ASSIGNMENT-3 REPORT

Name: Shrey Amin

Student ID: B00822245

Course: CSCI-5408

Analysis

1. How many tweets are positive, negative and neutral?

```
Total Positive Tweets : 231
Total Negative Tweets : 760
Total Neutral Tweets : 490
```

2. A sample tweet with polarity.

```
Total tweets : 1481
Tweets with polarity
 [['7 children killed in halifax house fire 2 adults sent to hospital ', 'negative'],
```

3. For Reuters, how many documents are created after cleaning?

```
total words 36442
total documents 19043
Keywords : ['canada', 'showers', 'continued', 'throughout',
```

4. Extracted data from the top the top ranked document.

```
Index of the highest ranked document : 7628  and its cosine value : 0.33091514259
Number of times query appeared in the highest ranked document : 3
Data in the highest ranked dcoument :
  &lt;canada development corp> said it agreed with investment dealers wood gundy
```

Data Upload

Tweets

The tweets collected are stored in the csv file. This file is then stored in the MongoDB database using MongoDB compass.

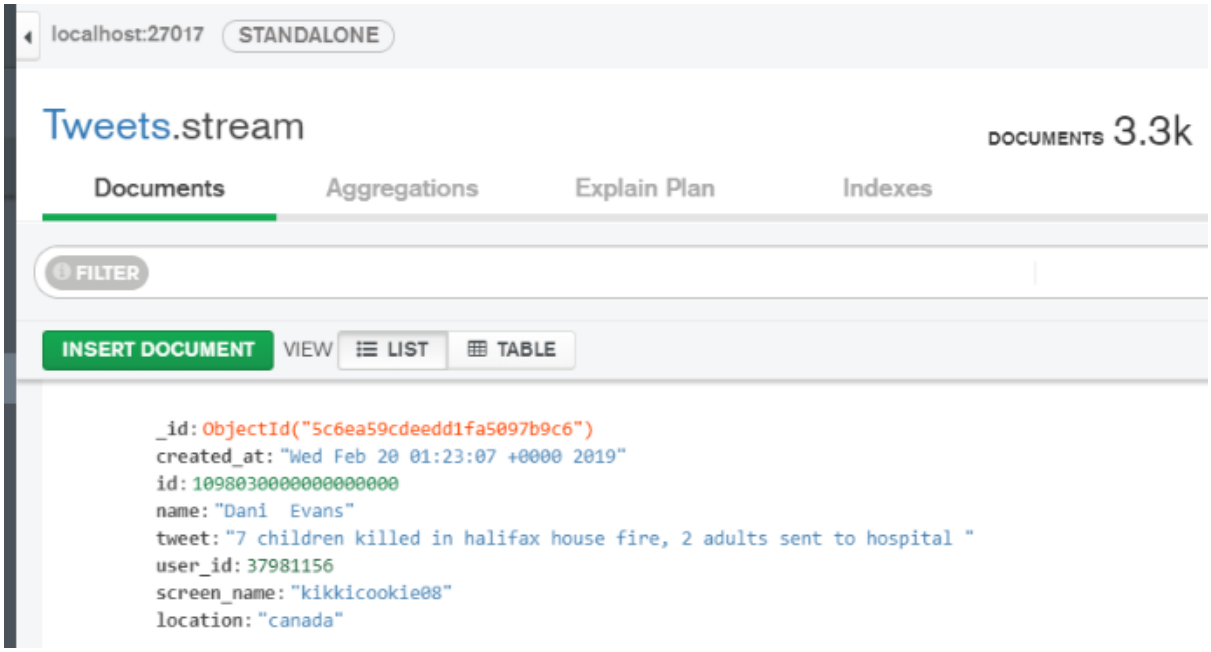

Figure 1 Tweets

Reuters

The given Reuters dataset is cleaned, and each document created is stored as row in the csv file. Then, the csv file is loaded into MongoDB.

Here data is stored in the local MongoDB server.

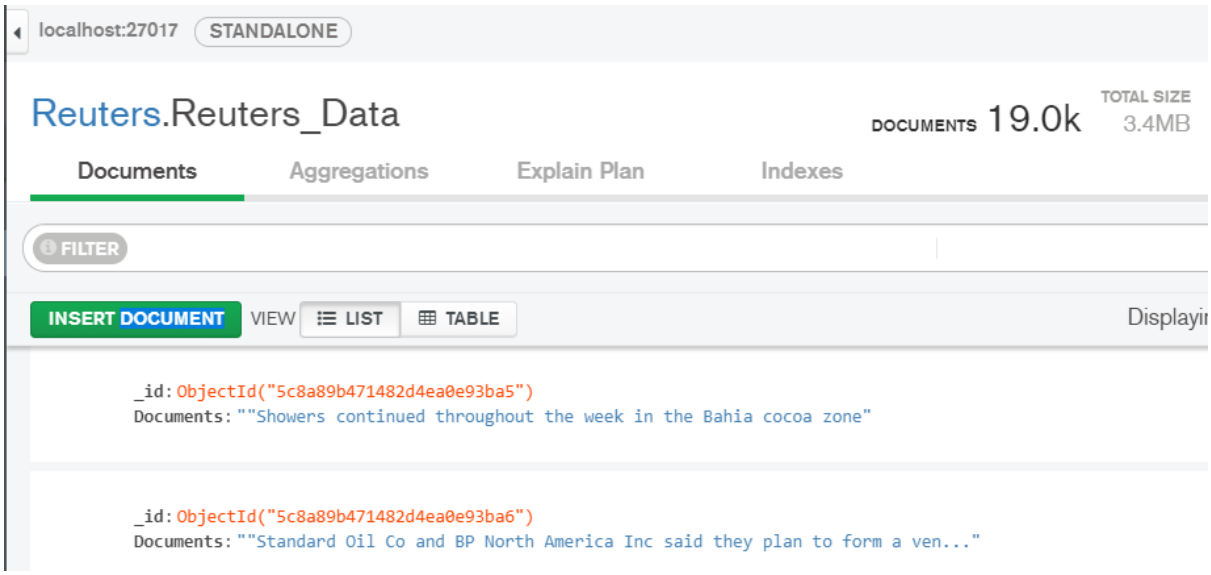

Figure 2 Documents created from Reuter dataset

Data Extraction, Transformation and Analysis

Tweets

Here the tweets collected in assignment-2 are used for sentiment analysis. Following are the steps performed for determining the polarity of each tweet:

- Load the tweets from the csv file and consider only its text attribute (not considered attributes like created_at, name, id, location and screen_name).
- Use only the tweets in English language.
- Perform cleaning by removing special characters and keyword “rt” from each tweet.
- Generate tokens (words) from each tweet.
- Remove stop words from the words generated.
- Find polarity of each tweet using the dictionary of positive and negative words.
- Each tweet and its polarity are stored in the output csv file.

```
"C:\Users\Shrey Amin\untitled\Scripts\python.exe" "C:/Users/Shrey Amin/PycharmProjects
positive dictionary : ['a+', 'abound', 'abounds', 'abundance', 'abundant', 'accessable
negative dictionary : ['2-faced', '2-faces', 'abnormal', 'abolish', 'abominable', 'abo
stopwords dictionary : ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', '
Raw tweets
 ['7 children killed in halifax house fire, 2 adults sent to hospital ', '"7 children
Cleaned tweets
 ['7 children killed in halifax house fire 2 adults sent to hospital ', ' 7 children i
Tokenising tweets
 [['7', 'children', 'killed', 'in', 'halifax', 'house', 'fire', '2', 'adults', 'sent',
Removed the stopwords
 [['7', 'children', 'killed', 'halifax', 'house', 'fire', '2', 'adults', 'sent', 'hosp
Total tweets : 1481
Tweets with polarity
 [['7 children killed in halifax house fire 2 adults sent to hospital ', 'negative'],
Total Positive Tweets : 231
Total Negative Tweets : 760
Total Neutral Tweets : 490
```

Figure 3 Sentiment Analysis

Reuters

The steps involve in information search retrieval process are as follows:

- Clean the Reuter datasets by considering the data in the “BODY” tag using regular expression.
- Data from each body tag is stored in the csv file. Each row in the csv file is considered as one document.
- Unique keywords are generated from the documents.
- TF and IDF matrix are created using number of keywords and documents.

- Based on these matrices and query “Canada”, Query Distance, distance matrix of each document and cosine values are calculated.
- The documents are ranked using the cosine values.
- Finally, data from the highest ranked document is extracted**.**

Initially the program was tested on sample example mentioned in the class lecture and the program gave the correct output. Then, I executed the script on the documents created from the Reuter dataset.

Document by Terms (TF multiplied with IDF)

	Canada	Halifax	School	Park
D1	0	0.584	0	0.584
D2	0	0.584	1.584	0.584
D3	0	0	0	0

Query: Halifax

	Canada	Halifax	School	Park
Query	0	1/2 x 0.584 = 0.292	0	0

D1 distance: $\sqrt{(0^2+0.584^2+ 0^2+0.584^2)} = 0.825$

D2 distance: $\sqrt{(0^2+0.584^2+1.584^2+0.584^2)} = 1.786$

D3 distance: $\sqrt{(0^2 + 0^2 + 0^2 + 0^2)} = 0$

Query distance: $\sqrt{(0^2 + 0.292^2 + 0^2 + 0^2)} = 0.292$

Cosine Similarity:

D1 and Query
cosine([0, 0.584, 0, 0.584], [0, 0.292, 0, 0])
= ((0*0)+(0.584*0.292)+(0*0)+(0.584*0))/(0.825*0.292)
= 0.707

D2 and Query
cosine([0, 0.584, 1.584, 0.584], [0, 0.292, 0, 0])
= ((0 *0)+(0.584 *0.292)+(1.584 *0)+(0.584 *0))/(1.786*0.292)
= 0.326

D3 and Query
cosine([0, 0, 0, 0], [0, 0.292, 0, 0])
= ((0*0)+(0*0.292)+(0*0)+(0*0))/(0*0.292) = err (not releavent to query)

Order of documents: D1, D2

Figure 4 Sample example

```
canada halifax park canada halifax school park canada
['canada', 'halifax', 'park', 'school']
Term Frequency matrix
 [[1, 1, 1, 0], [1, 1, 1, 1], [1, 0, 0, 0]]
occurrence of each word [[3.0, 2.0, 2.0, 1.0]]
idf table [[0.0, 0.5849625007211562, 0.5849625007211562, 1.5849625007211563]]
keyword halifax
index of query word 1
idf of query word 0.5849625007211562
No of documents having the query 2.0
query matrix using idf value and frequnedy count [[0.0, 0.2924812503605781, 0.0, 0.0]]
distnace matrix of each document [[0.8272619019995404], [1.7878669925898176], [0.0]]
calculate query distance 0.2924812503605781
cosine similarity between document and query [0.7071067811865476, 0.32718457421365993, 0]
index of highest ranked document is  0  and its cosine value is 0.7071067811865476
Number of times query appeared in the highest ranked document is  1
Data in the dcoument is
 canada halifax park
```

Figure 5 Document ranking on example

Below is the output of the program executed on the documents created from Reuters dataset. Based on the cosine values the documents are ranked and data is retrieved from the highest ranked document.

```
216973155684036, 13.216973155684036, 14.216973155684036, 13.216973155684036, 14.216973155684036, 1
keyword canada
Index of query word 0
idf of query word 4.8160937194018505
No of documents having the query 676.0
Query matrix using idf value and frequency count [[0.0071243989931980035, 0.0, 0.0, 0.0, 0.0, 0.0,
Distnace matrix of each document [[237.1546087372153], [68.45434553296299], [50.9280364610423], [2
Calculate query distance : 0.0071243989931980035
cosine similarity between document and query : [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
Index of the highest ranked document : 7628  and its cosine value : 0.33091514259734695
Number of times query appeared in the highest ranked document : 3
Data in the highest ranked dcoument :
  &lt;canada development corp> said it agreed with investment dealers wood gundy inc and dominion
```

```
Data in the highest ranked dcoument :
  &lt;canada development corp> said it agreed with
investment dealers wood gundy inc and dominion securities
inc to sell six mln common shares in canada for 60 750 000
 dlrs at 10 125 dlrs a share      the company said it
filed a preliminary prospectus for the offer in ontario
and was filing across the rest of canada
```

Figure 6 Document ranking on Reuter dataset

References

[1] B. Liu, "Opinion Mining, Sentiment Analysis, Opinion Extraction", *Cs.uic.edu*, 2019. [Online]. Available: https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html.

[2] "langdetect.detect Python Example", *Programcreek.com*, 2019. [Online]. Available: https://www.programcreek.com/python/example/102490/langdetect.detect.

[3] "Natural Language Toolkit — NLTK 3.4 documentation", *Nltk.org*, 2019. [Online]. Available: http://www.nltk.org/.

[4] "Extracting text from between tags", *DaniWeb*, 2019. [Online]. Available: https://www.daniweb.com/programming/software-development/threads/278313/extracting-text-from-between-tags#post1199989.