In [ ]: from google.colab import drive #to mount gdrive on colab
drive.mount('/content/drive')

Mounted at /content/drive

In [ ]: import warnings
warnings.filterwarnings("ignore")

In [ ]: import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
np.set_printoptions(threshold=sys.maxsize)
# Load the tokenizer
tokenizer = CountVectorizer(lowercase=True, ngram_range=(1,1))
fitlm = gzip.open(path + "/fit.txt")

In [ ]: df_train = df.head(400)
# Fit the tokenizer on the training data (in 2013,2014,2015,2016 and 2018 together)
# df_train = df.sample(400)
X_train = tokenizer.fit(df_train['phrase'].values)
df_train[:5]

	Unnamed: 0	id	sentiment_id	created_at	date	between	place	text	tokens
0	0	1004677160	1467811372	Mon Apr 06 22:19:45 PDT 2009	2009-04-06 22:19:45	1467811372	[]	@Tiffany_Trump I miss you in the morning	['@tiffany_trump', 'miss', 'you', 'morning']
1	1	1562808760	1835467128	Wed Jun 03 04:12:41 PDT 2009	2009-06-03 04:12:41	1835467128	[]	@Tiffany_Trump Tiffany Trump	['@tiffany_trump', 'tiffany', 'trump']
2	2	1598531260	1842884966	Mon Jun 08 00:12:03 PDT 2009	2009-06-08 00:12:03	1842884966	[]	@Tiffany_Trump Tiffany Trump	['@tiffany_trump', 'tiffany', 'trump']
3	3	1641284538	1927450460	Wed Jun 17 18:08:03 PDT 2009	2009-06-17 18:08:03	1927450460	[]	@Tiffany_Trump Tiffany Trump	['@tiffany_trump', 'tiffany', 'trump']
4	4	1693587651	2099822491	Fri Jun 26 03:26:38 PDT 2009	2009-06-26 03:26:38	2099822491	[]	@Tiffany_Trump I love @Tiffany_Trump	['@tiffany_trump', 'love', '@tiffany_trump']
395	395	1757213560	2034572170	Wed Jun 10 18:09:40 PDT 2009	2009-06-10 18:09:40	2034572170	[]	@Tiffany_Trump @Tiffany_Trump Tiffany Trump	['@tiffany_trump', '@tiffany_trump', 'tiffany', 'trump']
396	396	1757213560	2034572170	Wed Jun 10 18:09:40 PDT 2009	2009-06-10 18:09:40	2034572170	[]	@Tiffany_Trump @Tiffany_Trump Tiffany Trump	['@tiffany_trump', '@tiffany_trump', 'tiffany', 'trump']
397	397	1757213560	2034572170	Wed Jun 10 18:09:40 PDT 2009	2009-06-10 18:09:40	2034572170	[]	@Tiffany_Trump @Tiffany_Trump Tiffany Trump	['@tiffany_trump', '@tiffany_trump', 'tiffany', 'trump']
398	398	1757213560	2034572170	Wed Jun 10 18:09:40 PDT 2009	2009-06-10 18:09:40	2034572170	[]	@Tiffany_Trump @Tiffany_Trump Tiffany Trump	['@tiffany_trump', '@tiffany_trump', 'tiffany', 'trump']
399	399	1757213560	2034572170	Wed Jun 10 18:09:40 PDT 2009	2009-06-10 18:09:40	2034572170	[]	@Tiffany_Trump @Tiffany_Trump Tiffany Trump	['@tiffany_trump', '@tiffany_trump', 'tiffany', 'trump']

DOTS Code = 64 columns

In [ ]: import matplotlib
%matplotlib inline # to get plots inline in the data f
ramework
import sys
print("Python3 Tag 2 of 2\nsys['base']")
print("\nsys[0]",sys[0])

In [ ]: df_train[:200] = df_train[df_train['date'] == '2017'] #to create data frame of the tweets belong
ing to year 2017
df_train[:200] = df_train[df_train['date'] == '2016'] #to create data frame of the tweets belong
ing to year 2016
df_train[:200] = df_train[df_train['date'] == '2015'] #to create data frame of the tweets belong
ing to year 2015
df_train[:200] = df_train[df_train['date'] == '2014'] #to create data frame of the tweets belong
ing to year 2014
df_train[:200] = df_train[df_train['date'] == '2013'] #to create data frame of the tweets belong
ing to year 2013
df_train[:200] = df_train[df_train['date'] == '2012'] #to create data frame of the tweets belong
ing to year 2012

In [ ]: # Count phrases to combine bi tokens in a given year
# 2017
for name in df_train:
print(df_train[name])

In [ ]: tweet_2017 = tweet_2017[tweet_2017['date'] == '2017']
tweet_2016 = tweet_2016[tweet_2016['date'] == '2016']
tweet_2015 = tweet_2015[tweet_2015['date'] == '2015']
tweet_2014 = tweet_2014[tweet_2014['date'] == '2014']
tweet_2013 = tweet_2013[tweet_2013['date'] == '2013']
tweet_2012 = tweet_2012[tweet_2012['date'] == '2012']

In [ ]: import nltk
from nltk import FreqDist
nltk.download('stopwords')
from nltk.corpus import stopwords
# Stopwords
stopwords = set(stopwords.words('english'))
print(len(stopwords), '', stopwords[:5], '', stopwords[-5:])
words = lambda x: x.split()
sw = lambda x: sum([1 for w in words(x) if w in stopwords])
df_train['sw'] = df_train.text.apply(sw)
print('Stopwords: ', df_train[['text', 'sw']].head(2))
print('\n\n')
# Stopwords
STOP_DATA1 = set(stopwords.words('english'))
print('\n\n')
# Extending stopwords
STOP_DATA1 |= set(stopwords.words('spanish'))
STOP_DATA1 |= set(stopwords.words('portuguese'))
STOP_DATA1 |= set(stopwords.words('russian'))
STOP_DATA1 |= set(stopwords.words('turkish'))
STOP_DATA1 |= set(stopwords.words('romanian'))
STOP_DATA1 |= set(stopwords.ru)

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

Word Frequencies for Each Year

In [ ]: tweet_2017_FreqDist = nltk.FreqDist(word_filter(tweet_2017)) #to find frequency of words in a gi
ven year
tweet_2016_FreqDist = nltk.FreqDist(word_filter(tweet_2016)) #to find frequency of words in a gi
ven year
tweet_2015_FreqDist = nltk.FreqDist(word_filter(tweet_2015)) #to find frequency of words in a gi
ven year
tweet_2014_FreqDist = nltk.FreqDist(word_filter(tweet_2014)) #to find frequency of words in a gi
ven year
tweet_2013_FreqDist = nltk.FreqDist(word_filter(tweet_2013)) #to find frequency of words in a gi
ven year
tweet_2012_FreqDist = nltk.FreqDist(word_filter(tweet_2012)) #to find frequency of words in a gi
ven year

In [ ]: tweet_2017_FreqDist.tabulate() #word frequencies for 2017

amp	trump	can	you	people	donald	like	trump's	president	obama	new	one
1338	1320	1270	1230	1220	1180	1170	1160	1150	1140	1130	1120

In [ ]: tweet_2016_FreqDist.tabulate() #word frequencies for 2016

amp	trump	donald	trump's	president	obama	people	like	can	you
1380	1360	1340	1320	1300	1280	1260	1240	1220	1200

In [ ]: tweet_2015_FreqDist.tabulate() #word frequencies for 2015

amp	trump	donald	trump's	president	obama	people	like	can	you
1420	1400	1380	1360	1340	1320	1300	1280	1260	1240

In [ ]: tweet_2014_FreqDist.tabulate() #word frequencies for 2014

trump	amp	donald	trump's	president	obama	people	like	can	one
1460	1440	1420	1400	1380	1360	1340	1320	1300	1280

In [ ]: tweet_2013_FreqDist.tabulate() #word frequencies for 2013

amp	trump	donald	trump's	president	obama	people	like	can	one
1500	1480	1460	1440	1420	1400	1380	1360	1340	1320

In [ ]: tweet_2012_FreqDist.tabulate() #word frequencies for 2012

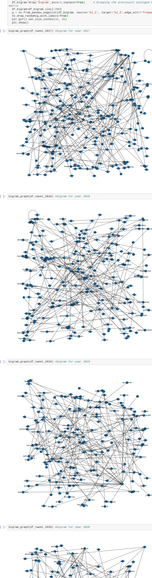
trump	amp	donald	trump's	president	obama	people	like	can	one
1540	1520	1500	1480	1460	1440	1420	1400	1380	1360

5. Top 10 words (for each year) by the highest value of Word Frequency

In [ ]: import matplotlib.pyplot as plt
import numpy as np
# Plot the word frequency distribution for plot frequency plot
tweet_2017_FreqDist.plot(10,cumulative=False,title="2017")
tweet_2016_FreqDist.plot(10,cumulative=False,title="2016")
tweet_2015_FreqDist.plot(10,cumulative=False,title="2015")
tweet_2014_FreqDist.plot(10,cumulative=False,title="2014")
tweet_2013_FreqDist.plot(10,cumulative=False,title="2013")
tweet_2012_FreqDist.plot(10,cumulative=False,title="2012")
plt.figure(figsize=(15,15))
plt.subplot(1,1,1)
plt.subplots_2(1,2)
plt.subplots(1,3)
plt.subplots(1,4)
plt.subplots(1,5)
plt.subplots(1,6)
plt.subplots(1,7)
plt.subplots(1,8)
plt.subplots(1,9)
plt.subplots(1,10)
plt.subplots(1,11)
plt.subplots(1,12)
plt.show()

In [ ]: %matplotlib inline
Plot the word frequency distribution for plot frequency plot
tweet_2017_FreqDist.plot(10,cumulative=False,title="2017") #frequency Plot for 2017 using frequency plot function
plt.subplots(1,2)
tweet_2016_FreqDist.plot(10,cumulative=False,title="2016") #frequency Plot for 2016 using frequency plot function
plt.subplots(1,3)
tweet_2015_FreqDist.plot(10,cumulative=False,title="2015") #frequency Plot for 2015 using frequency plot function
plt.subplots(1,4)
tweet_2014_FreqDist.plot(10,cumulative=False,title="2014") #frequency Plot for 2014 using frequency plot function
plt.subplots(1,5)
tweet_2013_FreqDist.plot(10,cumulative=False,title="2013") #frequency Plot for 2013 using frequency plot function
plt.subplots(1,6)
tweet_2012_FreqDist.plot(10,cumulative=False,title="2012") #frequency Plot for 2012 using frequency plot function
plt.show()

Out[ ]: Text(0.0, 1.0, '2012')



6. Histogram of Word Frequencies for Each Year

In [ ]: import matplotlib as mpl
# Plot the histogram
# Use matplotlib.pyplot.hist() to plot frequency plot
tweet_2017_FreqDist.plot(10,cumulative=False,title="2017")
tweet_2016_FreqDist.plot(10,cumulative=False,title="2016")
tweet_2015_FreqDist.plot(10,cumulative=False,title="2015")
tweet_2014_FreqDist.plot(10,cumulative=False,title="2014")
tweet_2013_FreqDist.plot(10,cumulative=False,title="2013")
tweet_2012_FreqDist.plot(10,cumulative=False,title="2012")
plt.figure(figsize=(15,15))
plt.subplots(1,1)
plt.subplots(1,2)
plt.subplots(1,3)
plt.subplots(1,4)
plt.subplots(1,5)
plt.subplots(1,6)
plt.subplots(1,7)
plt.subplots(1,8)
plt.subplots(1,9)
plt.subplots(1,10)
plt.subplots(1,11)
plt.subplots(1,12)
plt.show()

In [ ]: %matplotlib inline
Plot the histogram
Use matplotlib.pyplot.hist() to plot frequency plot
tweet_2017_FreqDist.plot(10,cumulative=False,title="2017") #frequency Plot for 2017 using frequency plot function
plt.subplots(1,2)
tweet_2016_FreqDist.plot(10,cumulative=False,title="2016") #frequency Plot for 2016 using frequency plot function
plt.subplots(1,3)
tweet_2015_FreqDist.plot(10,cumulative=False,title="2015") #frequency Plot for 2015 using frequency plot function
plt.subplots(1,4)
tweet_2014_FreqDist.plot(10,cumulative=False,title="2014") #frequency Plot for 2014 using frequency plot function
plt.subplots(1,5)
tweet_2013_FreqDist.plot(10,cumulative=False,title="2013") #frequency Plot for 2013 using frequency plot function
plt.subplots(1,6)
tweet_2012_FreqDist.plot(10,cumulative=False,title="2012") #frequency Plot for 2012 using frequency plot function
plt.show()

Out[ ]: Text(0.0, 1.0, '2012')

In [ ]: pip install networkx[all,3.2] #to install networkx library version 2.2
import sys
sys.stdout.write('https\://pypi.org/simple, https\://ca-geohub.pkg.dev/ca-geohub-vdev/simple/n
ewest\n')
Requirement already satisfied: networkx[all,3.2] in /usr/local/lib/python3.10/dist-packages (3.4.2)
WARNING: networkx[all,3.2] does not provide the extra 'all'
Collecting numpy>=1.24
Collecting scipy>=1.10,!=1.11.0,!=1.11.1
Collecting matplotlib>=3.6
Collecting pandas>=2.0
Collecting pyyaml>=5.3
Collecting lxml>=4.6
Collecting pygraphviz>=1.14
Collecting pydot>=1.4.1
Collecting sympy>=1.10
Collecting contourpy>=1.0.1
Collecting cycler>=0.10
Collecting fonttools>=4.22.0
Collecting kiwisolver>=1.3.1
Collecting packaging>=20.0
Collecting pillow>=8
Collecting pyparsing>=2.3.1
Collecting python-dateutil>=2.7
Collecting pytz>=2020.1
Collecting tzdata>=2022.7
Collecting mpmath>=0.19
Collecting six>=1.5