CSCI 5408

Assignment 3

REPORT 2

MEGHNA RAMACHANDRA HOLLA | B00812604

A. Data Upload

1. Twitter Data

I used tweets gathered from Assignment2 and the data is in CSV file. We were told by Dr. Dey that screenshots of accessing this file would be sufficient for 'Data Upload' section. I have written a python script 'combineTweetOnly' to combine my previous search and stream tweets that gives the output in 'allTweets.csv' file. The tweets gathered in 'allTweets.csv' are more that 1000 tweets. I have included these files in 'sentimentAnalysis' folder. Figure 1 indicates the method of accessing the files.

```
with open('cleanSearchFile.csv') as f:
    r=csv.reader(f,delimiter=',')
    dict1=[row[2] for row in r]

with open('cleanStreamFile.csv') as f:
    r=csv.reader(f,delimiter=',')
    dict2=[row[2] for row in r]
```

Figure 1 Data Upload of Twitter

These tweets were uploaded on the AWS server (Figure 2) and the sentiment analysis was implemented.

```
ubuntu@ip-172-31-16-183:~/assign3/sentimentAnalysis$ ls -l
total 664
-rw-rw-r-- 1 ubuntu ubuntu 131563 Mar 21 18:32 allTweets.csv
-rw-rw-r-- 1 ubuntu ubuntu  69990 Feb 26 14:52 cleanSearchFile.csv
-rw-rw-r-- 1 ubuntu ubuntu 140772 Feb 26 14:52 cleanStreamFile.csv
-rw-rw-r-- 1 ubuntu ubuntu    411 Mar 21 18:32 combineTweetOnly.py
-rw-rw-r-- 1 ubuntu ubuntu  49541 Mar 12 22:50 negative-words.txt
-rw-rw-r-- 1 ubuntu ubuntu  21095 Mar 12 22:50 positive-words.txt
-rw-rw-r-- 1 ubuntu ubuntu   2412 Mar 21 18:33 sentimentAnalysis.py
-rw-rw-r-- 1 ubuntu ubuntu 236313 Mar 21 18:33 sentimentTweets.csv
-rw-rw-r-- 1 ubuntu ubuntu   2912 Mar 12 12:52 stopwords.txt
```

Figure 2 Implementation on AWS

2. Reuter Data

I used PyCharm IDE in my local system to access Reuter data. I created a folder in the working directory for Reuter data and then accessed. Figure 3 indicates the method of accessing these files.

```
for i in range(0, 22):
    if(len(str(abs(i))))==1:
        f=open('reuters/reut2-00'+str(i)+'.sgm', 'r')
    else:
        f=open('reuters/reut2-0'+str(i)+'.sgm', 'r')
    data=f.read()
    f.close()
```

Figure 3 Data Upload of Reuter Data

The data in the working directory can be seen on Figure 4.

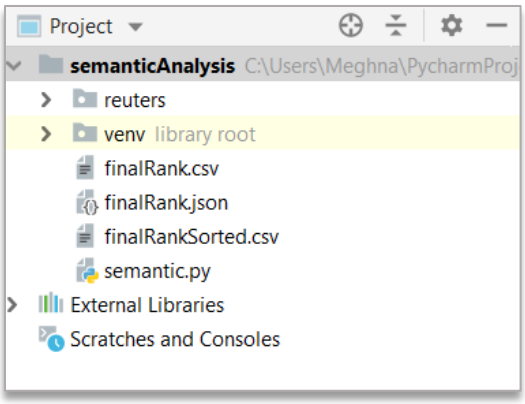

Figure 4 Project Directory in local system

B. Exploratory Study

The report on Sentiment Analysis can be found in the file '**Report1_Ramachandra Holla_Meghna_B00812604**'.

C. Data Extraction, Transformation and Analytics

1. Sentiment Analysis of Twitter Data

The tweets from Assignment2 that I gathered have been cleaned already. I later got to know from my TA that the few of the tweets are partially complete. Since there was time constraint, I was told by the TA to mention this in my report to avoid being penalised for it. Irrespective of this, my sentiment analysis is working with the data that I have. There are no special characters, emojis and hyperlinks in my tweets. Sentiment Analysis requires only tweets, so we were asked by Dr. Dey to extract only the tweets. The method that I followed to clean tweets for Sentiment Analysis is as follows:

- I initially combined the tweets that I had gathered in Assignment2 (cleanSearchFile.csv and cleanStreamFile.csv) by creating a script 'combineTweetsOnly.py'. The result of this is in 'allTweets.csv' file. There are only tweets in this file
- The file 'sentimentAnalysis.py' has the logic I wrote for Basic Sentiment Analysis.
- I gathered numerous stopwords in 'stopwords.txt' file, positive words in 'positivewords.txt' and negative words in 'negativewords.txt'
- I removed stop words from my tweets by referring the list of stop words.
- I also stemmed the tweets, but I noticed that words like 'family' or 'crazy' were getting stemmed into strange words. Therefore, I have commented out the logic for stemming, but it can still be implemented in my code if they are uncommented.

- I have then compared the number of positive and negative words in each tweet to identify the tweet's polarity. The tweet with more positive words is tagged as a positive tweet. The tweet with more negative words is tagged as a negative tweet. The tweet with equal number of positive and negative words is tagged as a neutral tweet.
- The final output shows number of Positive Tweets as 145 and number of Negative tweets as 846. I have more negative tweets because Halifax had 'accident' topic trending at the time of streaming. Figure 5 shows the output.

```
ubuntu@ip-172-31-16-183:~/assign3/sentimentAnalysis$ python sentimentAnalysis.py
('No. of Positive Tweets:', 145)
('No. of Negative Tweets:', 846)
```

Figure 5 No. of Positive and Negative Tweets

- The information of the polarity of the tweets is stored in 'sentimentTweets.csv' file. The file has five columns – original tweets, the final sentiment, tweets after cleaning for sentiment analysis, no. of positive words in a tweet and no. of negative words in a tweet. A sample of the polarity is shown in Figure 6.

the band in Halifax is playing sonny s dream with three electric guitars and I am honestly one chord away from filing a noise complaint	Negative
Customer Loyalty Representative Admiral Insurance Services Halifax NS Halifax Nova Scotia	Neutral
RT AlbiDeak Halifax Discovery Centre celebrates 38th annual World Whale Day CetaceanRights RacingExtinction	Positive

Figure 6 Sample Positive and Negative Tweets

2. Semantic Analysis of Reuter Data

I wrote my own logic for semantic analysis in 'semantic.py' file inside 'semanticAnalysis' folder. The steps taken to extract, transform and evaluate the Reuter data is as follows:

- I extracted 'Body' tag details from Reuter data using regular expression (Figure 7).

```
cleanData= re.sub('\n|\r', '', data)
result = re.findall('<BODY>(.*?)</BODY>', cleanData)
contents.extend(result)
```

Figure 7 Processing only Body Tag

- I stored all the extracted data in a dictionary instead of multiple files because I noticed that the program was becoming slow due to opening and closing of 1000s of files. I got **19043 articles** (documents) in the 22 SGM files. The time taken to process these 22 SGM files is about 30 minutes
- I removed special characters from the data as I did not want my code to take special characters as separate words.
- I gathered a set of unique words from the dictionary and calculated the word count across different articles. After further logarithmic calculations, IDF for each word was calculated.
- I calculated TF and implemented Distance of each article.
- Distance for query 'Canada' was calculated too.

- Cosine similarity of each article (document) with the query was calculated to get the ranks of the documents. I sorted the ranks in descending order (highest ranked document on top) and stored them in a JSON file.
- Dr. Dey asked me to either create a text file or a JSON file to store the ranked documents. Therefore, I created a JSON file to store the ranked documents namely '**finalRank.json**'

- I found **676 articles** that had 'Canada' in them.
- The output when I tested for articles from first five SGM files is:

```
('The top most rank is:', 0.0672746893664342)
('The article number:', 'reut_article1605')
('The content is', 'chrysler canada ltd whollyowned by chrysler corp said february car sal
es fell to 9 640units from year earlier 11 967 units chrysler canada said year to date car
 sales fell to 18 094units from 22 073 units in the same period last year reuter 3 ')
```

- The output when I tested with all the 22 SGM files is:

```
('The top most rank is:', 0.06525281677796065)
('The article number:', 'reut_article12314')
('The content is', 'israel introduces summer time april 26 canada introduces summer time reuter 3 ')
```

- The first few json objects from top ranked documents are:


```
[
  {
    "article":"reut_article12314",
    "rank":0.06525281677796065,
    "content":"israel introduces summer time april 26 canada
    introduces summer time reuter 3 "
  },
  {
    "article":"reut_article12712",
    "rank":0.05367385266119122,
    "content":"the bank of canada said the u s dollar atsoon
    was 1 3053 compared to1 3074 dollars yesterday the pound
    sterling rates at noon 2 1130 compared to 2 1147yesterday
    3 "
  },
  {
    "article":"reut_article11905",
    "rank":0.052833092841942694,
    "content":"shr 5 56 dlrs vs 3 88 dlrs net 47 5 mln vs 33 2
    mln revs 254 5 mln vs 243 5 mln note shr after preferred
    dividends itt corp lt itt owns 100 pct of itt canada
    common shares reuter 3 "
  },
  {
    "article":"reut_article11602",
    "rank":0.05172640292555252,
    "content":" lt bank of nova scotia said it raised itsu s
    dlr base lending rate in canada to 8 1 4 pct from eightpct
    effective immediately reuter 3 "
  },
```

References

[1] https://docs.python.org/2/library/re.html
[2] https://docs.python.org/3/tutorial/datastructures.html
[3] https://docs.python.org/3/library/math.html
[4] https://python-reference.readthedocs.io/en/latest/docs/dict/
[5] https://www.w3schools.com/python/python_json.asp
[6] https://dal.brightspace.com/d2l/le/content/86639/viewContent/1271284/View