DATA MANAGEMENT, WAREHOUSING, AND ANALYTICS

CSCI 5408

Fall 2018

ASSIGNMENT 2

Gurjot Singh (B00811724)- gurjot@dal.ca

Devanshu Srivastava (B00810667)- dv960112@dal.ca

TASK DESCRIPTION

The task was to perform Sentimental Analysis followed by ElasticSearch on Twitter Tweets Data using Tweepy API. The main objective was to determine the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions and emotions expressed within an online mention.

The first step was to create a Twitter Developer Account. Once the access was granted, Twitter application was created using the “Consumer Key” and the “Consumer Secret. The first program developed in Python language was used to capture the Twitter Tweets using the Twitter API and the credentials. 1001 tweets were fetched in a Comma-Separated Values(CSV) format. The raw data fetched from the Twitter API was cleaned to remove redundancies, special characters and the cleaned data was then used during the sentimental analysis process.

The second program was also developed in Python Language to perform the sentimental analysis on the tweets data. The lexicon file “Concreteness.json” downloaded from the internet was used in the sentimental analysis. Using this file, sentimental analysis was done on the extracted tweets and the polarity score was recorded along with the type of analysis i.e. “positive”, “negative”, or “neutral”.

After the successful execution of sentimental analysis on the tweets data, the data was uploaded to ElasticSearch Database using an ElasticSearch API.

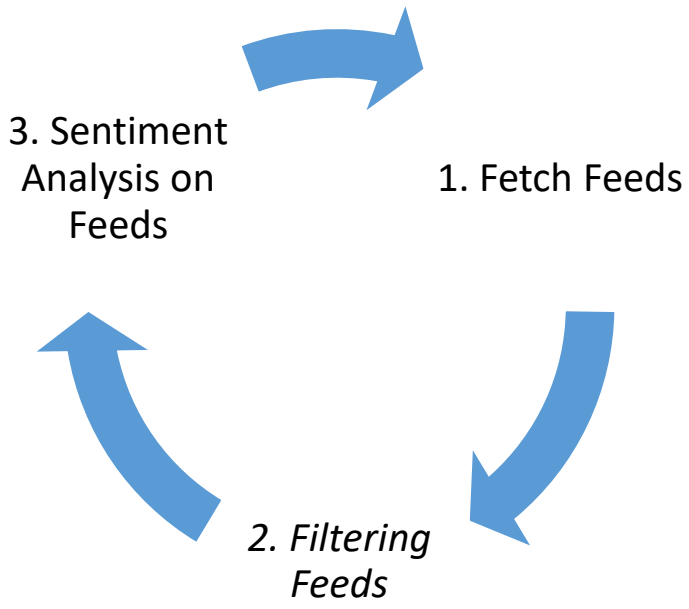
TWITTER TWEET EXTRACTION

The first step was to create a Twitter Developer Account. Once the access is granted, the “Consumer Key”, “Consumer Secret”, “Access Token” and “Access Token Secret” were ued for setting up connections to twitter. Whenever the application launches, the console prompts the user to input the queries (i.e. the profile names on twitter) on which the extraction of tweets were to be done and the input queries were comma separated.

```
def FetchTweets():
input  = input("Provide words in comma separated format: ")
yourstring = input.split(",")
queries = []
        for x in yourstring:
                queries.append( "@{0}".format(x))
 return queries
```

The function ‘FetchTweets’ was used to help the user to input the twitter profile feeds. The function was tested for 1000 tweets and no Third-party API’s were used to execute this approach.

The fetched tweets were then written into a CSV file ‘tweets_user.csv’. The raw data fetched from the Twitter API was then cleaned in a proper format to remove redundancies, special characters and the cleaned data was then stored in a new CSV file to enhance the sentiment analysis.



Calling for ETL (Extraction Transform Load) of Twitter feeds

SENTIMENT ANALYSIS ALGORITHM

The tweets were analyzed with the lexical file to provide the results. For this, the filtered tweets were iterated along with the lexical file: Concreteness.json containing 24000 key/value pairs of positive and negative words. The polarity was calculated on the basis of the matched tweets and the lexical file. The polarity and the result was then stored in 'sentimental_analysis.csv' file. The total time of execution of sentimental analysis of 1001 tweets was 35 seconds. The result was then categorized based on positive, negative or neutral words and the polarity score was stored.

SentimentAnalysis

SentimentAnalysis.py

GetPolarityFeeds

```
    #inserting to csv file

    with open(cleanTweetFile,'w',newline='',encoding='utf8') as outfile:
        writer=csv.writer(outfile)
        writer.writerow(['text'])
        for i in listItem:
            writer.writerow([i])


    ####performing semantal analysis

    with open(lexiconFile,"r") as f:
        loaded_json=json.load(f)
        print("START TIME",datetime.utcnow())

            #polarity=0
        for j in listItem:
            temp=j.split(" ")
            polarity=0
            for k in temp:
                for x in loaded_json:
                    if k==x:

                        polarity +=int(loaded_json[x])
                        #print(x,k,loaded_json[x],polarity)
                #print(polarity,j)
            #print("Tweet",j,"Polatiry: ",polarity)
           # print("TIME",datetime.utcnow())

            if(polarity==0):
                polarity_list.append(polarity)
                polarity_list1.append("Neutral")
#                sentimental_analysis.append("the polarity is neutral and polarity score is "+str(polarity)+" for : "+j)
            elif(polarity<0):
                polarity_list.append(polarity)
                polarity_list1.append("Negative")
                #sentimental_analysis.append("the polarity is negative and polarity score is "+str(polarity)+" for : "+j)
            elif(polarity>0):
                polarity_list.append(polarity)
                polarity_list1.append("Positive")
                #sentimental_analysis.append("the polarity is positive and polarity score is "+str(polarity)+" for : "+j)
        print("END TIME",datetime.utcnow())

    #creating a csv file to store sentimental anylisis
    with open(sentimentFile,'w',newline='',encoding='utf8') as outfile:
        writer=csv.writer(outfile)
        writer.writerow(['The Twitter Tweet',"The Sentiment(Positive,Negative, or Neutral)","The Sentiment Score"])
        for i in range(0,len(listItem)):
            writer.writerow([listItem[i],polarity_list1[i],polarity_list[i]])
```

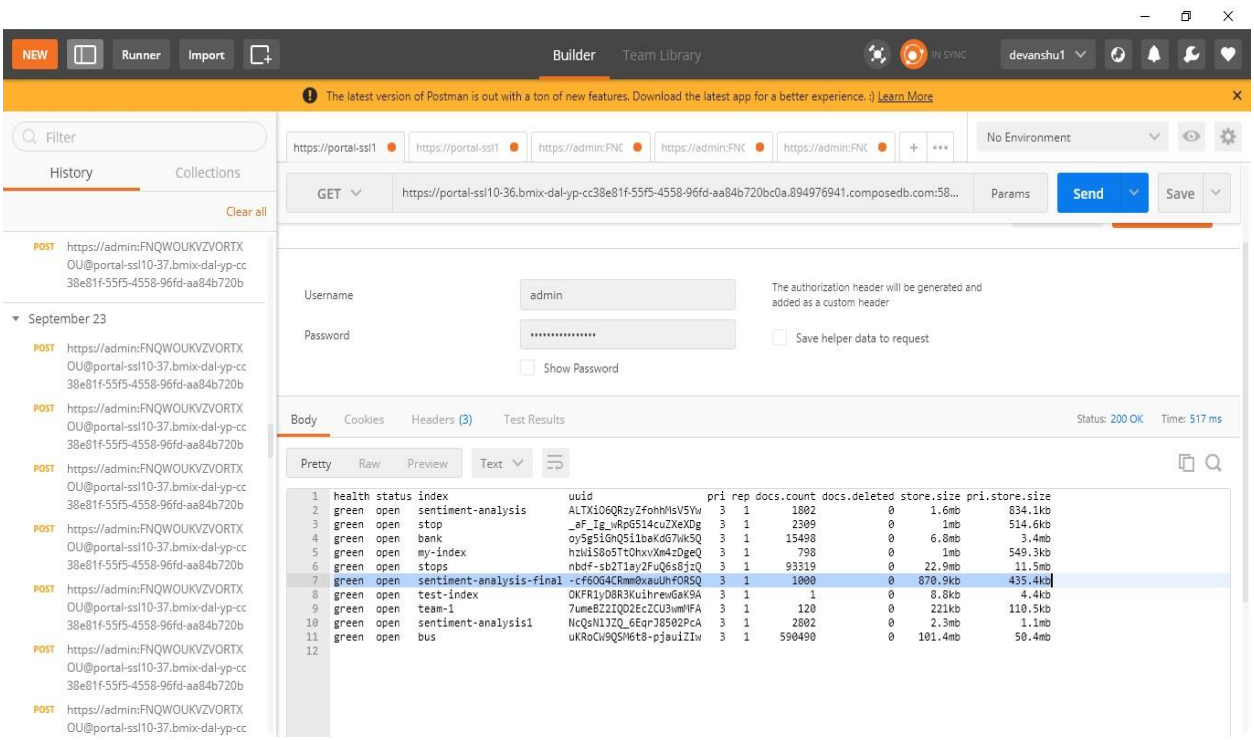
74 %

Code for Sentimental Analysis and Polarity Calculation

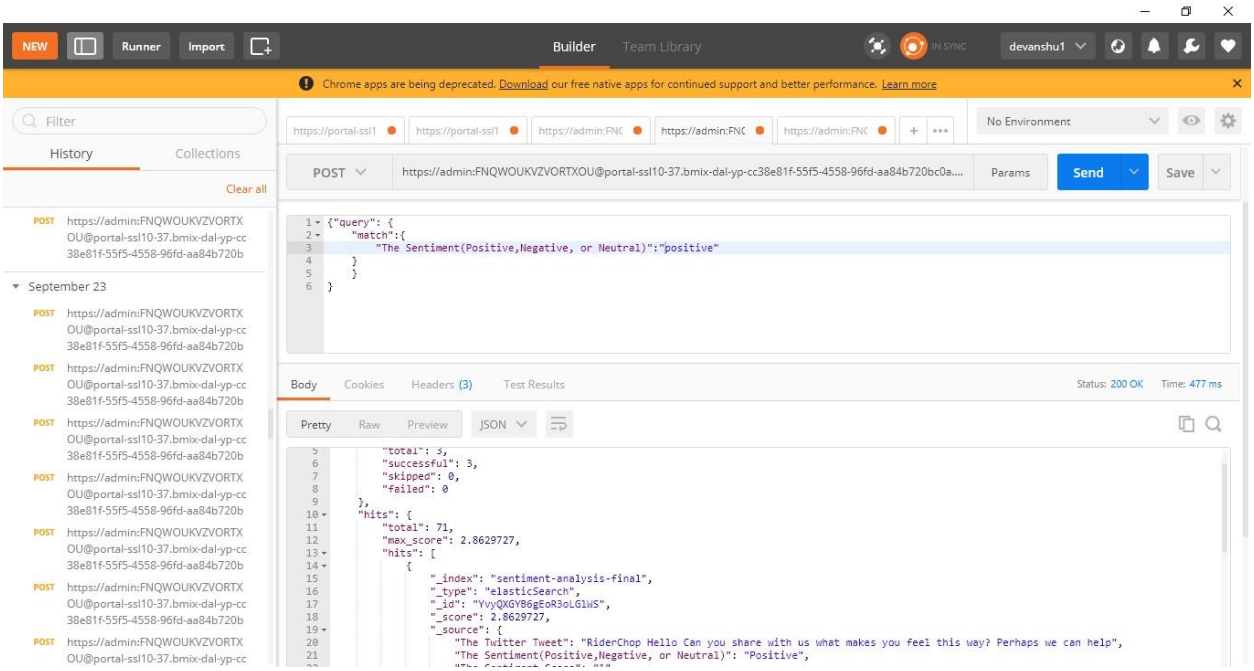
LOADING DATA INTO ELASTICSEARCH DATABASE

For ElasticSearch, 'from elasticsearch import helpers, Elasticsearch' was imported after the completion of Sentimental analysis process. Before initiating the ElasticSearch, the user input was required whether to perform ElasticSearch or not.

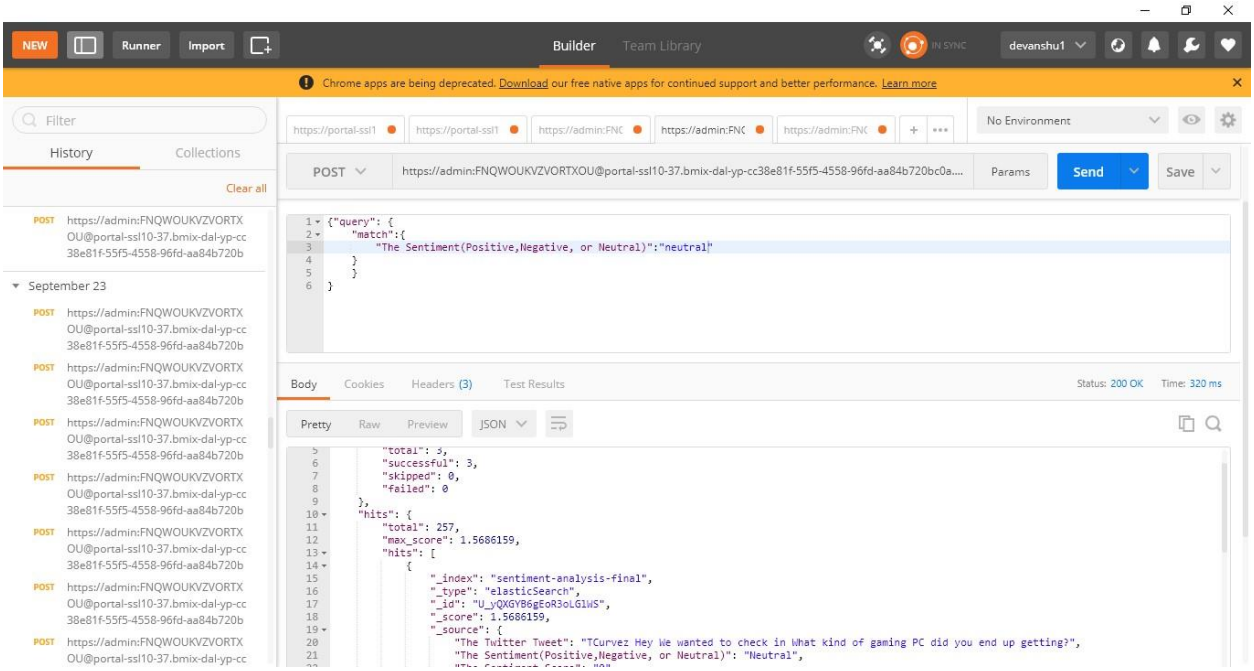
```
def InitiateElasticSearch():
    es = Elasticsearch(["https://admin:FNQWOUKVZVORTXOU@portal-ssl10-37.bmix-dal-yp-
cc38e81f-55f5-4558-96fd-aa84b720bc0a.894976941.composedb.com:58282/"])
    with open('sentimental_analysis.csv','r',encoding='utf-8') as f:
       reader = csv.DictReader(f)
       helpers.bulk(es, reader, index='sentiment-analysis', doc_type='elasticSearch')
    return "Data for Elastic Search uploaded."


def ElasticSearch():
    val = input("yes/no for Elastic Search") #User will input YES/NO for initiating
elastic search.
    if(val.upper() == "YES"):
        return InitiateElasticSearch()
    else:
        return "You entered NO, Sorry!"
```



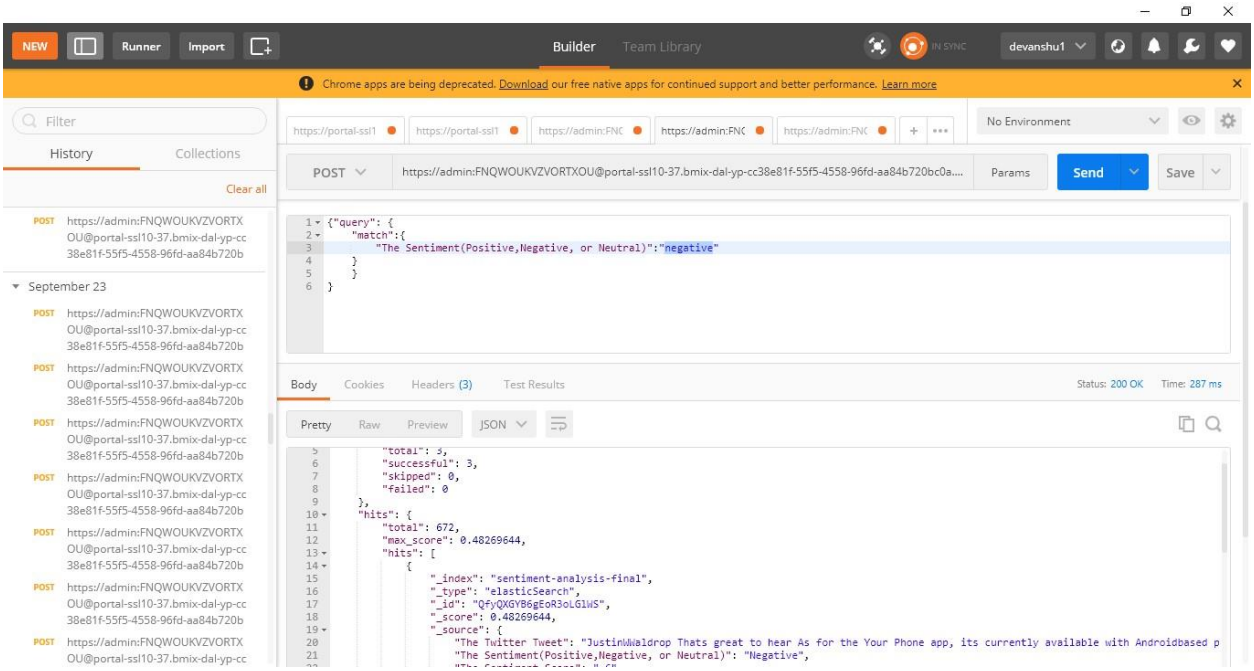
ElasticSearch Output



Positive Tweets Output



Neutral Tweets Output



Negative Tweets Output

CODE SUBMISSION

We have used one extra branch for developing the Sentiment-Analysis project and the name of the branch is dev_branch. Hence, we first commit our changes in that sub-branch and then we review the code and thereafter we merge it to main master branch. So that our code is well managed.

Link to Sentiment-Analysis (repository):

git clone https://git.cs.dal.ca/devanshu/Sentiment-Analysis.git

NOTE: Dalhousie CSID will be credentials are required to access the files

REFERENCES

[1]Remove specific character from a csv file, "Remove specific character from a csv file, and rewrite to a new file", *Stack Overflow*, 2018. [Online]. Available: https://stackoverflow.com/questions/17176542/remove-specific-character-from-a-csv-file-and-rewrite-to-a-new-file. [Accessed: 05- Oct- 2018].

[2]*GitHub*, 2018. [Online]. Available: https://github.com/williamleif/socialsent/blob/master/socialsent/data/lexicons/concreteness.json. [Accessed: 04- Oct- 2018].

[3]codecs — Codec registry and base classes — Python 3.7.1rc1 documentation", *Docs.python.org*, 2018. [Online]. Available: https://docs.python.org/3/library/codecs.html. [Accessed: 01- Oct- 2018].

[4]N. Congleton, "How to Parse Data From JSON Into Python - LinuxConfig.org", *Linuxconfig.org*, 2018. [Online]. Available: https://linuxconfig.org/how-to-parse-data-from-json-into-python. [Accessed: 06- Oct- 2018].

[5]"Python JSON", *W3schools.com*, 2018. [Online]. Available: https://www.w3schools.com/python/python_json.asp. [Accessed: 03- Oct- 2018].