A. Sentiment Analysis

In order to perform the sentiment analysis on the tweets, I have used the tweets that were extracted using the tweets.py algorithm for Assignment 3 which is added as a reference and I have also made few changes in extracting the number of tweets. I have modified the file SentimentAnalysis.py to extract the tweets based on the keywords provided (‘canada’, ‘halifax’, ‘university’, ‘dalhousie university’, ‘canada education’).

The tweets that are collected are cleaned the data such as removing URL’s and emoticons using the regular expressions. Each tweet is further divided into the bag of words where it contains each word and its count in the tweet saved into the json file. After saving the bag of words, I compared these words with the negative and positive words that are saved in the text file.

Based on the count of appearance of the negative and positive words in the respective tweet, I have saved the tweet no, tweet message or text, matched word and the polarity in tweets_sentiment.csv file.

Using the sentiments extracted from the tweets, I have visualized the data and generated multiple reports using Tableau Visualization tool.

Report 1:

Visualization of positive and negative words that appeared in tweet messages.

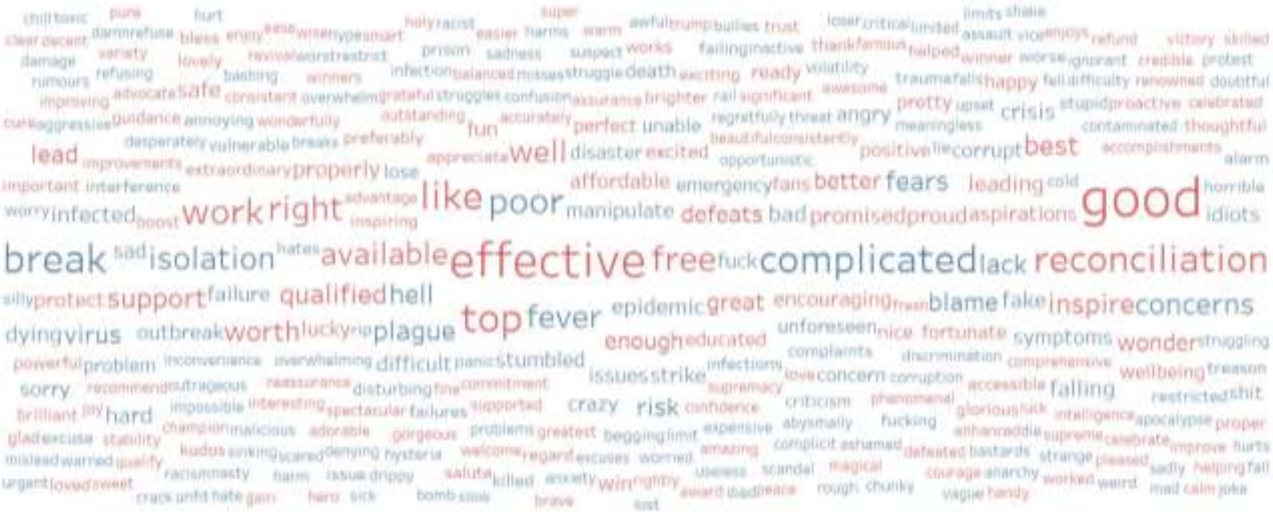


Figure 1 Positive and Negative words

Report 2:

Visualization of number of records that have positive and negative words that are appeared in the tweet.

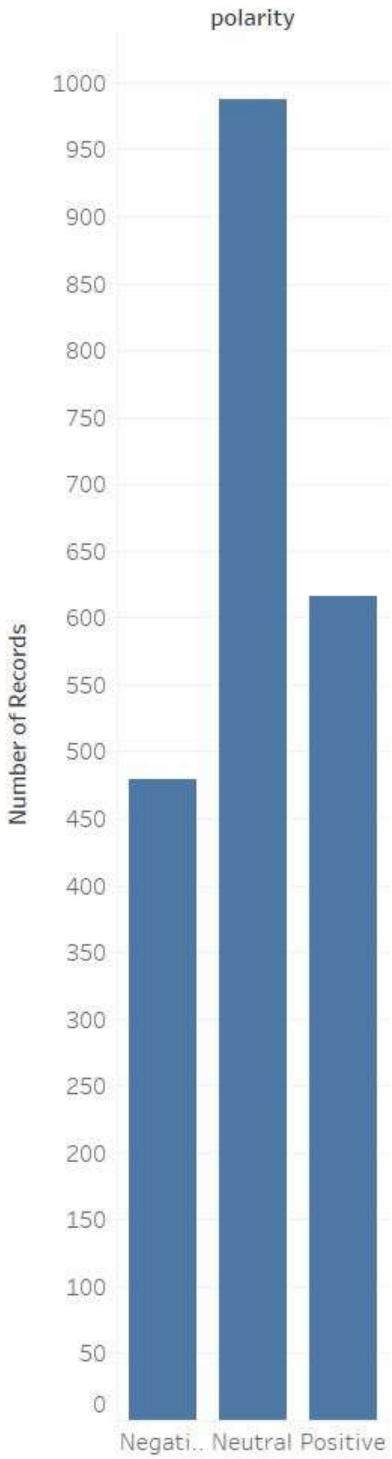
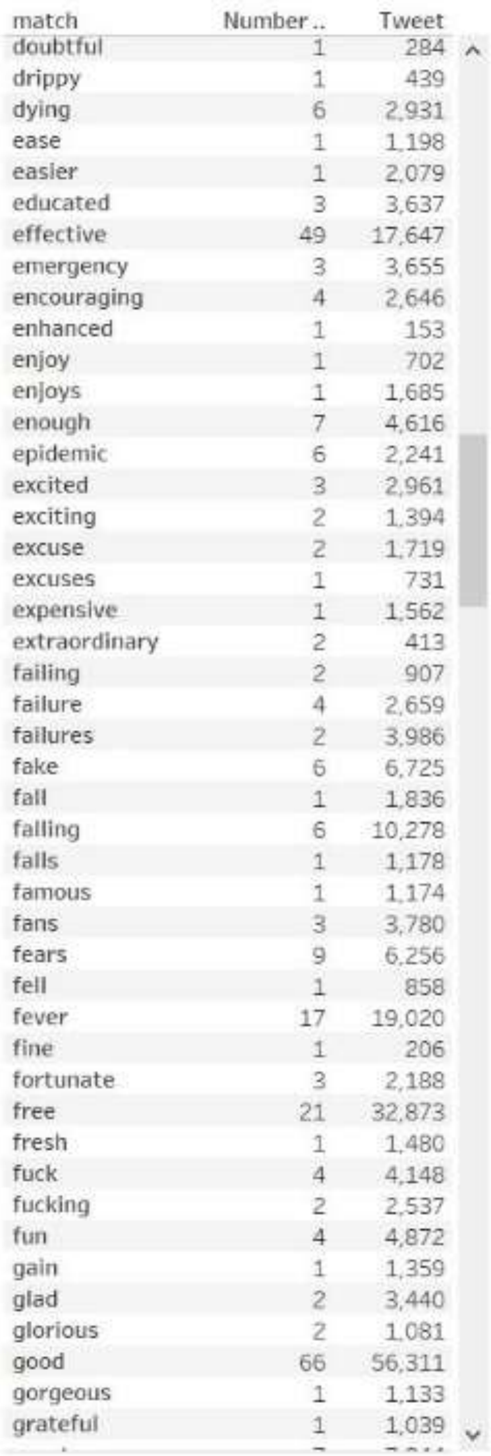


Figure 2 Number of records that are positive, neutral and negative tweets

Report 3:

Visualization of word that is matched single or multiple times in the respective tweet and the number of the tweet.



match	Number ..	Tweet
doubtful	1	284
drippy	1	439
dying	6	2,931
ease	1	1,198
easier	1	2,079
educated	3	3,637
effective	49	17,647
emergency	3	3,655
encouraging	4	2,646
enhanced	1	153
enjoy	1	702
enjoys	1	1,685
enough	7	4,616
epidemic	6	2,241
excited	3	2,961
exciting	2	1,394
excuse	2	1,719
excuses	1	731
expensive	1	1,562
extraordinary	2	413
failing	2	907
failure	4	2,659
failures	2	3,986
fake	6	6,725
fall	1	1,836
falling	6	10,278
falls	1	1,178
famous	1	1,174
fans	3	3,780
fears	9	6,256
fell	1	858
fever	17	19,020
fine	1	206
fortunate	3	2,188
free	21	32,873
fresh	1	1,480
fuck	4	4,148
fucking	2	2,537
fun	4	4,872
gain	1	1,359
glad	2	3,440
glorious	2	1,081
good	66	56,311
gorgeous	1	1,133
grateful	1	1,039

Figure 3 Table showing positive and negative words

Report 4:

Visualization of negative and positive words that appeared in tweets in circles in different sizes based on the times appeared in tweets.

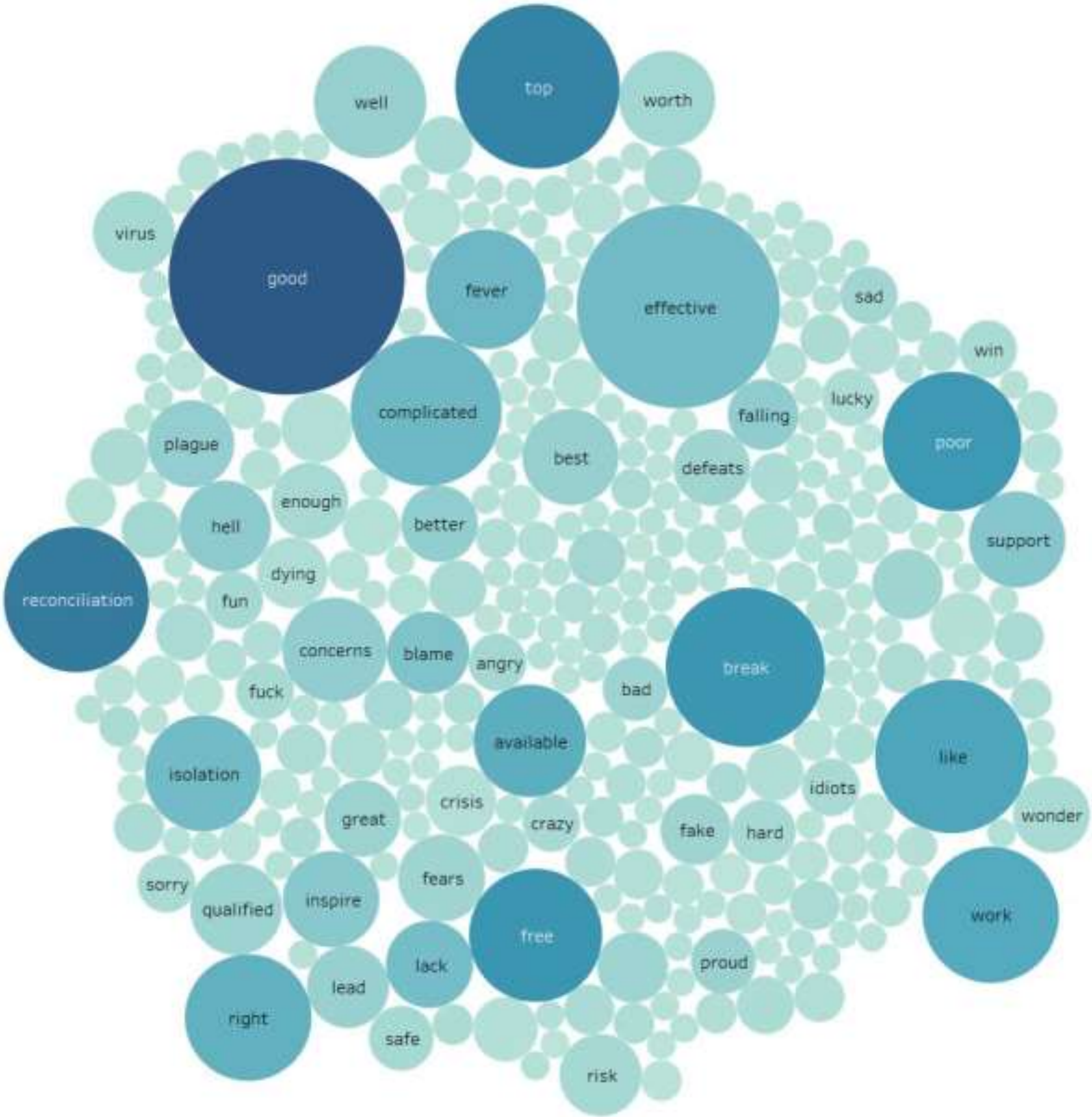


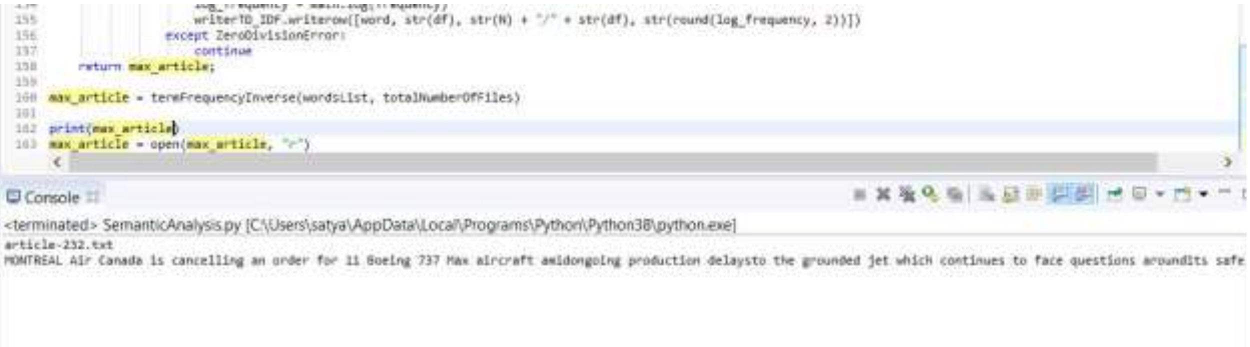
Figure 4 Positive and Negative words

B. Semantic Analysis

I have extracted the 500 news articles and cleaned the data using regular expressions. Initially, I have stores in the json format and converted into the csv format. I have stored the ‘Title’, ‘Content’, and ’Description’ of the news articles and further saved into the text files. Further, I used the search query for the words “Canada”, “University”, “Dalhousie University”, “Halifax”, “Business” by splitting the words to calculate TF-IDF (Term Frequency – Inverse Document Frequency). After splitting the words, I have calculated the number of documents, documents names that are having the keywords, number of times the particular word appeared in the document, log10(N/df) for the words that appeared and stored the words in the csv file (TF-IDF.csv)

Furthermore, I have also found the article that have maximum occurrences of the word ‘Canada’ compared to all the 500 articles. I have stored the details of the articles in the frequency_count.csv file and calculated the highest relative frequency by computing f/m in the python file (SemanticAnalysis.py).

The following figure is the screenshot that displays the article no and the content in the article that have maximum occurrences of the word ‘Canada’.



```
                log_frequency = math.log(frequency)
                writerTD_IDF.writerow([word, str(df), str(N) + "/" + str(df), str(round(log_frequency, 2))])
            except ZeroDivisionError:
                continue
    return max_article;

max_article = termFrequencyInverse(wordsList, totalNumberOfFiles)

print(max_article)
max_article = open(max_article, "r")
```

```
<terminated> SemanticAnalysis.py [C:\Users\satya\AppData\Local\Programs\Python\Python38\python.exe]
article-232.txt
MONTREAL Air Canada is cancelling an order for 11 Boeing 737 Max aircraft amidongoing production delaysto the grounded jet which continues to face questions aroundits safe
```

Figure 5 Highest Frequency Occurrence

References

[1]Satya Kumar Itekela, Assignment-3, submitted to CSCI-5408

[2]"Python: Count the occurrences of each word in a given sentence - w3resource", *w3resource*, 2020. [Online]. Available: https://www.w3resource.com/python-exercises/string/python-data-type-string-exercise-12.php. [Accessed: 10- Apr- 2020].

[3][Online]. Available: https://kb.tableau.com/articles/howto/creating-a-word-cloud. [Accessed: 10- Apr- 2020].

[4]Free Tableau Videos [Online]. Available: https://www.tableau.com/learn/training. [Accessed: 10- Apr- 2020].