Homework Assignment 2 - BIA 6304 - Corey Austen

Q1. Write a short description of the context of the dataset in your own words. Make sure your answer is no longer than three paragraphs, and should at minimum answer these questions: • Why did you choose the processing that you did? Give several specific examples. • What is the effect of the replacement on your feature space? Does this make sense? Is it helpful for answering your question? Why or why not?

Audience: technical – fellow data scientists or other technical staff.

A1. The dataset that I used was a csv file containing the text of Trump's State of the Union speech. The each row is a line from the speech, consisting of a 1-2 sentences, making it 149 rows. The question I am trying to answer is "What did Trump focus on primarily in this speech?" I chose to process this dataset with a count vectorizer, a min_df of 3%, max_df of 50%, an ngram_range or 1 - 2 word phrase. I chose to use a min_df of 3%, requiring a feature to appear in at least 4 documents, and a max_df of 50%, meaning that a feature is removed from the space if it appears in more than 74 documents. Through playing with the min_df and max_df at various levels, I found that these tend to show the most valuable information, removing truly unique words, while at the same time throwing out those that are constantly being repeated. I set the n_gram to 1 - 2 words because this did manage to increase the feature space by 3 features, which were "north korea", "last year", and "seong ho". Ranges higher than this did not add to the feature space. I also chose to use a modified version of the NLTK stop word list, adding in a handful of words found in the dataset that gave no insight into what the speech was about, such as "also", "home", "this", and "tonight". ¶

To further process the data, I created a dictionary to replace several words to consolidate similar features, such as "United States" consolidating into "america", "years" into "year", and "americans" and "us" into "american." The effect of this is that it reduced the size of the feature space and showed how often the "americans" are mentioned. This makes sense because trump uses several terms to identify the same thing, so separating them into separate features would cause that information to be lost. This helps to answer the question because it shows that he talked about the american people very often in the speech. I chose not to use a stemmer, because many of the features were chopped up and the meaning was lost. This may have been useful in a larger corpus, but in this case it was not needed.

Q2. Write a short description of how the sentiment analysis was done and what the outcome is. Make sure your answer is no longer than three paragraphs, and should at minimum answer these questions: • How did your processing affect the sentiment assignment, if at all? • What measure did you use to determine the sentiment label? Why? Do any of the label assignments surprise you? • Include a few specific examples of label assignment and how it was determined and why it does or does not make sense.

Audience: general – management or non-technical staff.

A2. The sentiment analysis was done by using the a_finn sentiment dictionary. The outcome, for the most part, seemed to be accurate. I also used the *and* sentiment dictionary on this corpus, but the a_finn dictionary appeared to be the most accurate.

Processing didn't seem to make much of a difference on how the sentiment labels were assigned., with the exception of row 29 (index 28). The processing actually changed the sentiment from "Negative" to "Positive," which is how it should appear, which I thought was surprising. Looking at the data, I would assume that the replacement of "us" with "american" had an effect on this. The sentiment label was determined by finding words listed in the a_finn dictionary and adding the associated value to the sentiment count, with positive words adding to the count, negative words subtracting from the count. If the count was greater than zero, the sentiment was "Positive", less than zero was "Negative", and equal to zero is "Neutral". The value is based on the calculated sentiment intensity of that word. For example: "good" may have a value of 1, "great" may have a value of 2, and "best" may have a value of 3.

We can see how this works in index 4: "each test has forged new american heroes to remind american who we are, and show american what we can be." This was "Positive" because the sentiment dictionary includes the word "heroes", which raised the sentiment count by a value 2. Another example of this can be seen in index 6: "we saw strangers shielding strangers from a hail of gunfire on the las vegas strip." The sentiment dictionary found "hail", with a value of 2 and "gun" with a value of -1. This resulted in a "Positive" sentiment. This is surprising because it actually came up with the correct sentiment. I would have expected the sentiment label to be incorrect because "hail of gunfire" would normally be a very negative thing, but this dictionary has "hail" meaning praise, rather than something that would shower from the sky.

Q3. Write a short description of the exercise and the outcome. Make sure your answer is no longer than three paragraphs, and should at minimum answer these questions: • How did you change the process to fix that outcome? • How would you explain (justify, rationalize) those changes if necessary?

Audience: general – management or non-technical staff.

A3. I found that the sentiment label in index 8 was incorrect, showing "Neutral" rather than "Positive":

"we heard about american like firefighter david dahlberg. he is here with american too. david faced down walls of flame to rescue almost 60 children trapped at a california summer camp threatened by wildfires."

In order to resolve this, I added an entry into the sentiment dictionary for the word "rescue" with a highly positive value of 3. This brought the sentiment count over 0, so it is now "Positive." I would justify this by saying no other entries in the the data are affected, since rescue only appeared two other times and each of the documents it appeared in already had a sentiment of "Positive."

Q4. Data science is all about analyzing data to draw a conclusion. You have just made a change to your analysis to match a conclusion about a label that you already made. Write a short description defending your actions. Is what you did right? What are the ethical issues involved, if any? What is your role as a data scientist? (Max 4 points)

Audience: general – management or non-technical staff.

A4. There are 3 times that "rescue" appears, with every other document receiving a "Positive" label. Entering this into the dictionary only changed the outcome for this particular document, so I feel that the change would not be an issue of skewing the analysis. I believe what I did here was right because the impact of the change I made was very minimal. I could see a change like this being unethical if a data scientist made changes that drastically altered the outcome of the analysis or if they altered the analysis to intentionally sway the outcome. However, the change I made corrected an obvious mistake in the sentiment analysis, so I would not consider it wrong or unethical.

The role of a data scientist is to honestly analyze data, build a predictive model, and present the information in a clear and consise manner. In my opinion, it should also be a priority for data scientists to change the data as little as possible. Changes should be made to clean and clarify and data, but they should not be made to change or skew the message that the data is telling.

T1. Read in or create a data frame with at least one column of text to be analyzed. This could be the text you used previously or new text. Based on the context of your dataset and the question you want to answer, identify at what processing you think is necessary (stop words, stemming, custom replacement, etc.) Compare the feature space before and after your processing.

In [1]:
```
# import module(s) into namespace
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup
import re
import requests
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
import math
from __future__ import division
from nltk.corpus import stopwords
from sklearn.feature_extraction import text
pd.set_option('display.max_colwidth', 150)
```

Pulling in a csv file containing the text of Trump's State of the Union Address.

In [2]:
```
pathname = "/Users/ca034330/Google Drive/Corey - School/Spring 2018 A/BIA 6304
 - Text Mining/HW_2/"
pd.set_option('display.max_colwidth', 15000)
```

In [3]:
```
speechdf = pd.read_csv(pathname + "trump_speech.csv", index_col = 0)
print(speechdf.shape)
print(list(speechdf)) #what does this code do?
```

```
(149, 1)
['text']
```

Initiating the prebuilt stop word list and creating a custom stopword list.

In [4]:
```
#Creating the nltk stopword list and adding terms to it that appear too often
 and add no meaning.
nltk_stopwords = stopwords.words("english")
skl_stopwords = text.ENGLISH_STOP_WORDS

custom_nltk = nltk_stopwords + ["000", "mr", "said", "says","say", "ms", "als
o", "has", "this", "one", "home", "every", "tonight" ]

#nltk_stopwords.remove('before') #In case I need to remove words
```

Checking the size of the feature space before word replacements and stemming.

In [5]:
```
cv = CountVectorizer(binary=False, min_df = .03, max_df = .5,stop_words = cust
om_nltk, ngram_range = (1,2))
cv_speech = cv.fit_transform(speechdf['text'])
print(cv_speech.shape)

names = cv.get_feature_names()
count = np.sum(cv_speech.toarray(), axis = 0)
count2 = count.tolist()
count_df = pd.DataFrame(count2, index = names, columns = ['count'])
count_df.sort_values(['count'], ascending = False)[0:10]
```

```
(149, 75)
```

Out[5]:

	count
american	31
america	27
people	26
us	24
americans	24
new	21
year	19
tax	15
country	15
last	13

Create a dictionary to replace words in the corpus.

In [6]:
```
#Creating a dictionary to replace words in the corpus in order to combine like
 terms.
speech_dict = {'united states':'america', 'americans': 'american', 'years':'ye
ar', "us":"american"}


def multiple_replace(dict, text):

  text = str(text).lower()

  # Create a regular expression  from the dictionary keys
  regex = re.compile("(%s)" % "|".join(map(re.escape, dict.keys())))

  # For each match, look-up corresponding value in dictionary
  return regex.sub(lambda mo: dict[mo.string[mo.start():mo.end()]], text)
```

In [7]:
```
#use the dictionary to replace words in the corpus

speechdf['cleantext'] = speechdf.text.apply(lambda x: multiple_replace(speech_
dict, x))
speechdf[0:1]
```

Out[7]:

	text	cleantext
0	Mr. Speaker, Mr. Vice President, members of Congress, the first lady of the United States, and my fellow Americans:	mr. speaker, mr. vice president, members of congress, the first lady of the america, and my fellow american:

Using the Porter Stemmer

In [8]:
```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

def stem_text(row):
    text = str(row).split()
    stemtext = [ps.stem(word) for word in text]
    stem2text = ' '.join(stemtext)
    return stem2text

speechdf['stemmed'] = speechdf["cleantext"].apply(lambda x: stem_text(x))
print( speechdf.text[0:1])
print("~~~~~~~~~~~~~~~~~~~")
print(speechdf.stemmed[0:1])
```

```
0    Mr. Speaker, Mr. Vice President, members of Congress, the first lady of
the United States, and my fellow Americans:
Name: text, dtype: object
~~~~~~~~~~~~~~~~~~~
0    mr. speaker, mr. vice president, member of congress, the first ladi of t
he america, and my fellow american:
Name: stemmed, dtype: object
```

Using the count vectorizer

In [9]:
```
cv = CountVectorizer(binary=False, min_df = .03, max_df = .5,stop_words = cust
om_nltk, ngram_range = (1,2))
cv_speech = cv.fit_transform(speechdf['cleantext'])
print(cv_speech.shape)

names = cv.get_feature_names()
count = np.sum(cv_speech.toarray(), axis = 0)
count2 = count.tolist()
count_df = pd.DataFrame(count2, index = names, columns = ['count'])
count_df.sort_values(['count'], ascending = False)[0:10]
```

```
(149, 71)
```

Out[9]:

	count
american	79
america	33
year	29
people	26
new	21
tax	15
country	15
last	13
congress	13
great	13

In [10]:
```
#The stemmer in this case did not clear up the feature space, it only increase
d the size and added terms that make no sense.
cv = CountVectorizer(binary=False, min_df = .03, max_df = .5,stop_words = cust
om_nltk, ngram_range = (1,3))
cv_speech = cv.fit_transform(speechdf['stemmed'])
print(cv_speech.shape)

names = cv.get_feature_names()
count = np.sum(cv_speech.toarray(), axis = 0)
count2 = count.tolist()
count_df = pd.DataFrame(count2, index = names, columns = ['count'])
count_df.sort_values(['count'], ascending = False)[0:10]
```

```
(149, 97)
```

Out[10]:

	count
american	80
thi	37
america	33
year	29
new	21
peopl	21
ha	20
hi	19
wa	16
tax	15

Not using a stemmer is the preferred method here.

T2. Create a sentiment dictionary from one of the sources in class or find/create your own (potential bonus points for appropriate creativity). Using your dictionary, create sentiment labels for the text entries in your corpus.

Load in the sentiment dictionaries.

In [11]:
```
pathname = "/Users/ca034330/Google Drive/Corey - School/Spring 2018 A/BIA 6304
 - Text Mining/HW_2/Dictionaries/"
```

Loading up the afinn sentiment dictionary.

In [12]:
```
afinn = {}
for line in open(pathname+"AFINN-111.txt"):
    tt = line.split('\t')
    afinn.update({tt[0]:int(tt[1])})
```

In [13]:
```
def afinn_sent(inputstring):

    sentcount =0
    for word in inputstring.split():
        if word.rstrip('?:!.,;') in afinn:
            sentcount = sentcount + afinn[word.rstrip('?:!.,;')]


    if (sentcount < 0):
        sentiment = 'Negative'
    elif (sentcount >0):
        sentiment = 'Positive'
    else:
        sentiment = 'Neutral'

    return sentiment
    #return sentcount
```

In [27]:
```
speechdf['afinn'] = speechdf["text"].apply(lambda x: afinn_sent(x))
```

In [28]:
```
speechdf['afinn_clean'] = speechdf["cleantext"].apply(lambda x: afinn_sent(x))
```

In [34]:
```
speechdf.iloc[0:10][['cleantext','afinn','afinn_clean']]
```

Out[34]:

	cleantext	afinn	afinn_clean
0	mr. speaker, mr. vice president, members of congress, the first lady of the america, and my fellow american:	Neutral	Neutral
1	less than one year has passed since i first stood at this podium, in this majestic chamber, to speak on behalf of the american people — and to address their concerns, their hopes and their dreams. that night, our new administration had already taken swift action. a new tide of optimism was already sweeping across our land.	Positive	Positive
2	each day since, we have gone forward with a clear vision and a righteoamerican mission — to make america great again for all american.	Positive	Positive
3	over the last year, we have made incredible progress and achieved extraordinary success. we have faced challenges we expected, and others we could never have imagined. we have shared in the heights of victory and the pains of hardship. we endured floods and fires and storms. but through it all, we have seen the beauty of america's soul, and the steel in america's spine.	Positive	Positive
4	each test has forged new american heroes to remind american who we are, and show american what we can be.	Positive	Positive
5	we saw the volunteers of the "cajun navy," racing to the rescue with their fishing boats to save people in the aftermath of a devastating hurricane.	Positive	Positive
6	we saw strangers shielding strangers from a hail of gunfire on the las vegas strip.	Positive	Positive
7	we heard tales of american like coast guard petty officer ashlee leppert, who is here tonight in the gallery with melania. ashlee was aboard one of the first helicopters on the scene in hoamericanton during hurricane harvey. through 18 hours of wind and rain, ashlee braved live power lines and deep water to help save more than 40 lives. thank you, ashlee.	Positive	Positive
8	we heard about american like firefighter david dahlberg. he is here with american too. david faced down walls of flame to rescue almost 60 children trapped at a california summer camp threatened by wildfires.	Neutral	Neutral
9	to everyone still recovering in texas, florida, louisiana, puerto rico, the virgin islands, california and everywhere else — we are with you, we love you, and we will pull through together.	Positive	Positive

In [33]:
```
# Cleaning the data changed row 28 from negative to positive. 'American' replaced 'us,' so maybe that had an effect.
speechdf.iloc[28:29][['text','cleantext']]
```

Out[33]:

	text	cleantext
28	One of Staub's employees, Corey Adams, is also with us tonight. Corey is an all-American worker. He supported himself through high school, lost his job during the 2008 recession and was later hired by Staub, where he trained to become a welder. Like many hardworking Americans, Corey plans to invest his tax?cut raise into his new home and his two daughters' education. Please join me in congratulating Corey.	one of staub's employees, corey adams, is also with american tonight. corey is an all-american worker. he supported himself through high school, lost his job during the 2008 recession and was later hired by staub, where he trained to become a welder. like many hardworking american, corey plans to invest his tax?cut raise into his new home and his two daughters' education. please join me in congratulating corey.

T3. Consider one of the entries in your corpus that had a surprising label. How would you change your analysis to get the “right” label? Show specific results.

In [49]:
```
# Line 9 (Index 8) should be "positive," not "neutral."
speechdf.iloc[8:9][['cleantext',"afinn"]]
```

Out[49]:

	cleantext	afinn
8	we heard about american like firefighter david dahlberg. he is here with american too. david faced down walls of flame to rescue almost 60 children trapped at a california summer camp threatened by wildfires.	Neutral

In [17]:
```
#Creating a new sentiment dictionary.
new_afinn = {}
for line in open(pathname+"AFINN-111.txt"):
    tt = line.split('\t')
    new_afinn.update({tt[0]:int(tt[1])})
```

In [22]:
```
# Adding a new term to the sentiment dictionary
new_afinn['rescue'] =3
```

In [19]:
```
def new_afinn_sent(inputstring):

    sentcount =0
    for word in inputstring.split():
        if word.rstrip('?:!.,;') in new_afinn:
            sentcount = sentcount + new_afinn[word.rstrip('?:!.,;')]

    if (sentcount < 0):
        sentiment = 'Negative'
    elif (sentcount >0):
        sentiment = 'Positive'
    else:
        sentiment = 'Neutral'

    return sentiment
```

In [23]:
```
speechdf['modified_afinn'] = speechdf["cleantext"].apply(lambda x: new_afinn_s
ent(x))
```

In [44]:
```
#Showing that the change to the dictionary changed the sentiment label.
speechdf.iloc[8:9][['cleantext','afinn','modified_afinn']]
```

Out[44]:

	cleantext	afinn	modified_afinn
8	we heard about american like firefighter david dahlberg. he is here with american too. david faced down walls of flame to rescue almost 60 children trapped at a california summer camp threatened by wildfires.	Neutral	Positive

In [46]:
```
#Showing that the change to the dictionary had no effect on the sentiment labe
l.
speechdf.iloc[5:6][['cleantext','afinn','modified_afinn']]
```

Out[46]:

	cleantext	afinn	modified_afinn
5	we saw the volunteers of the "cajun navy," racing to the rescue with their fishing boats to save people in the aftermath of a devastating hurricane.	Positive	Positive

In [48]:
```
#Showing that the change to the dictionary had no effect on the sentiment labe
l.
speechdf.iloc[129:130][['cleantext','afinn','modified_afinn']]
```

Out[48]:

	cleantext	afinn	modified_afinn
129	today he lives in seoul, where he rescues other defectors, and broadcasts into north korea what the regime fears the most — the truth.	Positive	Positive