import csv

from textblob import TextBlob

import re

from nltk.corpus import stopwords

import matplotlib.pyplot as plt; plt.rcdefaults()

import numpy as np

import matplotlib.pyplot as plt

infile = '@ManUtd\_tweets.csv'

# Create the symbol/smily masks to be removed

emoticons\_str = r"""

(?:

[:=;] # Eyes

[oO\-]? # Nose (optional)

[D\)\]\(\]/\\OpP] # Mouth

)"""

# create the tags, mentions and other handle masks to be removed from the target text

regex\_str = [

emoticons\_str,

r'<[^>]+>', # HTML tags

r'(?:@[\w\_]+)', # @-mentions

r"(?:\#+[\w\_]+[\w\'\_\-]\*[\w\_]+)", # hash-tags

r'http[s]?://(?:[a-z]|[0-9]|[$-\_@.&amp;+]|[!\*\(\),]|(?:%[0-9a-f][0-9a-f]))+', # URLs

r'(?:(?:\d+,?)+(?:\.?\d+)?)', # numbers

r"(?:[a-z][a-z'\-\_]+[a-z])", # words with - and '

r'(?:[\w\_]+)', # other words

r'(?:\S)' # anything else

]

tokens\_re = re.compile(r'('+'|'.join(regex\_str)+')', re.VERBOSE | re.IGNORECASE)

emoticon\_re = re.compile(r'^'+emoticons\_str+'$', re.VERBOSE | re.IGNORECASE)

# Method for tokenization on the basis of regex\_str (object processed through re module)

def tokenize(s):

return tokens\_re.findall(s)

# Method for preprocessing, removes the smilyes, links, numbers, etc as per defined in the regex\_str and emoticon\_str

def preprocess(s, lowercase=False):

tokens = tokenize(s)

if lowercase:

tokens = [token if emoticon\_re.search(token) else token.lower() for token in tokens]

return tokens

negMsg=0

posMsg=0

neutMsg=0

def get\_tweet\_sentiment(tweet):

'''

Utility function to classify sentiment of passed tweet

using textblob's sentiment method

'''

# create TextBlob object of passed tweet text

analysis = TextBlob(tweet)

# set sentiment

if analysis.sentiment.polarity > 0:

return 'positive'

elif analysis.sentiment.polarity == 0:

return 'neutral'

else:

return 'negative'

with open(infile, 'r') as csvfile:

rows = csv.DictReader(csvfile)

for row in rows:

text = row['text']

#text = tweet['text']

print("ORIGNAL TWEET TEXT:")

print(text)

print(" ")

# Remove special symbols from the string & convert to lowercase

text2 = text.lower()

text2=re.sub('[^A-Za-z0-9]+', ' ', text2)

# Remove the numbers from the string

text2 = ''.join(i for i in text2 if not i.isdigit())

print("SPECIAL CHARACTERS REMOVED:")

print(text2)

print(" ")

# Remove single letter and two letter words

text2=' '.join( [w for w in text2.split() if len(w)>1] )

print("REMOVED WORDS UPTO 1 CHARACTERS:")

print(text2)

print(" ")

# Apply text pre-processing and tokenization

text3 = preprocess(text2)

print("PRE-PROCESSING/TOKENIZATION APPLIED (SMILY, URLs REMOVED):")

print(text3)

print(" ")

# Remove stopwords from the given text using nltk stopwords (English language) (Note:Download the nltk corpora/stopwords)

stop = set(stopwords.words('english'))

text4 = [i for i in text3 if i not in stop]

print("STOPWORDS REMOVED:")

print(text4);

print(" ");

# convert string list to string for sentiment analysis using textblob module

text4str=' '.join(map(str, text4))

sentStr=get\_tweet\_sentiment(text4str)

if sentStr=="positive":

posMsg=posMsg+1;

elif sentStr=="neutral":

neutMsg=neutMsg+1;

elif sentStr=="negative":

negMsg=negMsg+1;

print("PRINT SENTIMENT RESULT:");

print("Number of Negative Tweets: "+str(negMsg))

print("Number of Positive Tweets: "+str(posMsg))

print("Number of Neutral Tweets: "+str(neutMsg))

print(" ");

objects = ('Postive', 'Negative','Neutral')

y\_pos = np.arange(len(objects))

performance = [posMsg,negMsg,neutMsg]

plt.bar(y\_pos, performance, align='center', alpha=0.5)

plt.xticks(y\_pos, objects)

plt.ylabel('No of tweets ')

plt.title('Sentiment Result')

plt.show()