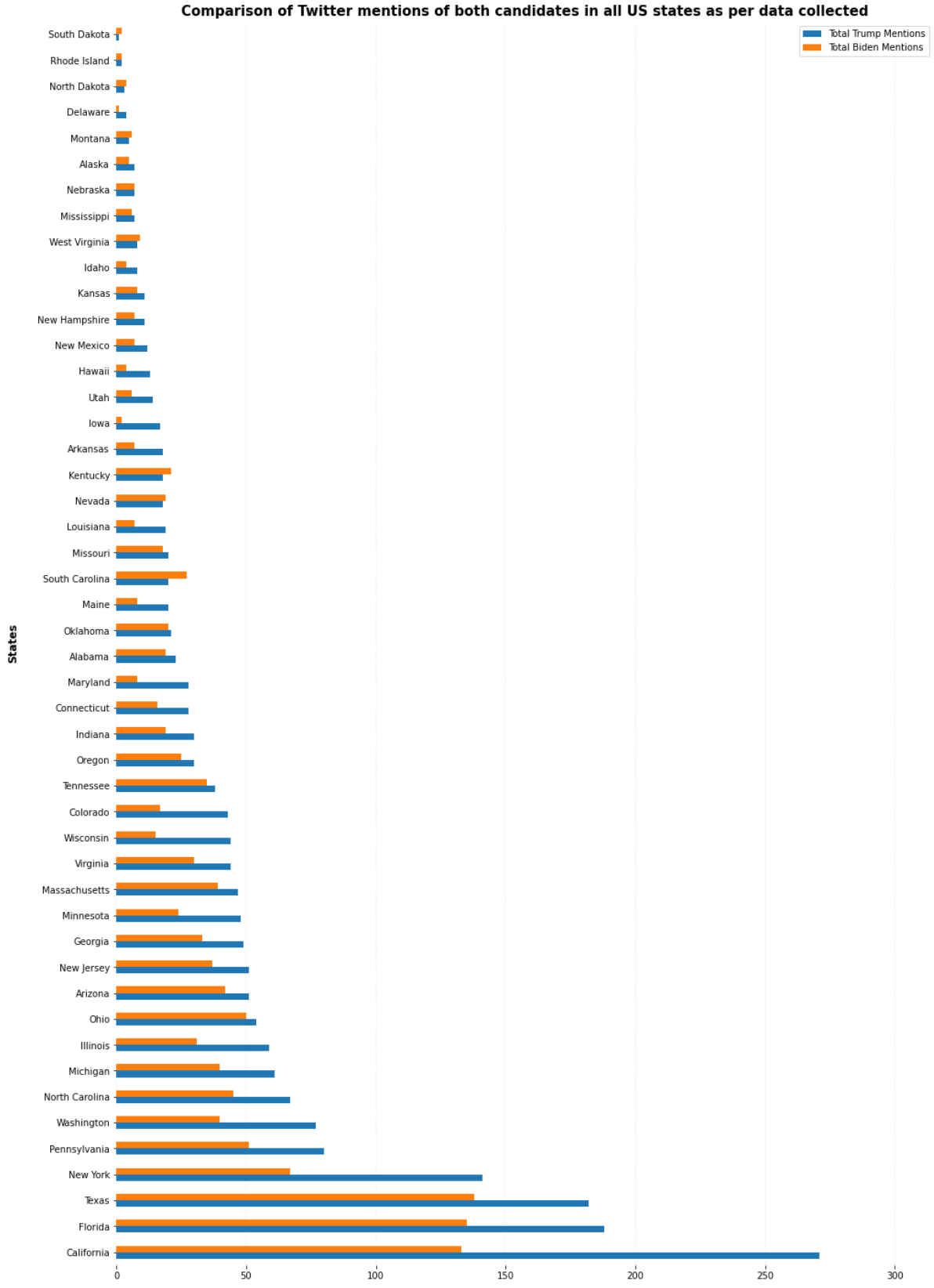
In [31]:
```
tweets_biden = tweets_election_df[tweets_election_df.subject == 'Biden']
tweets_biden.head()
```

Out[31]:

	id	text	location	subject	time_created	state
9	1315780274911801344	“I’m running as a proud Democrat, for the Sen...	Wyoming, USA	Biden	2020-10-12	Wyoming
34	1315780275742404614	Biden says he was raised in a Black Baptist C...	Georgia, USA	Biden	2020-10-12	Georgia
48	1315780275876421633	Joe Biden: "I'm running as a proud Democrat f...	Los Angeles, CA	Biden	2020-10-12	California
72	1315780277168410624	📺 NEW VIDEORetweet if you are ready to vote B...	Fairfax, VA	Biden	2020-10-12	Virginia
87	1315780277617254401	Just to be clear Biden and Harris would be fi...	Brookfield, MO, USA	Biden	2020-10-12	Missouri

In [32]:
```
df1 = pd.merge(tweets_trump['state'].value_counts(), tweets_biden['state'].val
ue_counts(), right_index = True,
               left_index = True)
df1 = df1.rename(columns = {"state_x": "Total Trump Mentions", "state_y": "Tot
al Biden Mentions"})
ax = df1.plot(kind='barh', figsize=(16, 25), zorder=2)

# Despine
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_visible(False)

#Replacing ticks with horizontal lines
#ax.tick_params(axis="both", which="both", bottom="off", top="off", labelbotto
m="on", left="off", right="off", labelleft="on")
vals = ax.get_xticks()
for tick in vals:
      ax.axvline(x=tick, linestyle='dashed', alpha=0.4, color='#eeeeee', zorde
r=1)

# Set y-axis label
ax.set_ylabel("States", labelpad=20, weight='bold', size=12)
ax.set_title('Comparison of Twitter mentions of both candidates in all US stat
es as per data collected',fontweight="bold", size=15)
```

Out[32]: Text(0.5, 1.0, 'Comparison of Twitter mentions of both candidates in all US states as per data collected')



In [33]:
```
df1.head()
```

Out[33]:

	Total Trump Mentions	Total Biden Mentions
California	271	133
Florida	188	135
Texas	182	138
New York	141	67
Pennsylvania	80	51

In [34]:
```
tweets_trump['sentiment'] = tweets_trump['text'].apply(lambda x: sid.polarity_
scores(x))
tweets_biden['sentiment'] = tweets_biden['text'].apply(lambda x: sid.polarity_
scores(x))
```

```
C:\Users\SUMANA\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
C:\Users\SUMANA\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
```

In [35]:
```
def sentimentVerdict(sentiment):
  if sentiment['compound'] >= 0.05:
    return "Positive"
  elif sentiment['compound'] <= -0.05:
    return "Negative"
  else:
    return "Neutral"
```

In [36]:
```
tweets_trump['sentiment_overall'] = tweets_trump['sentiment'].apply(lambda x:
sentimentVerdict(x))
tweets_trump
```

```
C:\Users\SUMANA\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
```

Out[36]:

	id	text	location	subject	time_created	state	se
5	1315780275004149761	Breaking: President Trump has tested negative...	Oklahoma	Trump	2020-10-12	Oklahoma	{'neg 'neu 'p
21	1315780275343749121	Gov DeSantis just entered Trump's Sanford hat...	Sammamish, WA	Trump	2020-10-12	Washington	{'neg 'neu 'p
22	1315780275260076034	Hello media, the broke a story that Donald T...	New York, USA	Trump	2020-10-12	New York	{'neg 'neu 'p
24	1315780275490549760	President Trump now totally over Coronavirus....	Minnesota	Trump	2020-10-12	Minnesota	{'neg 'neu 'p
38	1315780275817918465	Umpteenth reminder that in the closing days o...	Brooklyn, NY	Trump	2020-10-12	New York	{'neg 'neu 'pos
...	...	...	...	...	...	...	
16274	1315781636533952512	I just hit the live feed on of Trump's rally...	Washington, D.C.	Trump	2020-10-12	Washington	{'neg 'neu 'p
16280	1315781636630499328	Absolutely Austin. Trump was always going to ...	Brooklyn, NY	Trump	2020-10-12	New York	{'neg 'neu 'p
16286	1315781636890320897	That's "President Trump" sir...	Iowa	Trump	2020-10-12	Iowa	{'n 'n 'p 'com
16292	1315781637137989634	should be ashamed if they allow this. It's...	Virginia	Trump	2020-10-12	Virginia	{'neg 'neu 'pos
16293	1315781637158907905	So true I guess the Biden supporters missed...	Ohio, USA	Trump	2020-10-12	Ohio	{'neg 'neu 'pos

2018 rows × 8 columns

In [37]:
```
tweets_biden['sentiment_overall'] = tweets_biden['sentiment'].apply(lambda x:
sentimentVerdict(x))
tweets_biden
```

```
C:\Users\SUMANA\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
```

Out[37]:

	id	text	location	subject	time_created	state	
9	1315780274911801344	“I’m running as a proud Democrat, for the Sen...	Wyoming, USA	Biden	2020-10-12	Wyoming	{
34	1315780275742404614	Biden says he was raised in a Black Baptist C...	Georgia, USA	Biden	2020-10-12	Georgia	'
48	1315780275876421633	Joe Biden: "I'm running as a proud Democrat f...	Los Angeles, CA	Biden	2020-10-12	California	'
72	1315780277168410624	📺 NEW VIDEORetweet if you are ready to vote B...	Fairfax, VA	Biden	2020-10-12	Virginia	{'
87	1315780277617254401	Just to be clear Biden and Harris would be fi...	Brookfield, MO, USA	Biden	2020-10-12	Missouri	{'
...	...	...	...	...	...	...	
16255	1315781635921584128	“I’m running as a proud Democrat, for the Sen...	Florida, USA	Biden	2020-10-12	Florida	{
16258	1315781635845963777	There will be No Biden/Harris administration	Washington, USA	Biden	2020-10-12	Washington	{'
16260	1315781635942608903	Well I can only speak for myself, but I’m goi...	Virginia, USA	Biden	2020-10-12	Virginia	'
16275	1315781636559052800	In Cincinnati, Biden appears to be back into ...	Orofino, ID	Biden	2020-10-12	Idaho	'
16282	1315781636735275008	Tony Fauci will do an interview on CNN today,...	Ocean Bluff, MA	Biden	2020-10-12	Massachusetts	'

1303 rows × 8 columns

In [38]:
```
df2=pd.concat([tweets_biden, tweets_trump], ignore_index=True, sort=False)
```

In [39]:
```
df2
```

Out[39]:

	id	text	location	subject	time_created	state	sen
0	1315780274911801344	“I’m running as a proud Democrat, for the Sen...	Wyoming, USA	Biden	2020-10-12	Wyoming	{'neg' 'neu': 'pos'
1	1315780275742404614	Biden says he was raised in a Black Baptist C...	Georgia, USA	Biden	2020-10-12	Georgia	{'ne 'ne 'pc 'comp
2	1315780275876421633	Joe Biden: "I'm running as a proud Democrat f...	Los Angeles, CA	Biden	2020-10-12	California	{'ne 'neu': 'pos': '
3	1315780277168410624	📺 NEW VIDEORetweet if you are ready to vote B...	Fairfax, VA	Biden	2020-10-12	Virginia	{'neg': 'neu': 'pos':
4	1315780277617254401	Just to be clear Biden and Harris would be fi...	Brookfield, MO, USA	Biden	2020-10-12	Missouri	{'neg': 'neu': 'pos':
...	...	...	...	...	...	...	
3316	1315781636533952512	I just hit the live feed on of Trump’s rally...	Washington, D.C.	Trump	2020-10-12	Washington	{'neg': 'neu': 'pc '
3317	1315781636630499328	Absolutely Austin. Trump was always going to ...	Brooklyn, NY	Trump	2020-10-12	New York	{'neg': 'neu': 'pc '
3318	1315781636890320897	That’s “President Trump” sir...	Iowa	Trump	2020-10-12	Iowa	{'ne 'ne 'pc 'comp
3319	1315781637137989634	should be ashamed if they allow this. It’s...	Virginia	Trump	2020-10-12	Virginia	{'neg': 'neu': 'pos':
3320	1315781637158907905	So true I guess the Biden supporters missed...	Ohio, USA	Trump	2020-10-12	Ohio	{'neg': 'neu': 'pos':

3321 rows × 8 columns

In [40]:
```
df3=df2[['subject','state','sentiment_overall']]
```

In [41]:
```
df3
```

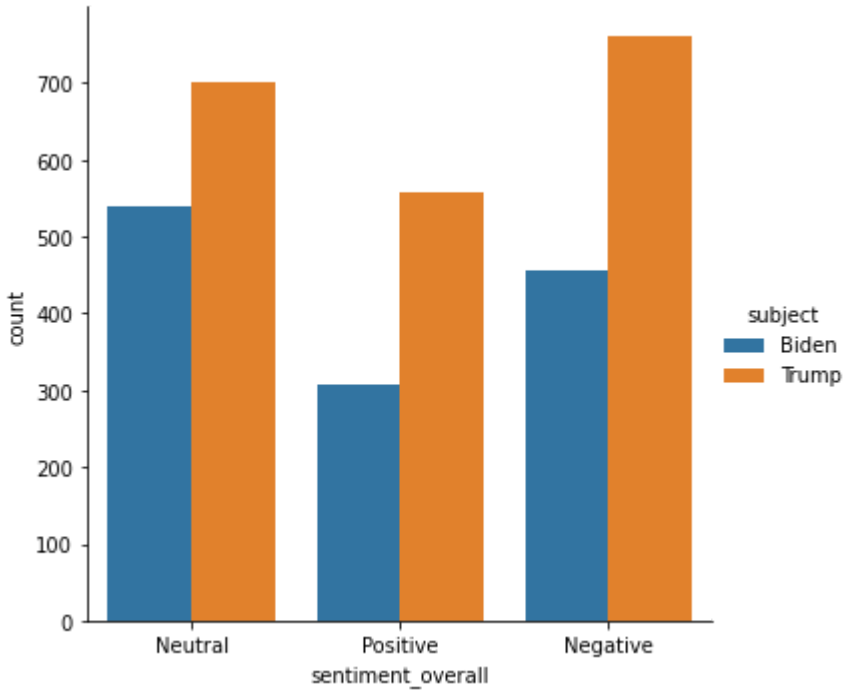
Out[41]:

	subject	state	sentiment_overall
0	Biden	Wyoming	Neutral
1	Biden	Georgia	Neutral
2	Biden	California	Positive
3	Biden	Virginia	Negative
4	Biden	Missouri	Positive
...	...	...	...
3316	Trump	Washington	Negative
3317	Trump	New York	Negative
3318	Trump	Iowa	Neutral
3319	Trump	Virginia	Negative
3320	Trump	Ohio	Positive

3321 rows × 3 columns

In [42]:
```
sns.catplot(data=df3,kind='count',x='sentiment_overall',hue='subject')
```

Out[42]:
```
<seaborn.axisgrid.FacetGrid at 0x24670306888>
```



In [44]:
```
tweets_trump_location = tweets_trump.groupby(['state', 'sentiment_overall']).c
ount()
tweets_trump_location = tweets_trump_location['id']
with pd.option_context('display.max_rows', None, 'display.max_columns', None):
    print(tweets_trump_location)
```

state	sentiment_overall	
Alabama	Negative	5
	Neutral	11
	Positive	7
Alaska	Neutral	5
	Positive	2
Arizona	Negative	19
	Neutral	19
	Positive	13
Arkansas	Negative	5
	Neutral	5
	Positive	8
California	Negative	97
	Neutral	97
	Positive	77
Colorado	Negative	15
	Neutral	20
	Positive	8
Connecticut	Negative	14
	Neutral	9
	Positive	5
Delaware	Negative	3
	Neutral	1
Florida	Negative	64
	Neutral	72
	Positive	52
Georgia	Negative	15
	Neutral	19
	Positive	15
Hawaii	Negative	5
	Neutral	5
	Positive	3
Idaho	Negative	2
	Neutral	3
	Positive	3
Illinois	Negative	28
	Neutral	17
	Positive	14
Indiana	Negative	12
	Neutral	6
	Positive	12
Iowa	Negative	7
	Neutral	3
	Positive	7
Kansas	Negative	7
	Neutral	3
	Positive	1
Kentucky	Negative	10
	Neutral	3
	Positive	5
Louisiana	Negative	9
	Neutral	7
	Positive	3
Maine	Negative	9
	Neutral	5
	Positive	6
Maryland	Negative	11

	Neutral	7
	Positive	10
Massachusetts	Negative	15
	Neutral	20
	Positive	12
Michigan	Negative	26
	Neutral	25
	Positive	10
Minnesota	Negative	31
	Neutral	9
	Positive	8
Mississippi	Negative	5
	Positive	2
Missouri	Negative	10
	Neutral	6
	Positive	4
Montana	Negative	2
	Neutral	1
	Positive	2
Nebraska	Negative	3
	Neutral	1
	Positive	3
Nevada	Negative	7
	Neutral	5
	Positive	6
New Hampshire	Negative	4
	Neutral	6
	Positive	1
New Jersey	Negative	21
	Neutral	13
	Positive	17
New Mexico	Negative	4
	Neutral	4
	Positive	4
New York	Negative	47
	Neutral	59
	Positive	35
North Carolina	Negative	28
	Neutral	25
	Positive	14
North Dakota	Negative	1
	Positive	2
Ohio	Negative	10
	Neutral	20
	Positive	24
Oklahoma	Negative	7
	Neutral	7
	Positive	7
Oregon	Negative	15
	Neutral	8
	Positive	7
Pennsylvania	Negative	29
	Neutral	27
	Positive	24
Rhode Island	Negative	1
	Positive	1
South Carolina	Negative	5

```
                Neutral              7
                Positive             8
South Dakota    Negative             1
Tennessee       Negative            13
                Neutral             13
                Positive            12
Texas           Negative            71
                Neutral             57
                Positive            54
Utah            Negative             4
                Neutral              7
                Positive             3
Virginia        Negative            14
                Neutral             18
                Positive            12
Washington      Negative            26
                Neutral             31
                Positive            20
West Virginia   Negative             6
                Neutral              2
Wisconsin       Negative            18
                Neutral             12
                Positive            14
Name: id, dtype: int64
```

In [45]:
```
tweets_trump_location
```

Out[45]:
```
state          sentiment_overall
Alabama        Negative             5
               Neutral             11
               Positive             7
Alaska         Neutral              5
               Positive             2
                                   ..
West Virginia  Negative             6
               Neutral              2
Wisconsin      Negative            18
               Neutral             12
               Positive            14
Name: id, Length: 136, dtype: int64
```

In [46]:
```
tweets_biden_location = tweets_biden.groupby(['state', 'sentiment_overall']).c
ount()
tweets_biden_location = tweets_biden_location['id']
with pd.option_context('display.max_rows', None, 'display.max_columns', None):
    print(tweets_biden_location)
```

state	sentiment_overall	
Alabama	Negative	5
	Neutral	6
	Positive	8
Alaska	Negative	1
	Neutral	2
	Positive	2
Arizona	Negative	11
	Neutral	23
	Positive	8
Arkansas	Negative	2
	Neutral	4
	Positive	1
California	Negative	48
	Neutral	57
	Positive	28
Colorado	Negative	7
	Neutral	6
	Positive	4
Connecticut	Negative	2
	Neutral	8
	Positive	6
Delaware	Neutral	1
Florida	Negative	47
	Neutral	52
	Positive	36
Georgia	Negative	10
	Neutral	11
	Positive	12
Hawaii	Negative	1
	Neutral	3
Idaho	Negative	2
	Positive	2
Illinois	Negative	13
	Neutral	12
	Positive	6
Indiana	Negative	6
	Neutral	6
	Positive	7
Iowa	Negative	1
	Positive	1
Kansas	Negative	4
	Neutral	3
	Positive	1
Kentucky	Negative	9
	Neutral	8
	Positive	4
Louisiana	Negative	3
	Neutral	3
	Positive	1
Maine	Negative	3
	Neutral	3
	Positive	2
Maryland	Negative	3
	Neutral	2
	Positive	3
Massachusetts	Negative	8

	Neutral	25
	Positive	6
Michigan	Negative	18
	Neutral	18
	Positive	4
Minnesota	Negative	9
	Neutral	12
	Positive	3
Mississippi	Negative	1
	Neutral	5
Missouri	Negative	9
	Neutral	5
	Positive	4
Montana	Negative	2
	Neutral	2
	Positive	2
Nebraska	Negative	3
	Neutral	3
	Positive	1
Nevada	Negative	8
	Neutral	7
	Positive	4
New Hampshire	Negative	2
	Neutral	4
	Positive	1
New Jersey	Negative	8
	Neutral	20
	Positive	9
New Mexico	Neutral	5
	Positive	2
New York	Negative	20
	Neutral	27
	Positive	20
North Carolina	Negative	16
	Neutral	17
	Positive	12
North Dakota	Negative	2
	Neutral	1
	Positive	1
Ohio	Negative	15
	Neutral	18
	Positive	17
Oklahoma	Negative	10
	Neutral	6
	Positive	4
Oregon	Negative	12
	Neutral	8
	Positive	5
Pennsylvania	Negative	19
	Neutral	18
	Positive	14
Rhode Island	Negative	1
	Positive	1
South Carolina	Negative	12
	Neutral	9
	Positive	6
South Dakota	Negative	1

```
                Positive              1
Tennessee       Negative             17
                Neutral              15
                Positive              3
Texas           Negative             51
                Neutral              56
                Positive             31
Utah            Negative              2
                Neutral               3
                Positive              1
Vermont         Negative              2
Virginia        Negative              9
                Neutral              12
                Positive              9
Washington      Negative             12
                Neutral              16
                Positive             12
West Virginia   Negative              2
                Neutral               6
                Positive              1
Wisconsin       Negative              6
                Neutral               7
                Positive              2
Wyoming         Negative              1
                Neutral               4
Name: id, dtype: int64
```

In [47]:
```
tweets_location_df = pd.DataFrame({'State': [state for state in states],
                                   'Trump Positive': [0 for state in sta
tes],
                                   'Trump Negative': [0 for state in sta
tes],
                                   'Trump Neutral': [0 for state in stat
es],
                                   'Trump Total': [0 for state in states
],
                                   'Biden Positive': [0 for state in sta
tes],
                                   'Biden Negative': [0 for state in sta
tes],
                                   'Biden Neutral': [0 for state in stat
es],
                                   'Biden Total': [0 for state in states
]})
tweets_location_df.set_index('State', inplace = True)
for state in states:
  positiveTrump, negativeTrump, neutralTrump, positiveBiden, negativeBiden, ne
utralBiden = 0, 0, 0, 0, 0, 0
  try:
    positiveTrump = tweets_trump_location[state]['Positive']
  except:
    positiveTrump = 0

  try:
    negativeTrump = tweets_trump_location[state]['Negative']
  except:
    negativeTrump = 0

  try:
    neutralTrump = tweets_trump_location[state]['Neutral']
  except:
    neutralTrump = 0

  try:
    positiveBiden = tweets_biden_location[state]['Positive']
  except:
    positiveBiden = 0

  try:
    negativeBiden = tweets_biden_location[state]['Negative']
  except:
    negativeBiden = 0

  try:
    neutralBiden = tweets_biden_location[state]['Neutral']
  except:
    neutralBiden = 0

  totalTrump = positiveTrump + negativeTrump + neutralTrump
  totalBiden = positiveBiden + negativeBiden + neutralBiden

  if totalTrump == 0:
    tweets_location_df.at[state, 'Trump Positive'], tweets_location_df.at[stat
```

```
e, 'Trump Negative'], tweets_location_df.at[state, 'Trump Neutral'] = 0,0,0
  else:
    tweets_location_df.at[state, 'Trump Positive'] = round((positiveTrump/tota
lTrump)*100.0)
    tweets_location_df.at[state, 'Trump Negative'] = round((negativeTrump/tota
lTrump)*100.0)
    tweets_location_df.at[state, 'Trump Neutral'] = round((neutralTrump/totalT
rump)*100.0)
  tweets_location_df.at[state, 'Trump Total'] = totalTrump

  if totalBiden == 0:
    tweets_location_df.at[state, 'Biden Positive'], tweets_location_df.at[stat
e, 'Biden Negative'], tweets_location_df.at[state, 'Biden Neutral'] = 0,0,0
  else:
    tweets_location_df.at[state, 'Biden Positive'] = round((positiveBiden/tota
lBiden)*100.0)
    tweets_location_df.at[state, 'Biden Negative'] = round((negativeBiden/tota
lBiden)*100.0)
    tweets_location_df.at[state, 'Biden Neutral'] = round((neutralBiden/totalB
iden)*100.0)
  tweets_location_df.at[state, 'Biden Total'] = totalBiden
```

In [48]:
```
tweets_location_df
```

Out[48]:

State	Trump Positive	Trump Negative	Trump Neutral	Trump Total	Biden Positive	Biden Negative	Biden Neutral	Biden Total
Alabama	30	22	48	23	42	26	32	19
Alaska	29	0	71	7	40	20	40	5
Arizona	25	37	37	51	19	26	55	42
Arkansas	44	28	28	18	14	29	57	7
California	28	36	36	271	21	36	43	133
Colorado	19	35	47	43	24	41	35	17
Connecticut	18	50	32	28	38	12	50	16
Delaware	0	75	25	4	0	0	100	1
Florida	28	34	38	188	27	35	39	135
Georgia	31	31	39	49	36	30	33	33
Hawaii	23	38	38	13	0	25	75	4
Idaho	38	25	38	8	50	50	0	4
Illinois	24	47	29	59	19	42	39	31
Indiana	40	40	20	30	37	32	32	19
Iowa	41	41	18	17	50	50	0	2
Kansas	9	64	27	11	12	50	38	8
Kentucky	28	56	17	18	19	43	38	21
Louisiana	16	47	37	19	14	43	43	7
Maine	30	45	25	20	25	38	38	8
Maryland	36	39	25	28	38	38	25	8
Massachusetts	26	32	43	47	15	21	64	39
Michigan	16	43	41	61	10	45	45	40
Minnesota	17	65	19	48	12	38	50	24
Mississippi	29	71	0	7	0	17	83	6
Missouri	20	50	30	20	22	50	28	18
Montana	40	40	20	5	33	33	33	6
Nebraska	43	43	14	7	14	43	43	7
Nevada	33	39	28	18	21	42	37	19
New Hampshire	9	36	55	11	14	29	57	7
New Jersey	33	41	25	51	24	22	54	37
New York	25	33	42	141	30	30	40	67
New Mexico	33	33	33	12	29	0	71	7
North Carolina	21	42	37	67	27	36	38	45

State	Trump Positive	Trump Negative	Trump Neutral	Trump Total	Biden Positive	Biden Negative	Biden Neutral	Biden Total
North Dakota	67	33	0	3	25	50	25	4
Ohio	44	19	37	54	34	30	36	50
Oklahoma	33	33	33	21	20	50	30	20
Oregon	23	50	27	30	20	48	32	25
Pennsylvania	30	36	34	80	27	37	35	51
Rhode Island	50	50	0	2	50	50	0	2
South Carolina	40	25	35	20	22	44	33	27
South Dakota	0	100	0	1	50	50	0	2
Tennessee	32	34	34	38	9	49	43	35
Texas	30	39	31	182	22	37	41	138
Utah	21	29	50	14	17	33	50	6
Vermont	0	0	0	0	0	100	0	2
Virginia	27	32	41	44	30	30	40	30
Washington	26	34	40	77	30	30	40	40
West Virginia	0	75	25	8	11	22	67	9
Wisconsin	32	41	27	44	13	40	47	15
Wyoming	0	0	0	0	0	20	80	5

In [49]:
```
tweets_location_df[['Trump Positive','Trump Negative','Trump Neutral','Biden P
ositive','Biden Negative','Biden Neutral']]
```

Out[49]:

State	Trump Positive	Trump Negative	Trump Neutral	Biden Positive	Biden Negative	Biden Neutral
Alabama	30	22	48	42	26	32
Alaska	29	0	71	40	20	40
Arizona	25	37	37	19	26	55
Arkansas	44	28	28	14	29	57
California	28	36	36	21	36	43
Colorado	19	35	47	24	41	35
Connecticut	18	50	32	38	12	50
Delaware	0	75	25	0	0	100
Florida	28	34	38	27	35	39
Georgia	31	31	39	36	30	33
Hawaii	23	38	38	0	25	75
Idaho	38	25	38	50	50	0
Illinois	24	47	29	19	42	39
Indiana	40	40	20	37	32	32
Iowa	41	41	18	50	50	0
Kansas	9	64	27	12	50	38
Kentucky	28	56	17	19	43	38
Louisiana	16	47	37	14	43	43
Maine	30	45	25	25	38	38
Maryland	36	39	25	38	38	25
Massachusetts	26	32	43	15	21	64
Michigan	16	43	41	10	45	45
Minnesota	17	65	19	12	38	50
Mississippi	29	71	0	0	17	83
Missouri	20	50	30	22	50	28
Montana	40	40	20	33	33	33
Nebraska	43	43	14	14	43	43
Nevada	33	39	28	21	42	37
New Hampshire	9	36	55	14	29	57
New Jersey	33	41	25	24	22	54
New York	25	33	42	30	30	40
New Mexico	33	33	33	29	0	71
North Carolina	21	42	37	27	36	38

State	Trump Positive	Trump Negative	Trump Neutral	Biden Positive	Biden Negative	Biden Neutral
North Dakota	67	33	0	25	50	25
Ohio	44	19	37	34	30	36
Oklahoma	33	33	33	20	50	30
Oregon	23	50	27	20	48	32
Pennsylvania	30	36	34	27	37	35
Rhode Island	50	50	0	50	50	0
South Carolina	40	25	35	22	44	33
South Dakota	0	100	0	50	50	0
Tennessee	32	34	34	9	49	43
Texas	30	39	31	22	37	41
Utah	21	29	50	17	33	50
Vermont	0	0	0	0	100	0
Virginia	27	32	41	30	30	40
Washington	26	34	40	30	30	40
West Virginia	0	75	25	11	22	67
Wisconsin	32	41	27	13	40	47
Wyoming	0	0	0	0	20	80

In [50]:
```
tweets_location_df.columns
```

Out[50]:
```
Index(['Trump Positive', 'Trump Negative', 'Trump Neutral', 'Trump Total',
       'Biden Positive', 'Biden Negative', 'Biden Neutral', 'Biden Total'],
      dtype='object')
```

In [51]:
```
tweets_location_df.reset_index(inplace=True)
```

In [52]:
```
tweets_location_df.head()
```

Out[52]:

	State	Trump Positive	Trump Negative	Trump Neutral	Trump Total	Biden Positive	Biden Negative	Biden Neutral	Biden Total
0	Alabama	30	22	48	23	42	26	32	19
1	Alaska	29	0	71	7	40	20	40	5
2	Arizona	25	37	37	51	19	26	55	42
3	Arkansas	44	28	28	18	14	29	57	7
4	California	28	36	36	271	21	36	43	133

In [53]:
```
tweets_location_df['Predicted Judgement'] = 'Neutral'
for index, row in tweets_location_df.iterrows():
  if row['Trump Total'] <= 15 and row['Biden Total'] <= 15:
    tweets_location_df.loc[index, 'Predicted Judgement'] = 'Insufficient Data'
  else:
    if row['Trump Positive'] > row['Biden Positive'] and (row['Trump Negative'
] < row['Biden Negative'] or row['Trump Neutral'] > row['Biden Neutral']):
      tweets_location_df.loc[index, 'Predicted Judgement'] = 'Strongly Republi
can'
    elif row['Biden Positive'] > row['Trump Positive'] and (row['Biden Negativ
e'] < row['Trump Negative'] or row['Biden Neutral'] > row['Trump Neutral']):
      tweets_location_df.loc[index, 'Predicted Judgement'] = 'Strongly Democra
tic'
    elif row['Trump Positive'] - row['Biden Positive'] > row['Biden Negative']
- row['Trump Negative']:
      tweets_location_df.loc[index, 'Predicted Judgement'] = 'Somewhat Republi
can'
    elif row['Biden Positive'] - row['Trump Positive'] > row['Trump Negative']
- row['Biden Negative']:
      tweets_location_df.loc[index, 'Predicted Judgement'] = 'Somewhat Democra
tic'
tweets_location_df = tweets_location_df.rename(columns={'Trump Positive': 'Tru
mp Positive (in %)',
                                                        'Trump Negative': 'Tru
mp Negative (in %)',
                                                        'Trump Neutral': 'Trum
p Neutral (in %)',
                                                        'Trump Total': 'Trump
 Total Mentions',
                                                        'Biden Positive': 'Bid
en Positive (in %)',
                                                        'Biden Negative': 'Bid
en Negative (in %)',
                                                        'Biden Neutral': 'Bide
n Neutral (in %)',
                                                        'Biden Total': 'Biden
 Total Mentions'})
```

In [54]:
```
tweets_location_df[['Trump Positive (in %)', 'Trump Negative (in %)',
       'Trump Neutral (in %)','Biden Positive (in %)','Biden Negative (in %)',
'Biden Neutral (in %)',
       'Predicted Judgement']]
```

Out[54]:

	Trump Positive (in %)	Trump Negative (in %)	Trump Neutral (in %)	Biden Positive (in %)	Biden Negative (in %)	Biden Neutral (in %)	Predicted Judgement
0	30	22	48	42	26	32	Somewhat Democratic
1	29	0	71	40	20	40	Insufficient Data
2	25	37	37	19	26	55	Somewhat Republican
3	44	28	28	14	29	57	Strongly Republican
4	28	36	36	21	36	43	Somewhat Republican
5	19	35	47	24	41	35	Somewhat Democratic
6	18	50	32	38	12	50	Strongly Democratic
7	0	75	25	0	0	100	Insufficient Data
8	28	34	38	27	35	39	Strongly Republican
9	31	31	39	36	30	33	Strongly Democratic
10	23	38	38	0	25	75	Insufficient Data
11	38	25	38	50	50	0	Insufficient Data
12	24	47	29	19	42	39	Somewhat Republican
13	40	40	20	37	32	32	Somewhat Republican
14	41	41	18	50	50	0	Somewhat Democratic
15	9	64	27	12	50	38	Insufficient Data
16	28	56	17	19	43	38	Somewhat Republican
17	16	47	37	14	43	43	Somewhat Republican
18	30	45	25	25	38	38	Somewhat Republican
19	36	39	25	38	38	25	Strongly Democratic
20	26	32	43	15	21	64	Somewhat Republican
21	16	43	41	10	45	45	Strongly Republican

	Trump Positive (in %)	Trump Negative (in %)	Trump Neutral (in %)	Biden Positive (in %)	Biden Negative (in %)	Biden Neutral (in %)	Predicted Judgement
22	17	65	19	12	38	50	Somewhat Republican
23	29	71	0	0	17	83	Insufficient Data
24	20	50	30	22	50	28	Somewhat Democratic
25	40	40	20	33	33	33	Insufficient Data
26	43	43	14	14	43	43	Insufficient Data
27	33	39	28	21	42	37	Strongly Republican
28	9	36	55	14	29	57	Insufficient Data
29	33	41	25	24	22	54	Somewhat Republican
30	25	33	42	30	30	40	Strongly Democratic
31	33	33	33	29	0	71	Insufficient Data
32	21	42	37	27	36	38	Strongly Democratic
33	67	33	0	25	50	25	Insufficient Data
34	44	19	37	34	30	36	Strongly Republican
35	33	33	33	20	50	30	Strongly Republican
36	23	50	27	20	48	32	Somewhat Republican
37	30	36	34	27	37	35	Strongly Republican
38	50	50	0	50	50	0	Insufficient Data
39	40	25	35	22	44	33	Strongly Republican
40	0	100	0	50	50	0	Insufficient Data
41	32	34	34	9	49	43	Strongly Republican
42	30	39	31	22	37	41	Somewhat Republican
43	21	29	50	17	33	50	Insufficient Data
44	0	0	0	0	100	0	Insufficient Data

	Trump Positive (in %)	Trump Negative (in %)	Trump Neutral (in %)	Biden Positive (in %)	Biden Negative (in %)	Biden Neutral (in %)	Predicted Judgement
45	27	32	41	30	30	40	Strongly Democratic
46	26	34	40	30	30	40	Strongly Democratic
47	0	75	25	11	22	67	Insufficient Data
48	32	41	27	13	40	47	Somewhat Republican
49	0	0	0	0	20	80	Insufficient Data

In [55]:
```
tweets_location_df.reset_index(inplace=True)
```

In [56]:
```
tweets_location_df
```

Out[56]:

	index	State	Trump Positive (in %)	Trump Negative (in %)	Trump Neutral (in %)	Trump Total Mentions	Biden Positive (in %)	Biden Negative (in %)	Biden Neutral (in %)	Me
0	0	Alabama	30	22	48	23	42	26	32	
1	1	Alaska	29	0	71	7	40	20	40	
2	2	Arizona	25	37	37	51	19	26	55	
3	3	Arkansas	44	28	28	18	14	29	57	
4	4	California	28	36	36	271	21	36	43	
5	5	Colorado	19	35	47	43	24	41	35	
6	6	Connecticut	18	50	32	28	38	12	50	
7	7	Delaware	0	75	25	4	0	0	100	
8	8	Florida	28	34	38	188	27	35	39	
9	9	Georgia	31	31	39	49	36	30	33	
10	10	Hawaii	23	38	38	13	0	25	75	
11	11	Idaho	38	25	38	8	50	50	0	
12	12	Illinois	24	47	29	59	19	42	39	
13	13	Indiana	40	40	20	30	37	32	32	
14	14	Iowa	41	41	18	17	50	50	0	
15	15	Kansas	9	64	27	11	12	50	38	
16	16	Kentucky	28	56	17	18	19	43	38	
17	17	Louisiana	16	47	37	19	14	43	43	
18	18	Maine	30	45	25	20	25	38	38	
19	19	Maryland	36	39	25	28	38	38	25	
20	20	Massachusetts	26	32	43	47	15	21	64	
21	21	Michigan	16	43	41	61	10	45	45	

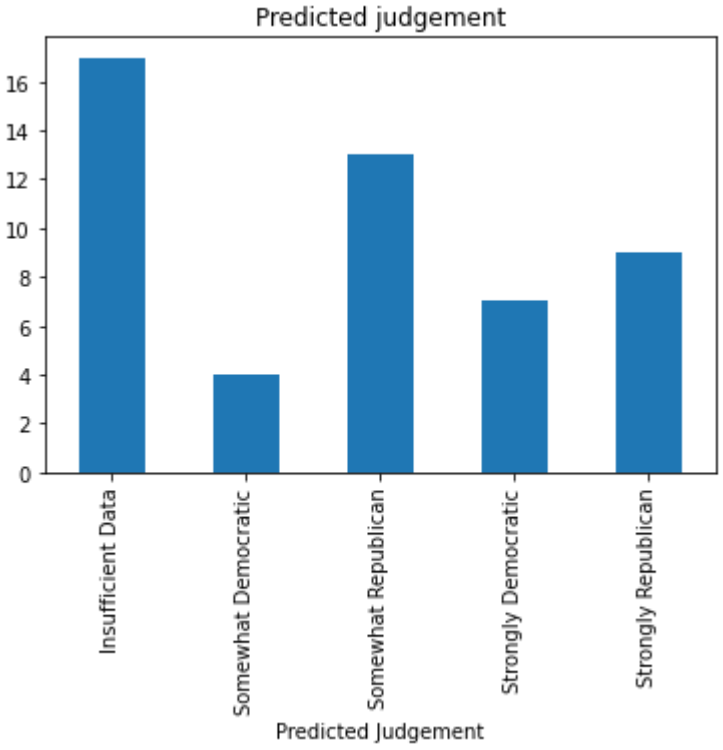
	index	State	Trump Positive (in %)	Trump Negative (in %)	Trump Neutral (in %)	Trump Total Mentions	Biden Positive (in %)	Biden Negative (in %)	Biden Neutral (in %)	Me
22	22	Minnesota	17	65	19	48	12	38	50	
23	23	Mississippi	29	71	0	7	0	17	83	
24	24	Missouri	20	50	30	20	22	50	28	
25	25	Montana	40	40	20	5	33	33	33	
26	26	Nebraska	43	43	14	7	14	43	43	
27	27	Nevada	33	39	28	18	21	42	37	
28	28	New Hampshire	9	36	55	11	14	29	57	
29	29	New Jersey	33	41	25	51	24	22	54	
30	30	New York	25	33	42	141	30	30	40	
31	31	New Mexico	33	33	33	12	29	0	71	
32	32	North Carolina	21	42	37	67	27	36	38	
33	33	North Dakota	67	33	0	3	25	50	25	
34	34	Ohio	44	19	37	54	34	30	36	
35	35	Oklahoma	33	33	33	21	20	50	30	
36	36	Oregon	23	50	27	30	20	48	32	
37	37	Pennsylvania	30	36	34	80	27	37	35	
38	38	Rhode Island	50	50	0	2	50	50	0	
39	39	South Carolina	40	25	35	20	22	44	33	
40	40	South Dakota	0	100	0	1	50	50	0	
41	41	Tennessee	32	34	34	38	9	49	43	
42	42	Texas	30	39	31	182	22	37	41	
43	43	Utah	21	29	50	14	17	33	50	
44	44	Vermont	0	0	0	0	0	100	0	

	index	State	Trump Positive (in %)	Trump Negative (in %)	Trump Neutral (in %)	Trump Total Mentions	Biden Positive (in %)	Biden Negative (in %)	Biden Neutral (in %)	Me
45	45	Virginia	27	32	41	44	30	30	40	
46	46	Washington	26	34	40	77	30	30	40	
47	47	West Virginia	0	75	25	8	11	22	67	
48	48	Wisconsin	32	41	27	44	13	40	47	
49	49	Wyoming	0	0	0	0	0	20	80	

In [57]:
```
df6=tweets_location_df.groupby('Predicted Judgement').size()
```

In [58]:
```
df6.plot.bar(title='Predicted judgement')
```

Out[58]:
```
<AxesSubplot:title={'center':'Predicted judgement'}, xlabel='Predicted Judgem
ent'>
```



APPLYING LSTM MODEL TO PREDICT THE ACCURACY

In [59]:
```
df2
```

Out[59]:

	id	text	location	subject	time_created	state	sen
0	1315780274911801344	“I’m running as a proud Democrat, for the Sen...	Wyoming, USA	Biden	2020-10-12	Wyoming	{'neg' 'neu': 'pos'
1	1315780275742404614	Biden says he was raised in a Black Baptist C...	Georgia, USA	Biden	2020-10-12	Georgia	{'ne 'ne 'pc 'comp
2	1315780275876421633	Joe Biden: "I'm running as a proud Democrat f...	Los Angeles, CA	Biden	2020-10-12	California	{'ne 'neu': 'pos': '
3	1315780277168410624	📺 NEW VIDEORetweet if you are ready to vote B...	Fairfax, VA	Biden	2020-10-12	Virginia	{'neg': 'neu': 'pos':
4	1315780277617254401	Just to be clear Biden and Harris would be fi...	Brookfield, MO, USA	Biden	2020-10-12	Missouri	{'neg': 'neu': 'pos':
...	...	...	...	...	...	...	
3316	1315781636533952512	I just hit the live feed on of Trump’s rally...	Washington, D.C.	Trump	2020-10-12	Washington	{'neg': 'neu': 'pc '
3317	1315781636630499328	Absolutely Austin. Trump was always going to ...	Brooklyn, NY	Trump	2020-10-12	New York	{'neg': 'neu': 'pc '
3318	1315781636890320897	That’s “President Trump” sir...	Iowa	Trump	2020-10-12	Iowa	{'ne 'ne 'pc 'comp
3319	1315781637137989634	should be ashamed if they allow this. It’s...	Virginia	Trump	2020-10-12	Virginia	{'neg': 'neu': 'pos':
3320	1315781637158907905	So true I guess the Biden supporters missed...	Ohio, USA	Trump	2020-10-12	Ohio	{'neg': 'neu': 'pos':

3321 rows × 8 columns

In [60]:
```
tweets=df2[['text','subject','sentiment_overall']]
```

In [61]:
```
tweets
```

Out[61]:

	text	subject	sentiment_overall
0	“I’m running as a proud Democrat, for the Sen...	Biden	Neutral
1	Biden says he was raised in a Black Baptist C...	Biden	Neutral
2	Joe Biden: "I'm running as a proud Democrat f...	Biden	Positive
3	📺 NEW VIDEORetweet if you are ready to vote B...	Biden	Negative
4	Just to be clear Biden and Harris would be fi...	Biden	Positive
...	...	...	...
3316	I just hit the live feed on of Trump’s rally...	Trump	Negative
3317	Absolutely Austin. Trump was always going to ...	Trump	Negative
3318	That’s “President Trump” sir...	Trump	Neutral
3319	should be ashamed if they allow this. It’s...	Trump	Negative
3320	So true I guess the Biden supporters missed...	Trump	Positive

3321 rows × 3 columns

In [64]:
```
max_fatures = 2000
tokenizer = Tokenizer(num_words=max_fatures, split=' ')
tokenizer.fit_on_texts(tweets['text'].values)
X = tokenizer.texts_to_sequences(tweets['text'].values)
X = pad_sequences(X)
X[:2]
```

Out[64]:
```
array([[  0,   0,   0,   0,   0,   0,   0,  39,  12,  16,   6,  32,  29,
          7,   1,  40,  24,  11,   4,  19,  30,  20,  22,   3,  28,  38,
         14,   5,  18,  43],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   4,  84,  13,
         36, 485,   8,   6, 228, 600, 435, 397,   6, 560,  37, 486,  13,
         36,   6, 679, 265]])
```

embed_dim, lstm_out, batch_size, droupout_x variables are hyperparameters, their values are somehow intuitive, can be and must be played with in order to achieve good results.

In [65]:
```
embed_dim = 128
lstm_out = 196

model = Sequential()
model.add(Embedding(max_fatures, embed_dim,input_length = X.shape[1]))
model.add(SpatialDropout1D(0.4))
model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(3,activation='softmax'))
model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = [
'accuracy'])
print(model.summary())
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 30, 128)           256000
_________________________________________________________________
spatial_dropout1d_1 (Spatial (None, 30, 128)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 196)               254800
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 591
=================================================================
Total params: 511,391
Trainable params: 511,391
Non-trainable params: 0
_________________________________________________________________
None
```

In [66]:
```
#training and testing data

Y = pd.get_dummies(tweets['sentiment_overall']).values
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.20, ran
dom_state = 42)
print(X_train.shape,Y_train.shape)
print(X_test.shape,Y_test.shape)
```

```
(2656, 30) (2656, 3)
(665, 30) (665, 3)
```

In [67]:
```
batch_size = 128
model.fit(X_train, Y_train, epochs = 15, batch_size=batch_size, verbose = 1)
```

```
C:\Users\SUMANA\Anaconda3\lib\site-packages\tensorflow_core\python\framework
\indexed_slices.py:433: UserWarning: Converting sparse IndexedSlices to a den
se Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "

Epoch 1/15
2656/2656 [==============================] - 5s 2ms/step - loss: 1.0398 - acc
uracy: 0.5087
Epoch 2/15
2656/2656 [==============================] - 4s 2ms/step - loss: 0.7605 - acc
uracy: 0.6156
Epoch 3/15
2656/2656 [==============================] - 4s 1ms/step - loss: 0.5364 - acc
uracy: 0.7828
Epoch 4/15
2656/2656 [==============================] - 3s 1ms/step - loss: 0.3906 - acc
uracy: 0.8554
Epoch 5/15
2656/2656 [==============================] - 3s 1ms/step - loss: 0.2823 - acc
uracy: 0.8927
Epoch 6/15
2656/2656 [==============================] - 3s 1ms/step - loss: 0.2236 - acc
uracy: 0.9187
Epoch 7/15
2656/2656 [==============================] - 3s 1ms/step - loss: 0.1717 - acc
uracy: 0.9413
Epoch 8/15
2656/2656 [==============================] - 3s 1ms/step - loss: 0.1399 - acc
uracy: 0.9541
Epoch 9/15
2656/2656 [==============================] - 3s 1ms/step - loss: 0.1171 - acc
uracy: 0.9620
Epoch 10/15
2656/2656 [==============================] - 4s 1ms/step - loss: 0.1015 - acc
uracy: 0.9714: 2s - loss:
Epoch 11/15
2656/2656 [==============================] - 4s 1ms/step - loss: 0.0785 - acc
uracy: 0.9748
Epoch 12/15
2656/2656 [==============================] - 4s 1ms/step - loss: 0.0592 - acc
uracy: 0.9800
Epoch 13/15
2656/2656 [==============================] - 4s 1ms/step - loss: 0.0575 - acc
uracy: 0.9834
Epoch 14/15
2656/2656 [==============================] - 4s 2ms/step - loss: 0.0538 - acc
uracy: 0.9857
Epoch 15/15
2656/2656 [==============================] - 5s 2ms/step - loss: 0.0537 - acc
uracy: 0.9868
```

Out[67]:
```
<keras.callbacks.callbacks.History at 0x2467014d288>
```

In [68]:
```
#validation set, and measuring score and accuracy

Y_pred = model.predict_classes(X_test,batch_size = batch_size)
```

In [69]:
```
df_test = pd.DataFrame({'true': Y_test.tolist(), 'pred':Y_pred})
df_test['true'] = df_test['true'].apply(lambda x: np.argmax(x))
print("confusion matrix",confusion_matrix(df_test.true, df_test.pred))
print(classification_report(df_test.true, df_test.pred))
```

```
confusion matrix [[240  13   7]
 [ 21 204  12]
 [ 20   8 140]]
              precision    recall  f1-score   support

           0       0.85      0.92      0.89       260
           1       0.91      0.86      0.88       237
           2       0.88      0.83      0.86       168

    accuracy                           0.88       665
   macro avg       0.88      0.87      0.88       665
weighted avg       0.88      0.88      0.88       665
```

In [70]:
```
# Separate majority and minority classes
data_majority = tweets[tweets['sentiment_overall'] == 'Negative']
data_minority = tweets[tweets['sentiment_overall'] == 'Positive']

bias = data_minority.shape[0]/data_majority.shape[0]
# lets split train/test data first then
train = pd.concat([data_majority.sample(frac=0.8,random_state=200),
         data_minority.sample(frac=0.8,random_state=200)])
test = pd.concat([data_majority.drop(data_majority.sample(frac=0.8,random_stat
e=200).index),
        data_minority.drop(data_minority.sample(frac=0.8,random_state=200).ind
ex)])

train = shuffle(train)
test = shuffle(test)
```

In [71]:
```
print('positive data in training:',(train.sentiment_overall == 'Positive').sum
())
print('negative data in training:',(train.sentiment_overall == 'Negative').sum
())
print('positive data in test:',(test.sentiment_overall == 'Positive').sum())
print('negative data in test:',(test.sentiment_overall == 'Negative').sum())
```

```
positive data in training: 692
negative data in training: 974
positive data in test: 173
negative data in test: 243
```

In [72]:
```
# Separate majority and minority classes in training data for upsampling
data_majority = train[train['sentiment_overall'] == 'Negative']
data_minority = train[train['sentiment_overall'] == 'Positive']

print("majority class before upsample:",data_majority.shape)
print("minority class before upsample:",data_minority.shape)

# Upsample minority class
data_minority_upsampled = resample(data_minority,
                                   replace=True,     # sample with replacement
                                   n_samples= data_majority.shape[0],    # to ma
tch majority class
                                   random_state=123) # reproducible results

# Combine majority class with upsampled minority class
data_upsampled = pd.concat([data_majority, data_minority_upsampled])

# Display new class counts
print("After upsampling\n",data_upsampled.sentiment_overall.value_counts(),sep
= "")

max_fatures = 2000
tokenizer = Tokenizer(num_words=max_fatures, split=' ')
tokenizer.fit_on_texts(tweets['text'].values) # training with whole data

X_train = tokenizer.texts_to_sequences(data_upsampled['text'].values)
X_train = pad_sequences(X_train,maxlen=29)
Y_train = pd.get_dummies(data_upsampled['sentiment_overall']).values
print('x_train shape:',X_train.shape)

X_test = tokenizer.texts_to_sequences(test['text'].values)
X_test = pad_sequences(X_test,maxlen=29)
Y_test = pd.get_dummies(test['sentiment_overall']).values
print("x_test shape", X_test.shape)
```

```
majority class before upsample: (974, 3)
minority class before upsample: (692, 3)
After upsampling
Negative    974
Positive    974
Name: sentiment_overall, dtype: int64
x_train shape: (1948, 29)
x_test shape (416, 29)
```

In [73]:
```
# model
embed_dim = 128
lstm_out = 192

model = Sequential()
model.add(Embedding(max_fatures, embed_dim,input_length = X_train.shape[1]))
model.add(SpatialDropout1D(0.4))
model.add(LSTM(lstm_out, dropout=0.4, recurrent_dropout=0.4))
model.add(Dense(2,activation='softmax'))
model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = [
'accuracy'])
print(model.summary())
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 29, 128)           256000
_________________________________________________________________
spatial_dropout1d_2 (Spatial (None, 29, 128)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 192)               246528
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 386
=================================================================
Total params: 502,914
Trainable params: 502,914
Non-trainable params: 0
_________________________________________________________________
None
```

In [74]:
```
batch_size = 128
# also adding weights
class_weights = {0: 1 ,
                 1: 1.6/bias }
model.fit(X_train, Y_train, epochs = 15, batch_size=batch_size, verbose = 1,
          class_weight=class_weights)
```

```
C:\Users\SUMANA\Anaconda3\lib\site-packages\tensorflow_core\python\framework
\indexed_slices.py:433: UserWarning: Converting sparse IndexedSlices to a den
se Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "

Epoch 1/15
1948/1948 [==============================] - 4s 2ms/step - loss: 1.0138 - acc
uracy: 0.4949
Epoch 2/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.8569 - acc
uracy: 0.5662
Epoch 3/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.5500 - acc
uracy: 0.8101
Epoch 4/15
1948/1948 [==============================] - 2s 1ms/step - loss: 0.3872 - acc
uracy: 0.8830
Epoch 5/15
1948/1948 [==============================] - 2s 1ms/step - loss: 0.2531 - acc
uracy: 0.9281
Epoch 6/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.1899 - acc
uracy: 0.9369
Epoch 7/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.1229 - acc
uracy: 0.9600
Epoch 8/15
1948/1948 [==============================] - 3s 2ms/step - loss: 0.0994 - acc
uracy: 0.9774
Epoch 9/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0918 - acc
uracy: 0.9784
Epoch 10/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0652 - acc
uracy: 0.9825
Epoch 11/15
1948/1948 [==============================] - 2s 1ms/step - loss: 0.0581 - acc
uracy: 0.9856
Epoch 12/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0468 - acc
uracy: 0.9902
Epoch 13/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0385 - acc
uracy: 0.9908
Epoch 14/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0377 - acc
uracy: 0.9908
Epoch 15/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0245 - acc
uracy: 0.9959
```

Out[74]: <keras.callbacks.callbacks.History at 0x2467ed62d08>

In [75]:
```
Y_pred = model.predict_classes(X_test,batch_size = batch_size)
df_test = pd.DataFrame({'true': Y_test.tolist(), 'pred':Y_pred})
df_test['true'] = df_test['true'].apply(lambda x: np.argmax(x))
print("confusion matrix",confusion_matrix(df_test.true, df_test.pred))
print(classification_report(df_test.true, df_test.pred))
```

```
confusion matrix [[232  11]
 [ 33 140]]
              precision    recall  f1-score   support

           0       0.88      0.95      0.91       243
           1       0.93      0.81      0.86       173

    accuracy                           0.89       416
   macro avg       0.90      0.88      0.89       416
weighted avg       0.90      0.89      0.89       416
```

In [76]:
```
# running model to few more epochs
model.fit(X_train, Y_train, epochs = 15, batch_size=batch_size, verbose = 1,
          class_weight=class_weights)
Y_pred = model.predict_classes(X_test,batch_size = batch_size)
df_test = pd.DataFrame({'true': Y_test.tolist(), 'pred':Y_pred})
df_test['true'] = df_test['true'].apply(lambda x: np.argmax(x))
print("confusion matrix",confusion_matrix(df_test.true, df_test.pred))
print(classification_report(df_test.true, df_test.pred))
```

```
Epoch 1/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0187 - acc
uracy: 0.9964
Epoch 2/15
1948/1948 [==============================] - 2s 1ms/step - loss: 0.0308 - acc
uracy: 0.9928
Epoch 3/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0172 - acc
uracy: 0.9959
Epoch 4/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0152 - acc
uracy: 0.9969
Epoch 5/15
1948/1948 [==============================] - 3s 2ms/step - loss: 0.0118 - acc
uracy: 0.9979
Epoch 6/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0145 - acc
uracy: 0.9969
Epoch 7/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0293 - acc
uracy: 0.9969
Epoch 8/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0153 - acc
uracy: 0.9959
Epoch 9/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0122 - acc
uracy: 0.9990
Epoch 10/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0071 - acc
uracy: 0.9990
Epoch 11/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0092 - acc
uracy: 0.9990
Epoch 12/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0075 - acc
uracy: 0.9974
Epoch 13/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0067 - acc
uracy: 0.9985
Epoch 14/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0049 - acc
uracy: 0.9985
Epoch 15/15
1948/1948 [==============================] - 3s 1ms/step - loss: 0.0035 - acc
uracy: 0.9995
confusion matrix [[228  15]
 [ 33 140]]
              precision    recall  f1-score   support

           0       0.87      0.94      0.90       243
           1       0.90      0.81      0.85       173

    accuracy                           0.88       416
   macro avg       0.89      0.87      0.88       416
weighted avg       0.89      0.88      0.88       416
```

In [79]:
```
scores=model.evaluate(X_test,Y_test,verbose=0)
print('Test accuracy:', scores[1])
```

```
Test accuracy: 0.8846153616905212
```

In [80]:
```
twt = ['Breaking: President Trump has tested negative']
#vectorizing the tweet by the pre-fitted tokenizer instance
twt = tokenizer.texts_to_sequences(twt)
#padding the tweet to have exactly the same shape as `embedding_2` input
twt = pad_sequences(twt, maxlen=29, dtype='int32', value=0)
print(twt)
sentiment = model.predict(twt,batch_size=1,verbose = 2)[0]
if(np.argmax(sentiment) == 0):
    print("negative")
elif (np.argmax(sentiment) == 1):
    print("positive")
```

```
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0 168  26   2  35 129 119]]
negative
```

In [81]:
```
twt = ['inaccurate facts, dont vote for him']
#vectorizing the tweet by the pre-fitted tokenizer instance
twt = tokenizer.texts_to_sequences(twt)
#padding the tweet to have exactly the same shape as `embedding_2` input
twt = pad_sequences(twt, maxlen=29, dtype='int32', value=0)
print(twt)
sentiment = model.predict(twt,batch_size=1,verbose = 2)[0]
if(np.argmax(sentiment) == 0):
    print("negative")
elif (np.argmax(sentiment) == 1):
    print("positive")
```

```
[[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0 49  7 81]]
negative
```

In [82]:
```
print('The Test accuracy is:', scores[1]*100)
```

```
The Test accuracy is: 88.46153616905212
```

CITATIONS Reffered class notes and notebooks https://towardsdatascience.com/sentiment-analysis-on-the-tweets-about-distance-learning-with-textblob-cc73702b48bc (https://towardsdatascience.com/sentiment-analysis-on-the-tweets-about-distance-learning-with-textblob-cc73702b48bc) https://towardsdatascience.com/sentiment-analysis-on-twitter-data-regarding-2020-us-elections-1de4bedbe866 (https://towardsdatascience.com/sentiment-analysis-on-twitter-data-regarding-2020-us-elections-1de4bedbe866) https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/ (https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/) https://medium.com/better-programming/twitter-sentiment-analysis-15d8892c0082 (https://medium.com/better-programming/twitter-sentiment-analysis-15d8892c0082) https://towardsdatascience.com/sentiment-analysis-using-lstm-step-by-step-50d074f09948 (https://towardsdatascience.com/sentiment-analysis-using-lstm-step-by-step-50d074f09948) https://medium.com/@lamiae.hana/a-step-by-step-guide-on-sentiment-analysis-with-rnn-and-lstm-3a293817e314 (https://medium.com/@lamiae.hana/a-step-by-step-guide-on-sentiment-analysis-with-rnn-and-lstm-3a293817e314) http://www.ijfis.org/journal/view.html?uid=809&&vmd=Full (http://www.ijfis.org/journal/view.html?uid=809&&vmd=Full)