Twitter Sentiment Analysis

-Pradyumna Anil Kumar Kubear

0th Assignment

Task-Envirornment setup

Firstly we need to download a package called snscrape for that we make use of conda install snscrape but unfortunately it is not supported. So we have to do pip install which is not advisable as conda will get damaged so we need to create a virtual envirornment for that we have to click on envirornment button on anaconda and then we must use of pip install snscrape command

1st assignment (File Name scraper.py)

In this assignment we are creating a separate json file called tweets.json which has attributes like id,date,content,username,likes and retweets

the details is shown below

#import packages

import json

import snscrape.modules.twitter as sntwitter

#Search the term ovarian cancer

searchTermVariable="Ovarian Cancer"

#Maximum limit

MaximumTweets=2000

#Declaring array variable

tweetsArray=[]

#iterate in the list of tweet variable tweetVariable

for index,tweetVariable in enumerate(sntwitter.TwitterSearchScraper(f"{searchTermVariable} near:\"United States\"").get\_items()):

    if index>=MaximumTweets:#if the maximum tweets exceed

        break#come out of the for loop.

        #Else add the elements to the array

    tweetsArray.append({

        "id":tweetVariable.id,

        "date":str(tweetVariable.date),

        "content":tweetVariable.content,

        "username":tweetVariable.username,

        "likes":tweetVariable.likeCount,

        "retweets":tweetVariable.retweetCount

        })

#open the file tweets.json in write mode if the file does not exist create a new file and add the elements

with open("tweets.json","w",encoding="utf-8") as f:#serializes output and writes it to tweets.json file

    json.dump(tweetsArray,f,ensure\_ascii=False,indent=4) #for serialization and writing output on screen

Here I made use of search term called “Ovarian Cancer” and I made use of those tweets whose location is in USA.

Command which I made use to run is **python scraper.py**

2)

#importing

import sys as System

import json

'''

this function reads AFINN file and it will return

a dictionary of scores

'''

def load\_sentiments(sent\_file):

    scoresArray={}#initializing scores array

    for line in sent\_file: #do iteration for each and every line

        termVariable,scoreVariable=line.split("\t") #split the line on basis of space that is we are

        #Asking compiler to store the termVariable and scoreVariable the array of files which is being storing data on space basis

        scoresArray[termVariable]=int(scoreVariable) #convert the score variable to integer format

    return scoresArray

'''

The score tweet function takes a tweet(python dictionary)

and scores dictionary and will return a sentiment score

for the tweet.

'''

def score\_tweet(tweet,scoresArray):

    scoreVariable=0#initializing score variable to zero

    textVariable=tweet.get("text","") #getting the text from tweet

    wordsArray=textVariable.split()#splitting the text content into array or list

    for word in wordsArray:#from words array iterate word by word.

        word=word.lower()#convert the word to lower case

        if word in scoresArray:#if you find a word in the scores array.

            scoreVariable+=scoresArray[word]#increment the scores variable

    return scoreVariable#return this score variable.

'''

This function reads sentiment and tweet files and loads the sentiment scores and

scores of each tweet printing out the result

'''

def main():

    sentimentFileVariable=open(System.argv[1])#open sentiment file from system

    tweetFileVariable=open(System.argv[2])#open tweet File from system

    scoresArray=load\_sentiments(sentimentFileVariable)#passing sentimentFileVariable

    for line in tweetFileVariable:#In tweet file variable do the following.

        tweetVariable=json.loads(line)#load json file line by line

        scoreVariable=score\_tweet(tweetVariable, scoresArray)#then pass it to score method to get the score

        print(tweetVariable,scoreVariable)#Print the overall score.

if \_\_name\_\_=='\_\_main\_\_':

    main()

This program will compute the score of each input and display the score of each tweet along with the score.

Command python tweet\_sentiment.py AFINN-111.txt data.json

3)

In order to display the scorers of new term we write the following program

import sys

import json

#this function is used to create a dictionary of word sentiments

def readSentimentsFunction(sent\_file):

    scoresArray={}#declaring empty scores array

    for lineVariable in sent\_file: #iterate sent\_file

        termVariable,scoreVariable=lineVariable.split("\t")#seperate and create an array on basis of space

        scoresArray[termVariable]=int(scoreVariable)

    return scoresArray#return the scores array which is being updated.

#to calculate sentiment of tweet on individual words

def caluculateTweetSentimentFunction(tweetVariable,scoresArray):

    wordsArray=tweetVariable.split()

    #Split on basis of space

    sentimentVariable=sum(scoresArray.get(word,0) for word in wordsArray)

    #To calculate the sentiment score we make use of this  line

    #For each and every word calculate sentiment score

    return sentimentVariable

    #Return the sentiment variable

#to calculate new sentiments

#To calculate sentiment values for non-sentiment carriying terms

def caluculateNewSentimentsFunction(sent\_file,tweet\_file):

    newScoresArray={}

    #initializing scores Array and word count

    wordCount={}

    for lineVariable in tweet\_file:

        #In tweet file iterate line by line

        tweetVariable=json.loads(lineVariable).get("text","")# get the text from json file

        sentiment=caluculateTweetSentimentFunction(tweetVariable, scoresArray)#caluclate the tweet sentiment

        wordsArray=tweetVariable.split()

        for word in wordsArray:#iterate words array word by word.

            if word not in scoresArray:# if the word is not present

                newScoresArray[word]=newScoresArray.get(word,0)+sentiment

                wordCount[word]=wordCount.get(word,0)+1

    for word,scoreVariable in newScoresArray.items():

        wordVariable=str(word)#decode word from bytes to string

        if wordCount[wordVariable]!=0:

            print(wordVariable,scoreVariable/wordCount[wordVariable])# to display count of each word.

if \_\_name\_\_=='\_\_main\_\_':

    sentFileVariable=open(sys.argv[1])

    tweetFileVariable=open(sys.argv[2])

    scoresArray=readSentimentsFunction(sentFileVariable)

    caluculateNewSentimentsFunction(scoresArray, tweetFileVariable)

Here we calculate the score of new term

Command python term\_sentiment.py AFINN-111.txt data.json

4)

# -\*- coding: utf-8 -\*-

"""

Created on Fri Feb 24 01:40:45 2023

@author: pradc

"""

import sys

import json

#This function does the task of those words which are not there in AFIN-11

def termFrequencyFunction(tweet\_file):

    totalCountVariable=0#initializing count variable to zero

    for line in tweet\_file:#for each line do the following.

        termCountArray={} #declaring an empty array#

        tweetArray=json.loads(line)#create a tweet array and provide inputs of each line in file

        if "text" in tweetArray:#if there is a text in array then.

            text=tweetArray["text"].lower()#lower case all those

            wordsArray=text.split()#split the text array on basis of spaces

            for word in wordsArray:# for every word in words array

                totalCountVariable+=1#increment the count variable

                if word in termCountArray:#if there is a word in term array.

                    termCountArray[word]+=1#increment term count array by 1

                else:#else do the following.

                    termCountArray[word]=1#initialize termCountArray by 1

            for term,count in termCountArray.items():

                frequency=float(count)/float(totalCountVariable)#compute the frequency

                print(term,frequency)#encode the frequency.

def main():

    tweet\_file=open('data.json',"r")#open tweet file in read mode

    termFrequencyFunction(tweet\_file)#calling the function.

    tweet\_file.close()#close the tweet file.

if \_\_name\_\_=='\_\_main\_\_':

    main()

This code is used to display frequency of occurrence of each tweet

Command: python term\_sentiment.py AFINN-111.txt data.json

5)

#importing packages

import json

from collections import defaultdict

#Open AFinn file

AfinnFileStorageVariable=open('AFINN-111.txt')

scoresArrayVariable=defaultdict(int)

#Compute scores array

#For each and every line do the following

for lineIterator in AfinnFileStorageVariable:

    termVariable,scoreVariable=lineIterator.split("\t")#split and create array on basis of space

    scoresArrayVariable[termVariable]=int(scoreVariable)#convert scores array variable to integer

#Declaring states array

states = {

        'AK': 'Alaska',

        'AL': 'Alabama',

        'AR': 'Arkansas',

        'AS': 'American Samoa',

        'AZ': 'Arizona',

        'CA': 'California',

        'CO': 'Colorado',

        'CT': 'Connecticut',

        'DC': 'District of Columbia',

        'DE': 'Delaware',

        'FL': 'Florida',

        'GA': 'Georgia',

        'GU': 'Guam',

        'HI': 'Hawaii',

        'IA': 'Iowa',

        'ID': 'Idaho',

        'IL': 'Illinois',

        'IN': 'Indiana',

        'KS': 'Kansas',

        'KY': 'Kentucky',

        'LA': 'Louisiana',

        'MA': 'Massachusetts',

        'MD': 'Maryland',

        'ME': 'Maine',

        'MI': 'Michigan',

        'MN': 'Minnesota',

        'MO': 'Missouri',

        'MP': 'Northern Mariana Islands',

        'MS': 'Mississippi',

        'MT': 'Montana',

        'NA': 'National',

        'NC': 'North Carolina',

        'ND': 'North Dakota',

        'NE': 'Nebraska',

        'NH': 'New Hampshire',

        'NJ': 'New Jersey',

        'NM': 'New Mexico',

        'NV': 'Nevada',

        'NY': 'New York',

        'OH': 'Ohio',

        'OK': 'Oklahoma',

        'OR': 'Oregon',

        'PA': 'Pennsylvania',

        'PR': 'Puerto Rico',

        'RI': 'Rhode Island',

        'SC': 'South Carolina',

        'SD': 'South Dakota',

        'TN': 'Tennessee',

        'TX': 'Texas',

        'UT': 'Utah',

        'VA': 'Virginia',

        'VI': 'Virgin Islands',

        'VT': 'Vermont',

        'WA': 'Washington',

        'WI': 'Wisconsin',

        'WV': 'West Virginia',

        'WY': 'Wyoming'

}

#Store the value 0 in dictionary of state sentiments variable

StateSentimentsVariable=defaultdict(lambda:[0,0])

tweetFileVariable=open('data.json')#open the file data.json and do the following

for lineIterator in tweetFileVariable:#for every line in tweet File

    tweet=json.loads(lineIterator)#load the file

    if "text" in tweet and tweet["place"] and tweet["place"]["country\_code"]=="US" and tweet["lang"] == "en":#if the country code is us and lies in the states of US then

        stateAbbrevation=tweet["place"]["full\_name"][-2:]

        if stateAbbrevation in states:#if state abbrevation lies in states array

            TweetSentimentVariable=sum(scoresArrayVariable.get(word,0) for word in tweet["text"].lower().split())#for each and every word split and create an array

            StateSentimentsVariable[stateAbbrevation][0]+=TweetSentimentVariable

            StateSentimentsVariable[stateAbbrevation][1]+=1#increment the count by 1

StateSentimentsVariable\_avg = {abbrevation: StateSentimentsVariable[abbrevation][0] / StateSentimentsVariable[abbrevation][1] for abbrevation in StateSentimentsVariable}# caluclate the average of each and every tweet of state

# Find the state with the highest average sentiment scoreVariable

displayHappiestState = max(StateSentimentsVariable\_avg, key=StateSentimentsVariable\_avg.get)

# Print the two-letter state abbreviation of the happiest state to standard output

print(displayHappiestState)

The above code is used to display the happiest state.

Command: python happiest\_state.py AFINN-111.txt data.json

6)

import json

#Importing packages

#Getting the hash tags

def getHashTagsFunction(tweet):

    entitiesArrayVariable=tweet.get('entities',{})

    hashtagsArrayVariable=entitiesArrayVariable.get('hashtags',[])

    return [tag['text'].lower() for tag in hashtagsArrayVariable]#return the texts which has tags.

def countTheNumberOfHashTags(tweet\_file):#to count the number of hash tags

    hashtagCountArray={}#declaring count array

    with open(tweet\_file,'r') as f:#open tweet file in read mode

        for lineIterator in f:#for every line

            tweetsArrayVariable=json.loads(lineIterator)#load the file

            hashtagsArrayVariable=getHashTagsFunction(tweetsArrayVariable)#calling function

            for singleHashTag in hashtagsArrayVariable:

                if singleHashTag in hashtagCountArray:#if there is hash tag

                    hashtagCountArray[singleHashTag]+=1#increment it by 1

                else:

                    hashtagCountArray[singleHashTag]=1#else initialize to 1

    return hashtagCountArray#return the function

def printTopTen(hashtagCountArray):#print top 10 hash tags.

    topHashTags=sorted(hashtagCountArray.items(),key=lambda valueVariable:valueVariable[1],reverse=True)[:10]

    for singleHashTag,count in topHashTags:#for every hash tag.

        print(f"{singleHashTag} {count:.1f}")#print the count along with hash tags.

if \_\_name\_\_=='\_\_main\_\_':

    tweetFileVariable='data.json'

    hashtagCount=countTheNumberOfHashTags(tweetFileVariable)

    printTopTen(hashtagCount)

Command: python top\_ten.py data.json