FIT5196 Task 1 in Assessment 1

Student Name: Sarthak Sareen

Student ID: 30761182

Date: 13/09/2020

Version: 1.0

- pandas (for dataframe, included in Anaconda Python 2.7)
- re (for regular expression, included in Anaconda Python 2.7)
- os (interacting with os modules included in Anaconda Python 2.7)
- langid (to check string language, included in Anaconda Python 2.7)

1. Indroduction

In the task we have to analyze the textual data that is the extracting the semi structures files. Each file comtains the id, text and the date created of the tweet. We have to take out the three things i.e id which is a 19 digit number, text which is the actual tweet and the date created that is the dat eat which the tweet was posted. After taking out we have to write all the data to a XML file.

1. Import libraries

In [ ]:

```
1
2 #importing the libraies
3 import os
4 import re
5 import langid
```

2. Loading all the files

In [ ]:

```
#reading all the file which is present the specified folder path where the file is.txt
path = "Ass-1-5196/"
all_files = os.listdir(path)
new_list = []
text2 =''
for root, dirs, files in os.walk(path):
    for file in files:
        if file.endswith('.txt'):
            with open(os.path.join(root, file), 'r',encoding='utf-8') as f:
                text = f.read()
                #saving the all the text into one string
                text2 = text2  + text


```

3. Function to perform the task

In [ ]:

```
#function distribute that will take out the data from the string text2 using regex and
def distribute(data):
    #using regex extracting id into a list
    list_id = re.findall(r'"id":"[0-9]{19}',data)
    #using regex extracting created date into a list
    list_created_date = re.findall(r'created_at":"[0-9]{4}-[0-9]{2}-[0-9]{2}',data)
    #using regex extracting text into a list
    list_text = list_text = re.findall(r'"text":".*?"',data)
    #defining a new dictonary
    d={}
    #adding the first element of the created date list as a key in dictonanry (d)
    #with values first element of id list and text list
    #first_text = []
    first_text = list_text[0].replace("//n","/n")
    first_text = first_text.replace(">","&gt;")
    first_text = first_text.replace("&","&amp;")
    first_text = first_text.replace('"',"&quot;")
    first_text = first_text.replace("'","&apos;")
    d[list_created_date[0].split('"')[2]] = [[list_id[0].split('"')[3],first_text]]
    #assigning this count for the index which is used to take out the id and text from
    count = 0
    #loop in created date list
    for key in list_created_date:
        count= count + 1
        if count >= len(list_created_date):
            #breaking the loop if count is greater than length of list created date
            break
        else:
            #assigning the id and text to a variable with the help of the count variabl
            id_element,text_element = list_id[count].split('"')[3],list_text[count].spl
            text_element = text_element.encode('utf-16', 'surrogatepass').decode('utf-1
            text_element = text_element.replace("//n","/n")
            text_element = text_element.replace("<","&lt;")
            text_element = text_element.replace(">","&gt;")
            text_element = text_element.replace("&","&amp;")
            text_element = text_element.replace('"',"&quot;")
            text_element = text_element.replace("'","&apos;")
            #checking the text is in english or not. Checking id and text are not empty
            if langid.classify(text_element)[0]=='en' and id_element!="" and text_eleme
                keydic = key.split('"')[2]
                #checking the key is present in the dictonary or not
                if keydic in d:
                    #appending the id and text n the present key
                    d[keydic].append([id_element,text_element])
                else:
                    #making the key and adding the values in it
                    d[keydic] = [[id_element,text_element]]
            #idf the text is not english then continue
            else:
                continue
    #returning the dictonary
    return d
#Calling function distribute function and saving it into final variable
final = distribute(text2)
```

In the above function the steps followed are :

1) The function is taking text which is the combined text of all the files as Data.
2) Taking out ID from the text using regex - "id":"[0-9]{19} this means re matches exacty this expression "id":" and then 19 more digits after this. We are taking 19 digits is because we are given that the id is 19 digits. Saving this into list_id.

3) Taking out the Created date from the text using regex - created_at":"[0-9]{4}-[0-9]{2}-[0-9]{2} this means that re matches exactly this expression created_at":" and then 4 digits which are in between 0-9 followed by this expression '-' then 2 digits in between 0-9 follwed by this expression '-' then 2 digits in between 0-9. Saving this regex in list_created date

4) Taking out the text from the text using regex - "text":".*?" this means it matches this expression "text":" then .* means anything ?" means stop when we get 0 or 1 this expression ". Saving this into list_text.

5) defining a empty dictonary

6) Adding the first element of created_date_list as key of the dictonary and values are the list of first element of list_id and first element of list_created. Splitting the three of the to remove the unwanted things in the text. Such as the created_date_list has first element as 'created_at" : "2020-03-23 splitting this to remove the created_date and taking out only 2020-02-23 from it. Performing the same task for id and text.

7) Initialising count which will be used for the index to take out theh id element from the list and text.

8) Looping in the created_date_list to take this as key of the dictonary.

```
i) First check is the count should be less that length of the ceated date list because if
the count which is used for index       should be in the limit of the created date list l
ength only.
  If this happens the loop will break.

ii) Then in the else condition id and text is being assigned to varibles id_element and t
ext_element respectively using the count as the index and splitting the unwanted text fro
m it.

iii) A) check if the text is english and id_element is not empty and text_element is not
 empty.
     --splitting the created date_list to take out only the date
     --checking if the key is already present then append the id_element and text_element
as values into the key
     --else making the new key and taking id_element and text_element as values into the
 specific key.

    B) else
     --if the text is english and id_element is not empty and text_element is not empty t
hen continue. Notjhing will go                into the new dictonary. This will skip the
 entry.
```

9) returning the dictonary

10) calling the function and saving it into a variable final.

In [ ]:

```
#making string
stri = '<?xml version="1.0" encoding="UTF-8"?>' + '\n' + '<data>\n'
#iterating in the final dictonary
for key,value in final.items():
    #adding keys of the dictonary in the string
    stri = stri +  '\t' + '<tweets date="'+str(key) + '">\n'
    #iterating in the values of the dictonary
    for i in value:
        #adding the tweet id and the text in the string
        stri = stri + '\t\t' + '<tweet id="'+ str(i[0]) + '">' + str(i[1]) + '</tweet>\
    stri = stri + '</tweets>\n'
stri = stri + '</data>'
stri= stri.replace("\\n","\n")
```

In the above cell a string is made which contains all the tweets date's tweet id and the text of the specifc date and concating all of the into one string which is stri. in between the string adding the tags of the data , tweet date and tweets to make the text structured liek a XML doc.

In [ ]:

```
#writing the string into the xml format
with open("30761182.xml", "w", encoding='utf-8') as text_file:
    text_file.write(stri)
```

Summary

1) First reading all the text files into one string.

the string contains all the text of all the files.

2) Then using regex extracting id , text , created date :

list_id = ["id":"1234554412323456789,"id":"1234533412341112345,....]

list_created_date = ["created date" :"2020-03-23,"created date":"2020-04-22,....]

list_text = ["text":"conrona virus","text":"virus is bad",.....]

3) Making a new dictonary with all the handling of the text. The dictonary formed id like :

```
{ "2020-03-22" : [[1233443300987475849.'coronora virus'],[2134443098712345980,'vir
us is bad'],....] }
```

4) Then making a string according to the XML document

5) writing the string in the output XMl file.

The output file is being creating as xml document having tags and is parsable.