Zack Edwards

Professor Raz Saremi

EM-224A Informatics and Software Development

5 May, 2020

Exercise 10

Analysis

This assignment asked us to comb through a csv file containing the emails which Hillary Clinton sent from a private email server. The first step was to extract information about the senders. This information was already isolated in its own column. By grabbing this information and cleaning it, I was able to determine the top 15 correspondents. Since one of these emails was Hillary herself, I collected one for more email for a total of 16 senders. I then graphed the frequency of correspondence for the 15 emails excluding Hillary. This graph shows us the steep drop off in frequency after the first 4 correspondents indicating a high frequency of communication with key operators. The extraction of the top correspondents shows us that the emails used by the top senders all are using @state.gov emails which indicates that they are state employees.

The second half of the task was to isolate and prepare the raw text for analysis. By putting these emails through this program I am able to obtain a word cloud of the most frequently used words. This word cloud can help give us an insight into the general topics being frequently discussed in the raw text and the overall sentiment. Some words which

pop out to me from the word cloud are ‘Benghazi”, “Unclassified”, “state department”, “redactions”, “Pakistan”, “Isreal”, “FOIA”, and many more. These words are all highly correlated with government business and meant to only be available to elected goverment officials with access. The reason Hillary got in trouble was for using a private email server which was more susceptible for being hacked and also it was less recorded. This was a very shady and highly illegal way for a politician to conduct state affairs.

Attachments

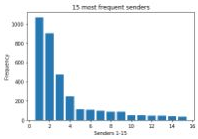
Code input for finding the top 15 senders:

```
top_15_senders = (removeDuplicates(sender_and_frequency)) #removing duplicate emails
top_15_senders = sorted(top_15_senders, key=getKey, reverse=True) #sorting by frequency


counter = 0
top_15_names = []
name = ''
for i in top_15_senders: #this loop finds the names for each sender
    for x in senders_list:
        if i[0] in x:
            L = len(i[0])
            name = x[:-len(i[0])]
        if name == 'H':
            name = 'Hillary Clinton'
    top_15_names.append(name)
    counter += 1
    if counter < 16:
        continue
    else:
        break


print('\nThese are the top 15 most frequent senders from the Clinton emails:')
counter = 0
for i in top_15_senders:
    print('Sender #', counter+1)
    print('Name: ', top_15_names[counter], '\nEmail: ', i[0], '\nFrequency: ', i[1], '\n')
    counter += 1
    if counter < 16:
        continue
    else:
        break
print('hrod17@clintonemail.com is the only email which in not an @state.gov, it is the ')
print('private email server which hillary infamously used to send classified information.')
print('Though the assignment was to print the top 15 correspondants I printed 16 since one was Hillary herself')
print('\nThis is a graph of the frequency of correspondance with the 15 most common senders (excluding Hillary)')
top_15_frequencies=[]
counter = 0
for i in top_15_senders:
    top_15_frequencies.append(i[1])
    counter += 1
    if counter < 16:
        continue
    else:
        break
#print(top_15_frequencies)
y= top_15_frequencies[1:]
x = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
```

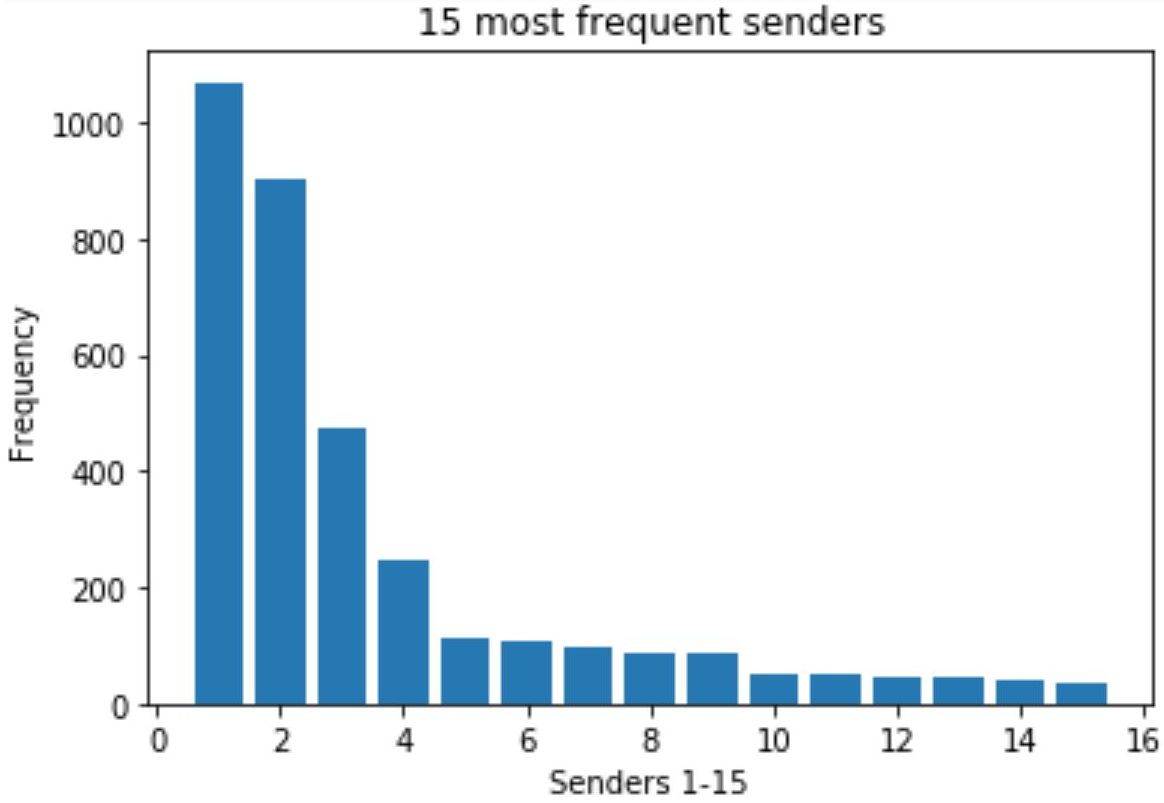
```
top_15_names = []
name = ''
for i in top_15_senders: #this loop finds the names for each sender
    for x in senders_list:
        if i[0] in x:
            L = len(i[0])
            name = x[:-len(i[0])]
        if name == 'H':
            name = 'Hillary Clinton'
    top_15_names.append(name)
    counter += 1
    if counter < 16:
        continue
    else:
        break


print('\nThese are the top 15 most frequent senders from the Clinton emails:')
counter = 0
for i in top_15_senders:
    print('Sender #', counter+1)
    print('Name: ', top_15_names[counter], '\nEmail: ', i[0], '\nFrequency: ', i[1], '\n')
    counter += 1
    if counter < 16:
        continue
    else:
        break
print('hrod17@clintonemail.com is the only email which in not an @state.gov, it is the ')
print('private email server which hillary infamously used to send classified information.')
print('Though the assignment was to print the top 15 correspondants I printed 16 since one was Hillary herself')
print('\nThis is a graph of the frequency of correspondance with the 15 most common senders (excluding Hillary)')
top_15_frequencies=[]
counter = 0
for i in top_15_senders:
    top_15_frequencies.append(i[1])
    counter += 1
    if counter < 16:
        continue
    else:
        break
#print(top_15_frequencies)
y= top_15_frequencies[1:]
x = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
plt.bar(x=x,height=y,data=y)
plt.xlabel('Senders 1-15')
plt.ylabel('Frequency')
plt.title('15 most frequent senders')
plt.show()
```

Code output for finding the top 15 senders:

```
Email:  <VermaRR@state.gov>
Frequency:  89

Sender # 10
Name:  Mills, Cheryl D
Email:  <MillsCD©state.gov>
Frequency:  87

Sender # 11
Name:  McHale, Judith A
Email:  <McHaleJA@state.gov>
Frequency:  53

Sender # 12
Name:  Sullivan, Jacob J
Email:  <SullivanJJ©state.gov>
Frequency:  50

Sender # 13
Name:  Sullivan, Jacob J
Email:  <Sullivann@state.gov>
Frequency:  47

Sender # 14
Name:  Verveer, Melanne S
Email:  <VerveerMS@state.gov>
Frequency:  46

Sender # 15
Name:  Muscatine, Lissa
Email:  <MuscatineL@state.gov>
Frequency:  41

Sender # 16
Name:  Jiloty, Lauren C
Email:  <JilotyLC©state.gov>
Frequency:  36

hrod17@clintonemail.com is the only email which in not an @state.gov, it is the
private email server which hillary infamously used to send classified information.
Though the assignment was to print the top 15 correspondants I printed 16 since one was Hillary herself

This is a graph of the frequency of correspondance with the 15 most common senders (excluding Hillary)
```



Graph output for the frequency of correspondence for the top 15 senders:



Code input for generating the word cloud:

```
from wordcloud import WordCloud, STOPWORDS
raw_list=[]
raw_text = data['RawText']
temp = [] #creating an empty temporary list
for i in raw_text: #a loop which eliminated 0's which occur from empty cells
    i = str(i) #interprettng each cell as a string
    i = i.strip().split()
    for x in i:
        temp.append(x)
raw_list = temp
temp_list = []

#this line removes punctuation
raw_list = [''.join(c for c in s if c not in string.punctuation) for s in raw_list]

stop = open('stopwords_en.txt', 'r')
stops = [] #initializing and cleaning the list of stopwords
stop_list = stop.readlines()
for i in stop_list:
    stops.append(i[:-1])
stop_list = stops

for i in raw_list: #This loops eliminates stop words from the data
    for x in i.split():
        if x not in stops:
            temp_list.append(x)
raw_list = temp_list
temp_list = []

for i in raw_list:
    if i.isdigit() == False:
        temp_list.append(i)
raw_list = temp_list #our cleaned list fo words

# Transforming the list into a string for displaying
text_str = ' '.join(raw_list)

# Crating and updating the stopword list
stpwords = set(STOPWORDS)
stpwords.add('will')
stpwords.add('said')


# Defining the wordcloud parameters
wc = WordCloud(background_color="white", max_words=2000, width=1000, height=1000,
               stopwords=stpwords)

# Generate word cloud
wc.generate(text_str)
```

```
for i in raw_list: #This loops eliminates stop words from the data
    for x in i.split():
        if x not in stops:
            temp_list.append(x)
raw_list = temp_list
temp_list = []

for i in raw_list:
    if i.isdigit() == False:
        temp_list.append(i)
raw_list = temp_list #our cleaned list fo words

# Transforming the list into a string for displaying
text_str = ' '.join(raw_list)

# Crating and updating the stopword list
stpwords = set(STOPWORDS)
stpwords.add('will')
stpwords.add('said')


# Defining the wordcloud parameters
wc = WordCloud(background_color="white", max_words=2000, width=1000, height=1000,
               stopwords=stpwords)

# Generate word cloud
wc.generate(text_str)

# Store to file
wc.to_file('hillaryemailwordcloud.png')


print('\nThis is the wordcloud for the content of Hillarys emails')
# Show the cloud
plt.imshow(wc)
plt.axis('off')
plt.show()

print('done')
```

```
This is the wordcloud for the content of Hillarys emails
```


```
done
```

The word cloud: