PH1998 Machine Learning Sentiment Analysis Homework Assignment

```
from sklearn.datasets import load_files
import numpy as np
import mglearn


# lemma_vect = TfidfVectorizer(tokenizer=spacy_tokenizer, min_df=5)
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS


# Create a pipeline - tf–idf utilizes the statistical properties of the training data
# pipe = make_pipeline(lemma_vect, LogisticRegression())
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression


# Prepare spacy tokenizer
# From custom_file import spacy_tokenizer
import re
import spacy
from html import unescape
import matplotlib.pyplot as plt


# Used for downloading tweets
import re
import tweepy
from tweepy import OAuthHandler
from textblob import TextBlob
```

PART 1: BUILD A SENTIMENT ANALYZER TO DISTINGUISH BETWEEN POSITIVE AND NEGATIVE TWEETS

A) MACHINE LEARNING-BASED APPROACH

1. Train Model Utilizing Movie Reviews Dataset

```
path = "
reviews_train = load_files("C:\\Users\\yiyao\\OneDrive\\Documents\\UTHSC Fall
    2019\\aclImdb_v1\\aclImdb\\train")
# load_files generates both training texts and labels
text_train, y_train = reviews_train.data, reviews_train.target #[0:1000]
print("length of text_train: {}".format(len(text_train)))
print("text_train[6]:\n{}".format(text_train[6]))

print("Samples per class (training): {}".format(np.bincount(y_train)))

reviews_test = load_files("C:\\Users\\yiyao\\OneDrive\\Documents\\UTHSC Fall
    2019\\aclImdb_v1\\aclImdb\\test")
text_test, y_test = reviews_test.data, reviews_test.target

print("Number of documents in test data: {}".format(len(text_test)))
print("Samples per class (test): {}".format(np.bincount(y_test)))
```

```
length of text_train: 25000
text_train[6]: b
"This movie has a special way of telling the story, at first i found
it rather odd as it jumped through time and I had no idea whats
happening.<br /><br />Anyway the story line was although simple, but
still very real and touching. You met someone the first time, you fell
in love completely, but broke up at last and promoted a deadly agony.
Who hasn't go through this? but we will never forget this kind of pain
in our life. <br /><br />I would say i am rather touched as two actor
has shown great performance in showing the love between the
characters. I just wish that the story could be a happy ending."
Samples per class (training): [12500 12500]
Number of documents in test data: 25000
Samples per class (test): [12500 12500]
```

```
# Remove HTML line breaks (<br />)
text_train = [doc.replace(b"<br />", b" ") for doc in text_train]
text_test  = [doc.replace(b"<br />", b" ") for doc in text_test]
```

# Examine the dataset
print(text_train[1], y_train[1])

```
b'Words can\'t describe how bad this movie is. I can\'t explain it by writing
 only. You have too see it for yourself to get at grip of how horrible a movi
e really can be. Not that I recommend you to do that. There are so many clich
\xc3\xa9s, mistakes (and all other negative things you can imagine) here that
 will just make you cry. To start with the technical first, there are a LOT o
f mistakes regarding the airplane. I won\'t list them here, but just mention
the coloring of the plane. They didn\'t even manage to show an airliner in th
e colors of a fictional airline, but instead used a 747 painted in the origin
al Boeing livery. Very bad. The plot is stupid and has been done many times b
efore, only much, much better. There are so many ridiculous moments here that
 i lost count of it really early. Also, I was on the bad guys\' side all the
time in the movie, because the good guys were so stupid. "Executive Decision"
 should without a doubt be you\'re choice over this one, even the "Turbulence
"-movies are better. In fact, every other movie in the world is better than t
his one.' 0
```

 # Employed two methods for word processing:

```
1. Tfidf and removal of all English stop words

2. Applied lemmatization from spacy in order to remove English stop wo
    rds

# Note that lemmatization reduced the number of words in the vocabulary
 from 26900 to 20658!
```

# regexp employed in CountVectorizer
# Transform a regular expression pattern into a regular expression object
regexp = re.compile('(?u)\\b\\w\\w+\\b')

# Load spacy language model - save old tokenizer
en_nlp = spacy.load('en_core_web_sm')
old_tokenizer = en_nlp.tokenizer
# Replace tokenizer with preceding regexp
en_nlp.tokenizer = lambda string: old_tokenizer.tokens_from_list(regexp.findall(string))

# Create a custom tokenizer using the spacy document processing pipeline
def spacy_tokenizer(document):
doc_spacy = en_nlp(document, disable=['ner', 'parser', 'tagger'])
return [token.lemma_ for token in doc_spacy]

```
# Tokenize the training dataset with two methods
# Define a count vectorizer with the custom tokenizer
tfidf = TfidfVectorizer(min_df=5, stop_words="english")
X_train  = tfidf.fit_transform(text_train)
print("X_train.shape: {}".format(X_train.shape))

lemma_tfidf = TfidfVectorizer(tokenizer=spacy_tokenizer, min_df=5)
X_train_lemma = lemma_tfidf.fit_transform(text_train)
print("X_train_lemma.shape: {}".format(X_train_lemma.shape))
```

```
X_train.shape: (25000, 26966)
X_train_lemma.shape: (25000, 20658)
```

```
#With respect to training the data, logistic regression was employed with Grid Search to
optimize the parameters on both pre-processed training datasets (with and without
lemmatization).
#Results indicate that the best C values for both training datasets are 1 and 10 with
corresponding best cross-validation scores of 0.888 and 0.889, respectively.
#Lemmatization method will be employed for testing on the dataset consisting of tweets.

# Apply Grid Search to lemmatized dataset
param_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}
grid = GridSearchCV(LogisticRegression(solver = 'lbfgs', max_iter  = 400), param_grid, cv=5,
        n_jobs=-1)
grid.fit(X_train, y_train)
print("Best cross-validation score for data without lemmatization:
        {:.2f}".format(grid.best_score_))
grid_lemma = GridSearchCV(LogisticRegression(solver = 'lbfgs', max_iter  = 400), param_grid,
        cv=5, n_jobs=-1)
grid_lemma.fit(X_train_lemma, y_train)
print("Best cross-validation score for data with lemmatization:
{:.2f}".format(grid_lemma.best_score_))
```

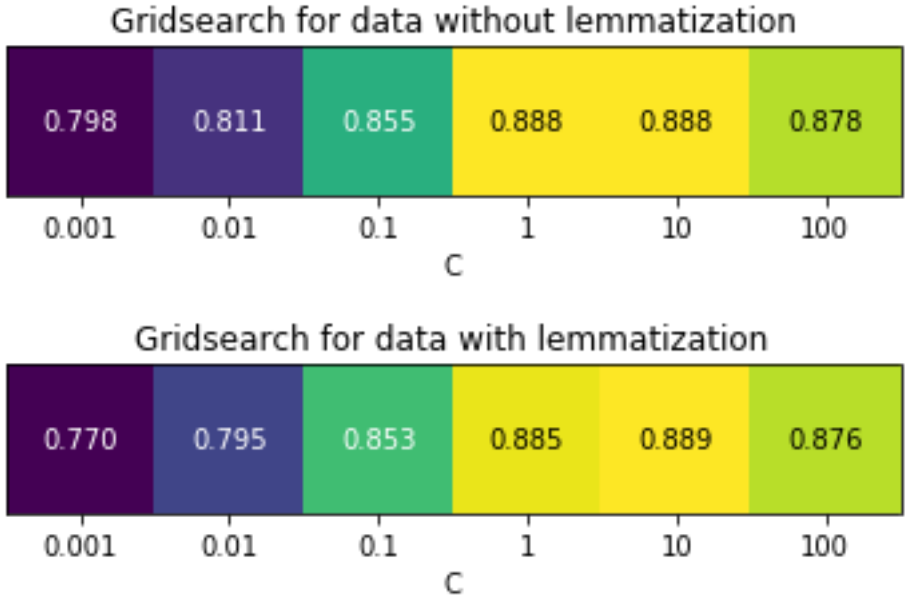
```
Best cross-validation score for data without lemmatization: 0.89
Best cross-validation score for data with lemmatization: 0.89
```

```
print("Best parameters for data without lemmatization:\n{}".format(grid.best_params_))
print("Best parameters for data with lemmatization:\n{}".format(grid_lemma.best_params_))
```

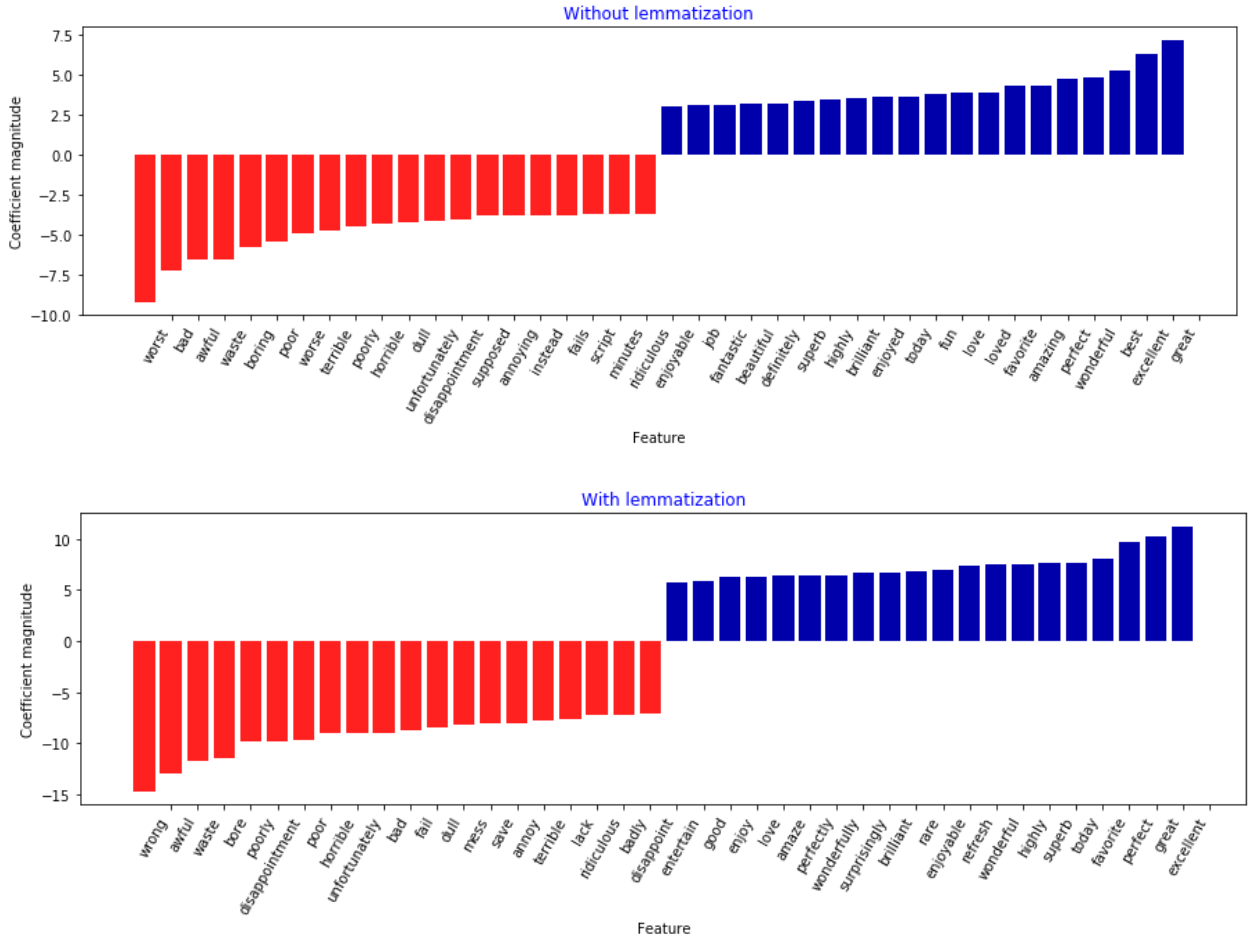
```
Best parameters for data without lemmatization:
{'C': 1}
Best parameters for data with lemmatization:
{'C': 10}
```

```
# Visualize dataset using heat map
fig, ax = plt.subplots(2)
# Extract scores from grid_search
scores = grid_lemma.cv_results_['mean_test_score'].reshape(6, -1).T
heatmap = mglearn.tools.heatmap(scores, xlabel="C", ylabel="", cmap="viridis", fmt="%.3f",
            xticklabels=param_grid['C'], yticklabels='', ax = ax[1])
ax[1].set_title('Gridsearch for data with lemmatization')
scores = grid.cv_results_['mean_test_score'].reshape(6, -1).T
heatmap = mglearn.tools.heatmap(scores, xlabel="C", ylabel="", cmap="viridis", fmt="%.3f",
            xticklabels=param_grid['C'], yticklabels='', ax = ax[0])
ax[0].set_title('Gridsearch for data without lemmatization')
```

```
Text(0.5, 1.0, 'Gridsearch for data without lemmatization')
```




```
# Explore feature importance in each of the models based on Grid Search
fig , axs = plt.subplots(2)
for ax, wordprocessing, gs, title in zip(axs, [tfidf, lemma_tfidf], [grid, grid_lemma], ['Without
        lemmatization', 'With lemmatization']):
feature_names = np.array(wordprocessing.get_feature_names())
mglearn.tools.visualize_coefficients(gs.best_estimator_.coef_, feature_names,
        n_top_features=20)
 plt.title(title, color = 'blue')
```



2. Test on Tweets Dataset

```
# Prepare access to Twitter data
# consumer_key = 'fill_in_yours'
# consumer_secret = 'fill_in_yours'
# access_token = 'fill_in_yours'
# access_token_secret = 'fill_in_yours'
# create OAuthHandler object
auth = OAuthHandler(consumer_key, consumer_secret)
# Set access token and secret
auth.set_access_token(access_token, access_token_secret)
# Create tweepy API object to fetch tweets
api = tweepy.API(auth)

# Download tweets for usage in testing model
tweet_topic_dictionary = {} #empty dictionary containing {topic:[list of tweet for topic]}
topics = ['impeachment', 'cybertruck', 'blackfriday']
maxTweets = 3000
```

```
for topic in topics:
    tweet_for_topics = []
    searchQuery = topic  # Topic of interest
    maxTweets = maxTweets
    tweetsPerQry = 100  # Max no. of tweets permitted by API

    # Set since_id to that ID if results from a specific ID onwards are required
    # Otherwise, default to lower limit
    sinceId = None

    # Set max_id to that ID if results are below a specific ID
    # Otherwise, default to upper limit
    max_id = -1

    tweetCount = 0
    print("-"*60 + "\nDownloading total of {0} tweets for topic {1}\n".format(maxTweets,
str.upper(topic)) + 60*"-" )

    while tweetCount < maxTweets:
        try:
            if (max_id <= 0):
                if (not sinceId):
                    new_tweets = api.search(q=searchQuery, count=tweetsPerQry, languages=['en'],
tweet_mode = 'extended')

                else:
                    new_tweets = api.search(q=searchQuery, count=tweetsPerQry,
                                            since_id=sinceId, languages=['en'], tweet_mode = 'extended')
            else:
                if (not sinceId):
                    new_tweets = api.search(q=searchQuery, count=tweetsPerQry,
                                            max_id=str(max_id - 1), languages=['en'], tweet_mode = 'extended')
                else:
                    new_tweets = api.search(q=searchQuery, count=tweetsPerQry,
                                            max_id=str(max_id - 1),
                                            since_id=sinceId, languages=['en'], tweet_mode = 'extended')
            if not new_tweets:
                print("No more tweets found")
                break
            for tweet in new_tweets:
                # Download the full tweet and retweets
                try:
                    tweet_text = tweet.retweeted_status.full_text
                    #print(tweet_text)
                    if tweet.lang == "en":
```

```
                # Clean tweet text by removing links, special characters using simple regex
                statements
                parsed_tweet = ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", "
                ", tweet_text).split())

                # Print('parsed_tweet: ', parsed_tweet)
                tweet_for_topics.append(parsed_tweet)
                # Filter tweets (only append the tweets in English)
            except:
                pass
        tweetCount += len(new_tweets)
        print("Downloaded {0} tweets. Getting total of {1} tweets after removing those not in
English".format(tweetCount, len(tweet_for_topics)))
        max_id = new_tweets[-1].id
      except tweepy.TweepError as e:
        # Exit if any error occurs
        print("some error : " + str(e))
        break
   tweet_topic_dictionary[topic] = tweet_for_topics


------------------------------------------------------------
Downloading total of 3000 tweets for topic IMPEACHMENT
------------------------------------------------------------
Downloaded 100 tweets. Getting total of 91 tweets after removing those
not in English
Downloaded 200 tweets. Getting total of 184 tweets after removing
those not in English
Downloaded 300 tweets. Getting total of 275 tweets after removing
those not in English
Downloaded 400 tweets. Getting total of 366 tweets after removing
those not in English
Downloaded 500 tweets. Getting total of 453 tweets after removing
those not in English
Downloaded 600 tweets. Getting total of 546 tweets after removing
those not in English
Downloaded 700 tweets. Getting total of 634 tweets after removing
those not in English
Downloaded 800 tweets. Getting total of 724 tweets after removing
those not in English
Downloaded 900 tweets. Getting total of 819 tweets after removing
those not in English
Downloaded 1000 tweets. Getting total of 909 tweets after removing
those not in English
Downloaded 1100 tweets. Getting total of 1001 tweets after removing
those not in English
Downloaded 1200 tweets. Getting total of 1091 tweets after removing
those not in English
```

```
Downloaded 1300 tweets. Getting total of 1179 tweets after removing
those not in English
Downloaded 1400 tweets. Getting total of 1266 tweets after removing
those not in English
Downloaded 1500 tweets. Getting total of 1352 tweets after removing
those not in English
Downloaded 1600 tweets. Getting total of 1443 tweets after removing
those not in English
Downloaded 1700 tweets. Getting total of 1535 tweets after removing
those not in English
Downloaded 1800 tweets. Getting total of 1625 tweets after removing
those not in English
Downloaded 1900 tweets. Getting total of 1718 tweets after removing
those not in English
Downloaded 2000 tweets. Getting total of 1807 tweets after removing
those not in English
Downloaded 2100 tweets. Getting total of 1893 tweets after removing
those not in English
Downloaded 2200 tweets. Getting total of 1987 tweets after removing
those not in English
Downloaded 2300 tweets. Getting total of 2078 tweets after removing
those not in English
Downloaded 2400 tweets. Getting total of 2166 tweets after removing
those not in English
Downloaded 2500 tweets. Getting total of 2257 tweets after removing
those not in English
Downloaded 2600 tweets. Getting total of 2348 tweets after removing
those not in English
Downloaded 2700 tweets. Getting total of 2433 tweets after removing
those not in English
Downloaded 2800 tweets. Getting total of 2521 tweets after removing
those not in English
Downloaded 2900 tweets. Getting total of 2614 tweets after removing
those not in English
Downloaded 3000 tweets. Getting total of 2707 tweets after removing
those not in English
------------------------------------------------------------
Downloading total of 3000 tweets for topic CYBERTRUCK
------------------------------------------------------------
Downloaded 100 tweets. Getting total of 60 tweets after removing those
not in English
Downloaded 200 tweets. Getting total of 112 tweets after removing
those not in English
Downloaded 300 tweets. Getting total of 173 tweets after removing
those not in English
Downloaded 400 tweets. Getting total of 240 tweets after removing
those not in English
Downloaded 500 tweets. Getting total of 304 tweets after removing
those not in English
Downloaded 600 tweets. Getting total of 374 tweets after removing
those not in English
```

```
Downloaded 700 tweets. Getting total of 440 tweets after removing
those not in English
Downloaded 800 tweets. Getting total of 516 tweets after removing
those not in English
Downloaded 900 tweets. Getting total of 590 tweets after removing
those not in English
Downloaded 1000 tweets. Getting total of 657 tweets after removing
those not in English
Downloaded 1100 tweets. Getting total of 730 tweets after removing
those not in English
Downloaded 1200 tweets. Getting total of 796 tweets after removing
those not in English
Downloaded 1300 tweets. Getting total of 871 tweets after removing
those not in English
Downloaded 1400 tweets. Getting total of 936 tweets after removing
those not in English
Downloaded 1500 tweets. Getting total of 1001 tweets after removing
those not in English
Downloaded 1600 tweets. Getting total of 1062 tweets after removing
those not in English
Downloaded 1700 tweets. Getting total of 1122 tweets after removing
those not in English
Downloaded 1800 tweets. Getting total of 1200 tweets after removing
those not in English
Downloaded 1900 tweets. Getting total of 1274 tweets after removing
those not in English
Downloaded 2000 tweets. Getting total of 1342 tweets after removing
those not in English
Downloaded 2100 tweets. Getting total of 1413 tweets after removing
those not in English
Downloaded 2200 tweets. Getting total of 1481 tweets after removing
those not in English
Downloaded 2300 tweets. Getting total of 1553 tweets after removing
those not in English
Downloaded 2400 tweets. Getting total of 1627 tweets after removing
those not in English
Downloaded 2500 tweets. Getting total of 1708 tweets after removing
those not in English
Downloaded 2600 tweets. Getting total of 1788 tweets after removing
those not in English
Downloaded 2700 tweets. Getting total of 1869 tweets after removing
those not in English
Downloaded 2800 tweets. Getting total of 1944 tweets after removing
those not in English
Downloaded 2900 tweets. Getting total of 2016 tweets after removing
those not in English
Downloaded 3000 tweets. Getting total of 2086 tweets after removing
those not in English
------------------------------------------------------------
Downloading total of 3000 tweets for topic BLACKFRIDAY
------------------------------------------------------------
```

Downloaded 85 tweets. Getting total of 56 tweets after removing those not in English
Downloaded 177 tweets. Getting total of 112 tweets after removing those not in English
Downloaded 259 tweets. Getting total of 160 tweets after removing those not in English
Downloaded 340 tweets. Getting total of 217 tweets after removing those not in English
Downloaded 423 tweets. Getting total of 269 tweets after removing those not in English
Downloaded 509 tweets. Getting total of 317 tweets after removing those not in English
Downloaded 591 tweets. Getting total of 365 tweets after removing those not in English
Downloaded 691 tweets. Getting total of 417 tweets after removing those not in English
Downloaded 777 tweets. Getting total of 472 tweets after removing those not in English
Downloaded 863 tweets. Getting total of 513 tweets after removing those not in English
Downloaded 943 tweets. Getting total of 563 tweets after removing those not in English
Downloaded 1030 tweets. Getting total of 620 tweets after removing those not in English
Downloaded 1116 tweets. Getting total of 672 tweets after removing those not in English
Downloaded 1209 tweets. Getting total of 719 tweets after removing those not in English
Downloaded 1304 tweets. Getting total of 779 tweets after removing those not in English
Downloaded 1396 tweets. Getting total of 826 tweets after removing those not in English
Downloaded 1482 tweets. Getting total of 869 tweets after removing those not in English
Downloaded 1568 tweets. Getting total of 914 tweets after removing those not in English
Downloaded 1657 tweets. Getting total of 966 tweets after removing those not in English
Downloaded 1748 tweets. Getting total of 1005 tweets after removing those not in English
Downloaded 1835 tweets. Getting total of 1053 tweets after removing those not in English
Downloaded 1926 tweets. Getting total of 1112 tweets after removing those not in English
Downloaded 2021 tweets. Getting total of 1162 tweets after removing those not in English
Downloaded 2113 tweets. Getting total of 1219 tweets after removing those not in English
Downloaded 2204 tweets. Getting total of 1269 tweets after removing those not in English
Downloaded 2293 tweets. Getting total of 1313 tweets after removing those not in English

```
Downloaded 2390 tweets. Getting total of 1372 tweets after removing
those not in English
Downloaded 2483 tweets. Getting total of 1430 tweets after removing
those not in English
Downloaded 2577 tweets. Getting total of 1494 tweets after removing
those not in English
Downloaded 2670 tweets. Getting total of 1556 tweets after removing
those not in English
Downloaded 2770 tweets. Getting total of 1618 tweets after removing
those not in English
Downloaded 2863 tweets. Getting total of 1667 tweets after removing
those not in English
Downloaded 2958 tweets. Getting total of 1721 tweets after removing
those not in English
Downloaded 3047 tweets. Getting total of 1766 tweets after removing
those not in English
```

```
X_test_from_twitter = tweet_topic_dictionary['impeachment'] +
tweet_topic_dictionary['cybertruck'] + tweet_topic_dictionary['blackfriday']
print('Total number of tweets for all topics {0}'.format(len(X_test_from_twitter)))
for tweet in X_test_from_twitter[:5]:
    print("--------------------")
    print(tweet)
```

```
Total number of tweets for all topics 6559
--------------------
Trump The whole impeachment thing is unfair I should be able to call evidence
 and present my own witnesses Nadler You have until Dec 6th to do exactly tha
t Trump Wait what You weren t supposed to agree I can t do that This is a per
jury trap
--------------------
I will be representing our Country in London at NATO while the Democrats are
holding the most ridiculous Impeachment hearings in history Read the Transcri
pts NOTHING was done or said wrong The Radical Left is undercutting our Count
ry Hearings scheduled on same dates as NATO
--------------------
Trump says he didn t direct Giuliani s Ukraine efforts Witnesses say otherwis
e Public testimony portrayed Giuliani as the driver behind Trump s pressure c
ampaign to get Ukraine to investigate Joe and Hunter Biden
--------------------
Found out at Thanksgiving that my mom amp 2 other relatives all elderly won t
 vote for him again All lifelong Rep s The impeachment trial tipped them into
 totally making up their minds We need to keep fighting It s starting to work
--------------------
House Judiciary Top Republican Collins letter to Chair Nadler on Wed s impeac
hment inquiry hearing An equal distribution of experts for the Dec 4 hearing
would be a small concession to demonstrate to the American people this impeac
hment inquiry is not merely political theater
```

```
# Evaluate the trained model with movie reviews on the 8000 downloaded tweets
X_test_lemma = lemma_tfidf.transform(X_test_from_twitter)
y_test_lemma = grid_lemma.predict(X_test_lemma)

# Formulate prediction for
X_test  = tfidf.transform(X_test_from_twitter)
y_test = grid.predict(X_test)
positive_tweets, negative_tweets = [], []
for x, y in zip(X_test_from_twitter, y_test):
    if y == 0:
        negative_tweets.append(x)
    else:
        positive_tweets.append(x)
# Explore Results

totaltweets = len(X_test_from_twitter)
# Percentage of positive tweets
print("Positive tweets percentage: {} %".format(len(positive_tweets)/totaltweets*100))
# Percentage of negative tweets
print("Negative tweets percentage: {} %".format(len(negative_tweets)/totaltweets*100))

b = [14, 100, 400, 500, 620, -400,-109, -1]

# Print first 5 positive tweets
print("\n\nPositive tweets:")
c = [positive_tweets[i] for i in b]
for tweet in c:
    print("--------------------")
    print(tweet)
# Print first 5 negative tweets
print("\n\nNegative tweets:")
c = [negative_tweets[i] for i in b]
for tweet in c:
    print("--------------------")
    print(tweet)
```

```
Positive tweets percentage: 38.37475224881842 %
Negative tweets percentage: 61.625247751181575 %

Positive tweets:
--------------------
Nadler thinks he s SLICK plays with FIRE Every time DemoNazis try to
trickwitness into implicating Trump they uncover more ObamaState
corruption prove Trump innocent You have to own your ignorance before
you can learn
--------------------
```

The first person who should testify in a Judiciary Committee hearing about impeachment is Chairman Adam Schiff He has A LOT to answer for
---
It is a tribute to an awakened Republican Party that despite the most strenuous no holds barred efforts by the media Left not a single Republican in the House or the Senate seems willing to go along with impeachment United like this the GOP is unstoppable
---
Congressman reacts to the latest in the battle with Democrats over the impeachment inquiry Check this out
---
Trump did nothing impeachable day by day review of all impeachment hearing testimony shows
---
It s time to Treat Yo self This year we re going all out with our special BlackFriday giveaway Win gaming mini PCs a i9 processor and games thanks to How to enter Easy click here and follow the instructions
---
ad blackfriday amazon echoshow echo afflink as an Amazon associate I earn from qualifying purchases
---
Rare Gold Players Pack SBC will be available for 2 hrs FUT20 BlackFriday

Negative tweets:
---
on Pres Trump s use of profanity to lash out against impeachment at a rally saying that if President Obama spoke that way Women would be fainting Children would be running and hiding under beds Grown men would be in tears
---
I will be representing our Country in London at NATO while the Democrats are holding the most ridiculous Impeachment hearings in history Read the Transcripts NOTHING was done or said wrong The Radical Left is undercutting our Country Hearings scheduled on same dates as NATO

Freshman swing district Democrat said she did not run to impeach President Donald Trump but VOTED for impeachment inquiry Any message for her about the BS sham impeachment deep state has initiated on our
---
The least Chairman Nadler can do to show an ounce of fairness to the American people and indicate that Judiciary proceedings will be more than raw political theater is allow an equal distribution of witnesses for Wednesday s impeachment hearing
---
Schiff s impeachment hearings wasting time when Congress must do real work
---

```
Antivirus Deals McAfee Total Protection 3 Device Antivirus Software
Internet Security 1 Year Subscription for 15 99 80 off deals PC geek
tech blackfriday web gaming ransomware Windows10 laptop infosec
------------------
Giving 10 000 to 3 people to celebrate BLACK FRIDAY and CYBER MONDAY
Retweet FOLLOW NE Notifications MUST be on or you can t win Comment
Money Ends in 24 hours BlackFriday2019 CyberMonday BlackFriday
BlackFridaySale CyberMonday2019
------------------
Happy BlackFriday I am still sending 350 13 500 to the first 300
people to like amp retweet this comment your cashapp google pay apple
pay or paypal down below HOLIDAY SPECIAL RETWEET amp COMMENT
```

B) KNOWLEDGE-BASED APPROACH USING TEXTBLOB (DEFAULT/ PATTERN ANALYZER)

```
from textblob import TextBlob
from textblob.classifiers import NaiveBayesClassifier
from textblob.classifiers import DecisionTreeClassifier

positive_from_textblob, negative_from_textblob, neutral_from_textblob = [], [] , []

for txt in X_test_from_twitter:
    analysis1 = TextBlob(txt)
    if analysis1.sentiment.polarity > 0:
        positive_from_textblob.append(txt)
    if analysis1.sentiment.polarity < 0:
        negative_from_textblob.append(txt)
    if analysis1.sentiment.polarity == 0:
        neutral_from_textblob.append(txt)

totaltweets = len(X_test_from_twitter)
# Percentage of positive tweets
print("Positive tweets percentage: {} %".format(len(positive_from_textblob)/totaltweets*100))
# Percentage of negative tweets
print("Negative tweets percentage: {} %".format(len(negative_from_textblob)/totaltweets*100))
# Percentage of neutral tweets
print("Neutral  tweets percentage: {} %".format(len(neutral_from_textblob)/totaltweets*100))

b =  [14, 100, 400, 500, 620, -400,-109, -1]

# Print first 5 positive tweets
print("\n\nPositive tweets:")
c = [positive_from_textblob[i] for i in b]
```

```
for tweet in c:
    print("--------------------")
    print(tweet)
# Print first 5 negative tweets
print("\n\nNegative tweets:")
c = [negative_from_textblob[i] for i in b]
for tweet in c:
    print("--------------------")
    print(tweet)
# Print first 5 neutral tweets
print("\n\nNeutral tweets:")
c = [neutral_from_textblob[i] for i in b]
for tweet in c:
    print("--------------------")
    print(tweet)
```

```
Positive tweets percentage: 43.7414239975606 %
Negative tweets percentage: 18.02103979265132 %
Neutral  tweets percentage: 38.237536209788075 %

Positive tweets:
--------------------
Nadler thinks he s SLICK plays with FIRE Every time DemoNazis try to
trickwitness into implicating Trump they uncover more ObamaState
corruption prove Trump innocent You have to own your ignorance before
you can learn
--------------------
Hey More often than not your dad only golfs and tweets The ONLY reason
he went to Afghanistan was because of impeachment The only pattern
here is he only does decent things for selfish reasons FUN FACT Obama
visited troops in Iraq after 5 months
--------------------
I want to wish all of my family here on Twitter a Happy Thanksgiving
with the safety of being with family and friends and the knowledge
that when you go back to work Trump will have to provide a good excuse
for withholding tax returns and The Impeachment Report is due
--------------------
Louie Gohmert Wants Three Witnesses at Nadler s Impeachment Hearings
Schiff Staffers Abigail Grace and Sean Misko and the Whistleblower via
--------------------
Congressman reacts to the latest in the battle with Democrats over the
impeachment inquiry Check this out
--------------------
SCOOP Walmart dodged tax on nearly 2 billion according to a
whistleblower filing sent to the IRS and leaked to me Happy
BlackFriday 1
--------------------
Giving 10 000 to 3 people to celebrate BLACK FRIDAY and CYBER MONDAY
Retweet FOLLOW NE Notifications MUST be on or you can t win Comment
```

Money Ends in 24 hours BlackFriday2019 CyberMonday BlackFriday BlackFridaySale CyberMonday2019

Rare Gold Players Pack SBC will be available for 2 hrs FUT20 BlackFriday

Negative tweets:

I will be representing our Country in London at NATO while the Democrats are holding the most ridiculous Impeachment hearings in history Read the Transcripts NOTHING was done or said wrong The Radical Left is undercutting our Country Hearings scheduled on same dates as NATO

President Trump now faces a critical choice whether to legitimize the impeachment inquiry proceedings by allowing his lawyers to participate or refuse to take part in an inquiry he says is a sham Analysis
---
I will be representing our Country in London at NATO while the Democrats are holding the most ridiculous Impeachment hearings in history Read the Transcripts NOTHING was done or said wrong The Radical Left is undercutting our Country Hearings scheduled on same dates as NATO

I will be representing our Country in London at NATO while the Democrats are holding the most ridiculous Impeachment hearings in history Read the Transcripts NOTHING was done or said wrong The Radical Left is undercutting our Country Hearings scheduled on same dates as NATO

Sarah Huckabee Sanders falsely claims that all witnesses during the impeachment hearings admit not to witnessing anything
---
With extreme effort Cybertruck might hit a 0 30 drag coefficient which would be insane for a truck Requires tweaking many small details
---
only a few hours left to shop my blackfriday sales Get up to 40 off at and 20 off at sales end at 11 59pm tonight
---
Are you Black Fridaying Coz when else are you gonna cop the things you ve been meaning to buy hii 2019 BlackFriday

Neutral tweets:

Trump did nothing impeachable day by day review of all impeachment hearing testimony shows
---
Democrats are focused on impeachment instead of the needs of the country like passing the USMCA reforming our asylum laws and getting an infrastructure bill passed This is the work left undone This is the opportunity cost of impeachment
---

```
These Dem Sens r running against Trump in 2020Elections Should these
Sens recuse themselves from sitting as a juror at Senate Impeachment
Trial
------------------
DC Court of Appeals to Hear Two Impeachment Cases on Jan 3rd May Throw
Off Democrats Shotgun Impeachment Plans via
------------------
What the Senate does or doesn t do is immaterial We can t worry about
them Since September 24 when Nancy Pelosi announced that she had
opened an Official Impeachment Inquiry we have been on offense That
means Trump has been and is on defense Our job is to keep it that way
------------------
My brother has been working on a mashed potato cybertruck for over an
hour
------------------
GET 20 off ALL orders all weekend using the code BLACKFRIDAY
------------------
Shop the KKW Fragrance Diamonds Trio Gift Set for only 45 get 20 30
off exclusions apply while supplies last for our BlackFriday sale at
KKWFRAGRANCE
```

PART 2: TOPIC MODELING AND DOCUMENT CLUSTERING ON TWEETS DATA

```
# Latent Dirichlet Allocation (LDA)

from sklearn.decomposition import LatentDirichletAllocation

# Eliminate extremely common words
# Top max_features ordered by term frequency

vect = CountVectorizer(max_features=5000, stop_words = 'english', max_df = 0.15)
X = vect.fit_transform(X_test_from_twitter)
feature_names = np.array(vect.get_feature_names())

# tfidf = TfidfVectorizer(min_df=5, stop_words="english")
# X  = tfidf.fit_transform(X_test_from_twitter)
# feature_names = np.array(tfidf.get_feature_names())

# lemma_tfidf = TfidfVectorizer(tokenizer=spacy_tokenizer, min_df=5)
# X = lemma_tfidf.fit_transform(X_test_from_twitter)
# feature_names = np.array(lemma_tfidf.get_feature_names())


# Learn a topic model with 7 topics
lda = LatentDirichletAllocation(n_topics=10
                                , learning_method="batch",
                                max_iter=50, random_state=0)
```

```
document_topics = lda.fit_transform(X)
print("lda.components_.shape: {}".format(lda.components_.shape))

# Examine the most important words for each of the topics
# Sort the features for each topic (a row in the components_)
# Invert rows with [:, ::-1] to allow for descended sorting
sorting = np.argsort(lda.components_, axis=1)[:, ::-1]
# Retrieve the feature names from the vectorizer

# Print the 10 topics:
mglearn.tools.print_topics(topics=range(10), feature_names=feature_names,
                           sorting=sorting, topics_per_chunk=5, n_words=10)
```

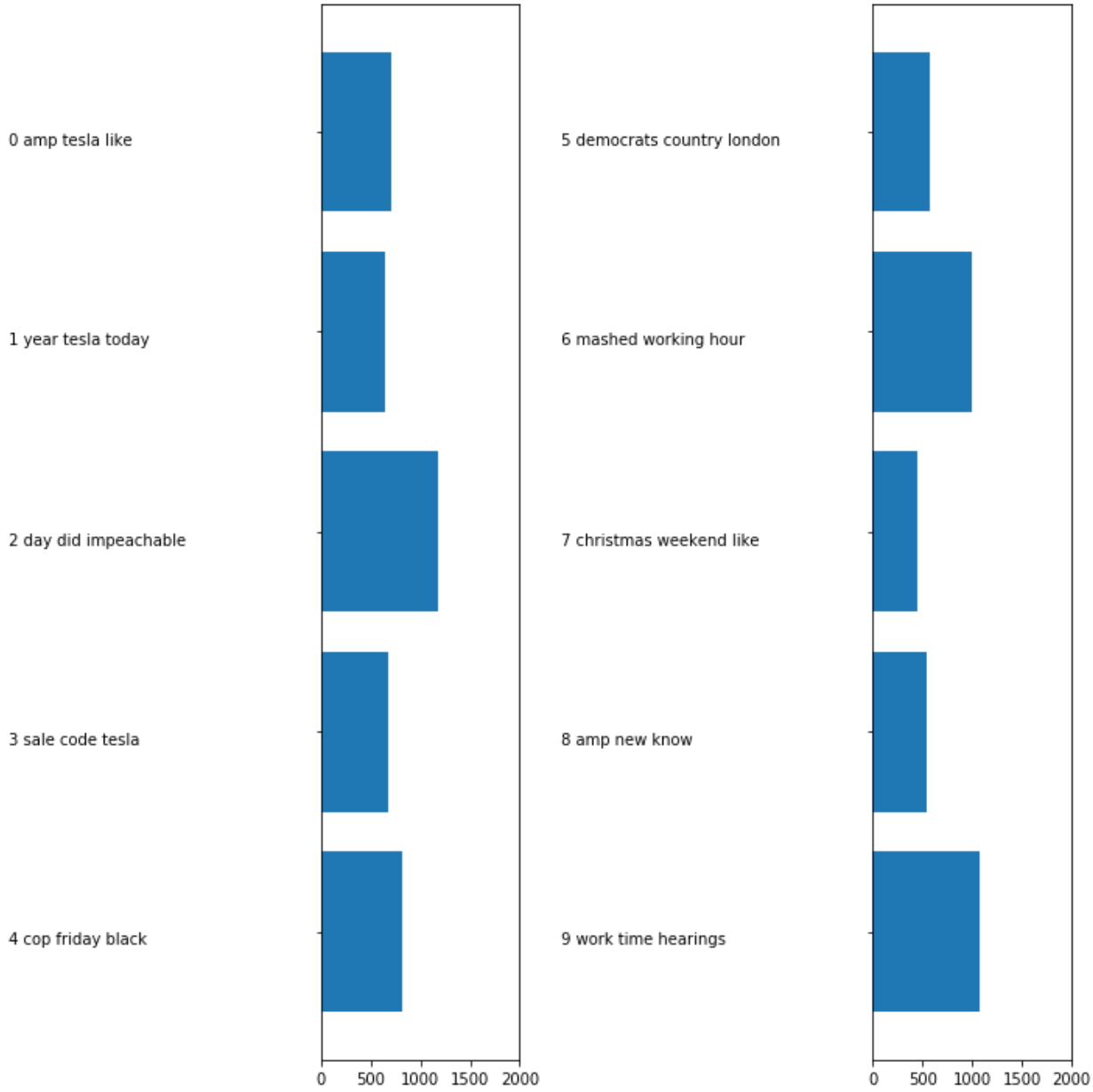
```
lda.components_.shape: (10, 5000)
topic 0       topic 1       topic 2       topic 3       topic 4
--------      --------      --------      --------      --------
amp           year          day           sale          cop
tesla         tesla         did           code          friday
like          today         impeachable   tesla         black
black         thanksgiving  testimony     use           10
friday        usmca         hearing       free          cyber
cybermonday   signed        review        buy           people
sale          pelosi        shows         20            monday
win           man           according     friday        celebrate
elon          house         billion       shop          follow
deals         democrats     happy         checkout      giving


topic 5       topic 6       topic 7       topic 8       topic 9
--------      --------      --------      --------      --------
democrats     mashed        christmas     amp           work
country       working       weekend       new           time
london        hour          like          know          hearings
holding       brother       sales         just          schiff
nato          potato        shop          record        real
ridiculous    just          black         giving        congress
representing  committee     chi           12            wasting
like          elon          deal          win           past
doing         musk          thing         thanksgiving  republican
level         list          mars          love          media
```

Topics 1, 3, 4, and 7 are related to Thanksgiving while Topic 9 is related to political issues.

```
# Investigate the types of reviews assigned to this topic
sale = np.argsort(document_topics[:, 3])[::-1] #sort by weight of topic 4
print(sale)
# Print the five documents for which topic is most important
for i in music[:5]:
    print(X_test_from_twitter[i])
[5687 5664 6528 ... 7045 5732 6864]
The Black Friday Sale Starts Today at Loomrack Up to 65 Off Sitewide
Buy 1 Item Get 25 Off Use Code BF25OFF Buy 2 Items Get 50 Off Use Code
BF50OFF Buy 3 Items Get 65 Off Use Code BF65OFF Shop BlackFriday sales
loomrack Cybertruck is Grade 301 Delorean was grade 304 Grade 304 is
still prone to corrosion and pitting Grade 301 is very prone to this
as it has less chromium and more carbon making it better for rockets
True stainless steel is grade 316 and above Bought a new computer on
BlackFriday cuz my old one had a hardware problem and was to slow and
I m gonna use photoshop and premiere pro now Gonna upload my yt video
tomorrow too I m excited but I never have time for everything It s
BlackFriday weekend and the deals are here Take advantage of these
sales from Bluehost Check out these BlackFridayDeals blog sale
CyberMonday bloggers selfhosted hosting WebHosting webshost
BlackFridayWeek BlackFridaySale forever kissing forever matching is
having a GIANT 50 off your order of 50 sale until DECEMBER 2 Just use
the code BLACKFRIDAY and that s it This amazing company donates one
headband for each

fig, ax = plt.subplots(1, 2, figsize=(10, 10))
topic_names = ["{:>2} ".format(i) + " ".join(words)
               for i, words in enumerate(feature_names[sorting[:, :3]])]
# Insert two columns bar chart:
for col in [0, 1]:
    start = col * 5
    end = (col + 1) * 5
    ax[col].barh(np.arange(5), np.sum(document_topics, axis=0)[start:end])
    ax[col].set_yticks(np.arange(5))
    ax[col].set_yticklabels(topic_names[start:end], ha="left", va="top")
    ax[col].invert_yaxis()
    ax[col].set_xlim(0, 2000)
    yax = ax[col].get_yaxis()
    yax.set_tick_params(pad=200)
plt.tight_layout()
```


0 amp tesla like
1 year tesla today
2 day did impeachable
3 sale code tesla
4 cop friday black
0
500
1000
1500
2000
5 democrats country london
6 mashed working hour
7 christmas weekend like
8 amp new know
9 work time hearings
0
500
1000
1500
2000

Discussion

In order to analyze sentiment analysis, one would need to employ both supervised classification machine learning-based and lexicon-based methods (e.g. Textblob default analyzer). Specifically, machine learning based techniques require two sets of documents consisting of a training and a test set. A training set is employed by an automatic classifier to learn the differentiating characteristics of documents while a test set is employed to verify the performance of the classifier. In this particular analysis, the train dataset is associated with movie reviews while the test dataset is associated with tweets. Since the training dataset only includes a binary label, the machine learning model is only capable of distinguishing between positive and negative sentiments based on knowledge extracted from movie reviews. Contrarily, sentiment analysis using Textblob employs a different algorithm to compute and average the positiveness of each input. Moreover, this method generates positive, negative, and neutral sentiment. Therefore, it may not be suitable to compare the percentages of positive and negative tweets derived from the aforementioned methods.

For a more holistic manifestation of the results, specifically with regard to topic modeling, I decided to modify the class code to obtain more pertinent tweets by ignoring those without English language and capturing full tweets. Nevertheless, an issue that I have encountered for some of the tweets relates to the fact that the model only assigned outputs as either positive or negative sentiments without the inclusion of neural sentiments. Perhaps a SVC model may be employed instead of the Logistic Regression model to resolve this issue.

It is also crucial to understand the key advantages and disadvantages associated with both methods. Specifically, while word pre-processing is required for both methods, the Textblob-based method is more simple and efficient in assigning polarity and subjectivity. Nevertheless, the machine learning-based method is more flexible in terms of model selection. With regard to a few major disadvantages, the machine learning-based method generally requires a large expert-annotated training corpora, which the majority of real-world data lack, while for the lexicon-based method, one may encounter the issue of over-analyzing or under-analyzing the results. One other key disadvantage of the lexicon-based method relates to the issue that the polarity of numerous words is both domain- and context-dependent, which results in the difficulty in assigning positive and negative sentiments for similar phrases.