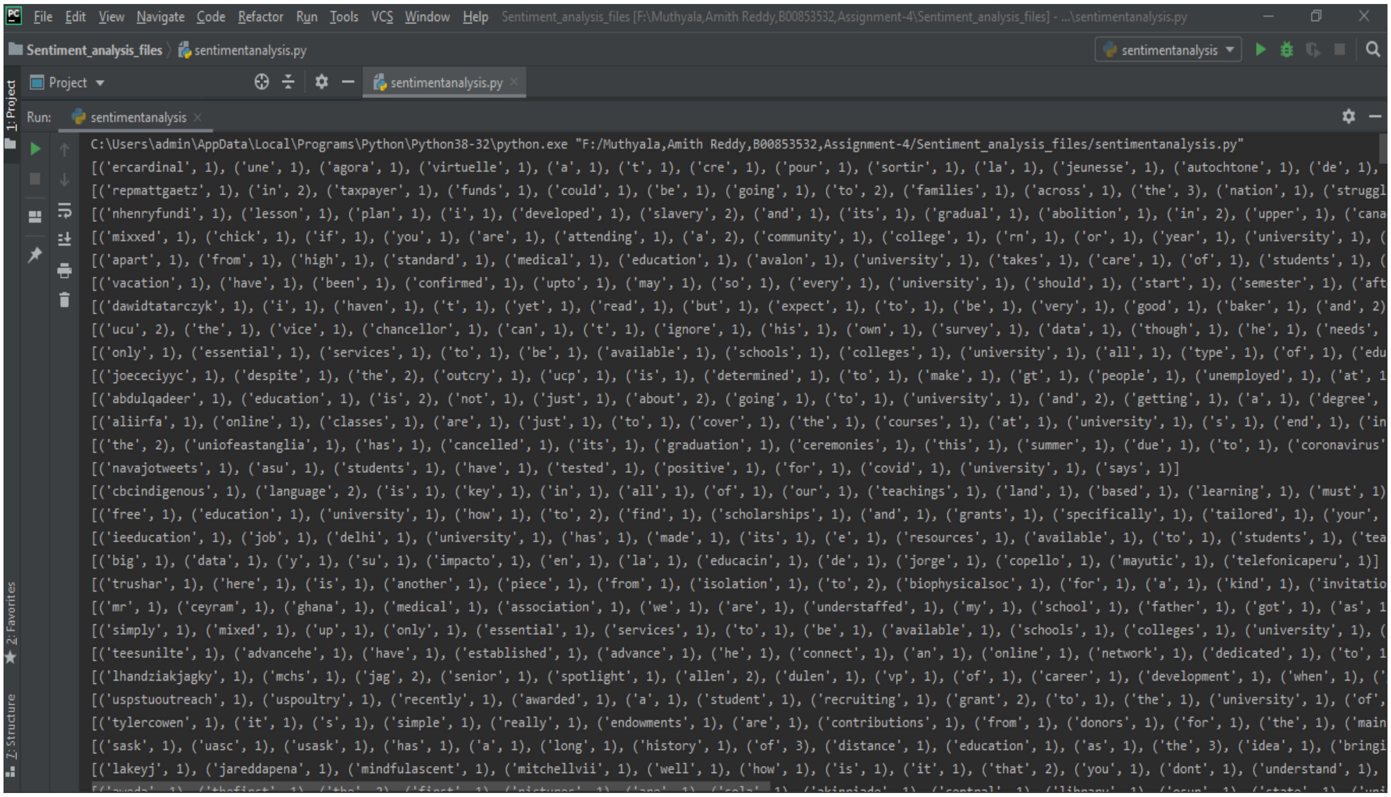
A. Sentiment Analysis:

1. I have considered the twitter data that I have obtained in previous assignment. I have included the python file to generate the tweets in the Sentiment Analysis folder with the file name as ‘twitter.py’. I have included the csv file as ‘tweets.csv’ which I have obtained in the last assignment. Now to consider only the tweets message, I wrote a python script ‘sentimentanalysis.py’ which considers only tweet messages from ‘tweets.csv’.

2. I wrote a function ‘clean’ to clean the twitter data. It was completed in the previous assignment. I have done cleaning tasks such as removing hashtags, special characters, URLs, and emoticons. This cleaning function is included in the ‘twitter.py’ file.

3. I wrote a script ‘sentimentanalysis.py’ which includes creation of bag-of-words for each tweet. I have removed all the duplicate tweets and generated a new csv file as ‘noduplicatetweetscsv.csv’. I have used this file which contains the tweets and successfully generated the bag-of-words and moved to the next step for the sentiment analysis. The below screenshot shows the bag-of-words for each tweet in a new line. The following bag-of-words can be generated by running ‘sentimentanalysis.py’


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Sentiment_analysis_files [F:\Muthyala,Amith Reddy,B00853532,Assignment-4\Sentiment_analysis_files] - ...\sentimentanalysis.py
Sentiment_analysis_files
sentimentanalysis.py
sentimentanalysis
Project
sentimentanalysis.py
Run:
sentimentanalysis
1: Project
2: Favorites
Z: Structure

```
C:\Users\admin\AppData\Local\Programs\Python\Python38-32\python.exe "F:/Muthyala,Amith Reddy,B00853532,Assignment-4/Sentiment_analysis_files/sentimentanalysis.py"
[('ercardinal', 1), ('une', 1), ('agora', 1), ('virtuelle', 1), ('a', 1), ('t', 1), ('cre', 1), ('pour', 1), ('sortir', 1), ('la', 1), ('jeunesse', 1), ('autochtone', 1), ('de', 1),
[('repmattgaetz', 1), ('in', 2), ('taxpayer', 1), ('funds', 1), ('could', 1), ('be', 1), ('going', 1), ('to', 2), ('families', 1), ('across', 1), ('the', 3), ('nation', 1), ('struggl
[('nhenryfundi', 1), ('lesson', 1), ('plan', 1), ('i', 1), ('developed', 1), ('slavery', 2), ('and', 1), ('its', 1), ('gradual', 1), ('abolition', 1), ('in', 2), ('upper', 1), ('cana
[('mixxed', 1), ('chick', 1), ('if', 1), ('you', 1), ('are', 1), ('attending', 1), ('a', 2), ('community', 1), ('college', 1), ('rn', 1), ('or', 1), ('year', 1), ('university', 1), (
[('apart', 1), ('from', 1), ('high', 1), ('standard', 1), ('medical', 1), ('education', 1), ('avalon', 1), ('university', 1), ('takes', 1), ('care', 1), ('of', 1), ('students', 1), (
[('vacation', 1), ('have', 1), ('been', 1), ('confirmed', 1), ('upto', 1), ('may', 1), ('so', 1), ('every', 1), ('university', 1), ('should', 1), ('start', 1), ('semester', 1), ('aft
[('dawidtatarczyk', 1), ('i', 1), ('haven', 1), ('t', 1), ('yet', 1), ('read', 1), ('but', 1), ('expect', 1), ('to', 1), ('be', 1), ('very', 1), ('good', 1), ('baker', 1), ('and', 2)
[('ucu', 2), ('the', 1), ('vice', 1), ('chancellor', 1), ('can', 1), ('t', 1), ('ignore', 1), ('his', 1), ('own', 1), ('survey', 1), ('data', 1), ('though', 1), ('he', 1), ('needs',
[('only', 1), ('essential', 1), ('services', 1), ('to', 1), ('be', 1), ('available', 1), ('schools', 1), ('colleges', 1), ('university', 1), ('all', 1), ('type', 1), ('of', 1), ('edu
[('joececiyyc', 1), ('despite', 1), ('the', 2), ('outcry', 1), ('ucp', 1), ('is', 1), ('determined', 1), ('to', 1), ('make', 1), ('gt', 1), ('people', 1), ('unemployed', 1), ('at', 1
[('abdulqadeer', 1), ('education', 1), ('is', 2), ('not', 1), ('just', 1), ('about', 2), ('going', 1), ('to', 1), ('university', 1), ('and', 2), ('getting', 1), ('a', 1), ('degree',
[('aliirfa', 1), ('online', 1), ('classes', 1), ('are', 1), ('just', 1), ('to', 1), ('cover', 1), ('the', 1), ('courses', 1), ('at', 1), ('university', 1), ('s', 1), ('end', 1), ('in
[('the', 2), ('uniofeastanglia', 1), ('has', 1), ('cancelled', 1), ('its', 1), ('graduation', 1), ('ceremonies', 1), ('this', 1), ('summer', 1), ('due', 1), ('to', 1), ('coronavirus'
[('navajotweets', 1), ('asu', 1), ('students', 1), ('have', 1), ('tested', 1), ('positive', 1), ('for', 1), ('covid', 1), ('university', 1), ('says', 1)]
[('cbcindigenous', 1), ('language', 2), ('is', 1), ('key', 1), ('in', 1), ('all', 1), ('of', 1), ('our', 1), ('teachings', 1), ('land', 1), ('based', 1), ('learning', 1), ('must', 1)
[('free', 1), ('education', 1), ('university', 1), ('how', 1), ('to', 2), ('find', 1), ('scholarships', 1), ('and', 1), ('grants', 1), ('specifically', 1), ('tailored', 1), ('your',
[('ieeducation', 1), ('job', 1), ('delhi', 1), ('university', 1), ('has', 1), ('made', 1), ('its', 1), ('e', 1), ('resources', 1), ('available', 1), ('to', 1), ('students', 1), ('tea
[('big', 1), ('data', 1), ('y', 1), ('su', 1), ('impacto', 1), ('en', 1), ('la', 1), ('educacin', 1), ('de', 1), ('jorge', 1), ('copello', 1), ('mayutic', 1), ('telefonicaperu', 1)]
[('trushar', 1), ('here', 1), ('is', 1), ('another', 1), ('piece', 1), ('from', 1), ('isolation', 1), ('to', 2), ('biophysicalsoc', 1), ('for', 1), ('a', 1), ('kind', 1), ('invitatio
[('mr', 1), ('ceyram', 1), ('ghana', 1), ('medical', 1), ('association', 1), ('we', 1), ('are', 1), ('understaffed', 1), ('my', 1), ('school', 1), ('father', 1), ('got', 1), ('as', 1
[('simply', 1), ('mixed', 1), ('up', 1), ('only', 1), ('essential', 1), ('services', 1), ('to', 1), ('be', 1), ('available', 1), ('schools', 1), ('colleges', 1), ('university', 1), (
[('teesunilte', 1), ('advancehe', 1), ('have', 1), ('established', 1), ('advance', 1), ('he', 1), ('connect', 1), ('an', 1), ('online', 1), ('network', 1), ('dedicated', 1), ('to', 1
[('lhandziakjagky', 1), ('mchs', 1), ('jag', 2), ('senior', 1), ('spotlight', 1), ('allen', 2), ('dulen', 1), ('vp', 1), ('of', 1), ('career', 1), ('development', 1), ('when', 1), ('
[('uspstuoutreach', 1), ('uspoultry', 1), ('recently', 1), ('awarded', 1), ('a', 1), ('student', 1), ('recruiting', 1), ('grant', 2), ('to', 1), ('the', 1), ('university', 1), ('of',
[('tylercowen', 1), ('it', 1), ('s', 1), ('simple', 1), ('really', 1), ('endowments', 1), ('are', 1), ('contributions', 1), ('from', 1), ('donors', 1), ('for', 1), ('the', 1), ('main
[('sask', 1), ('uasc', 1), ('usask', 1), ('has', 1), ('a', 1), ('long', 1), ('history', 1), ('of', 3), ('distance', 1), ('education', 1), ('as', 1), ('the', 3), ('idea', 1), ('bringi
[('lakeyj', 1), ('jareddapena', 1), ('mindfulascent', 1), ('mitchellvii', 1), ('well', 1), ('how', 1), ('is', 1), ('it', 1), ('that', 2), ('you', 1), ('dont', 1), ('understand', 1),
[('aweda', 1), ('thefirst', 1), ('the', 2), ('first', 1), ('pictures', 1), ('are', 1), ('cola', 1), ('akinniade', 1), ('central', 1), ('library', 1), ('osun', 1), ('state', 1), ('uni
```

4. I downloaded the positive words and negative words from an online source which they have provided for general research usage [1]. The files are included in the Sentiment Analysis folder with file names as ‘positive-words.txt’ [3] and ‘negative-words.txt’ [4]. After obtaining

the positive and negative words, I have compared the bag-of-words with these positive and negative words and found the matches. The code for this is included in Sentiment Analysis folder as ‘sentimentanalysis.py’.

5. The next step is to tag each “positive”, “negative”, or “neutral”. First, I have initialized a sum variable to 0. Then, I have incremented the sum by 1 for each positive word and decremented the sum by 1 for each negative word. Now the sum variable contains the final result which helps in deciding about that particular tweet. When the sum is greater than 0, the tweet is tagged as “positive”. The tweet is “neutral” if the sum is equal to 0. Further when the sum is less than 0, the tweet is tagged as “negative”. The code for this is included in Sentiment Analysis folder as ‘sentimentanalysis.py’.

6. For the visualization purpose, I wrote a script ‘visualizationwords.py’ to generate ‘visualization2.csv’ which contains columns such as ‘Tweet_id’, ‘match_word’, ‘match_word_count’, ‘word_polarity’. I have downloaded and installed Tableau from the Internet. Then, the ‘visualization2.csv’ file is loaded into the Tableau Workbook. The ‘match_word_count’ column is the measure which is loaded as size and column value. The ‘word_polarity’ and ‘match_word’ columns are dimensions which are dragged onto the rows for the visualization.

   I have done 11 visualizations on the Tableau Workbook and included all the screenshots and visualization sheets in Assignment folder. I found this Visualization tool very helpful to analyse the most frequently occurring words in the positive and negative tweets which I have obtained.
   The first visualization is a dashboard that has three charts which clearly shows the most frequently occurring positive word and negative word. The below screenshot represents the first visualization.

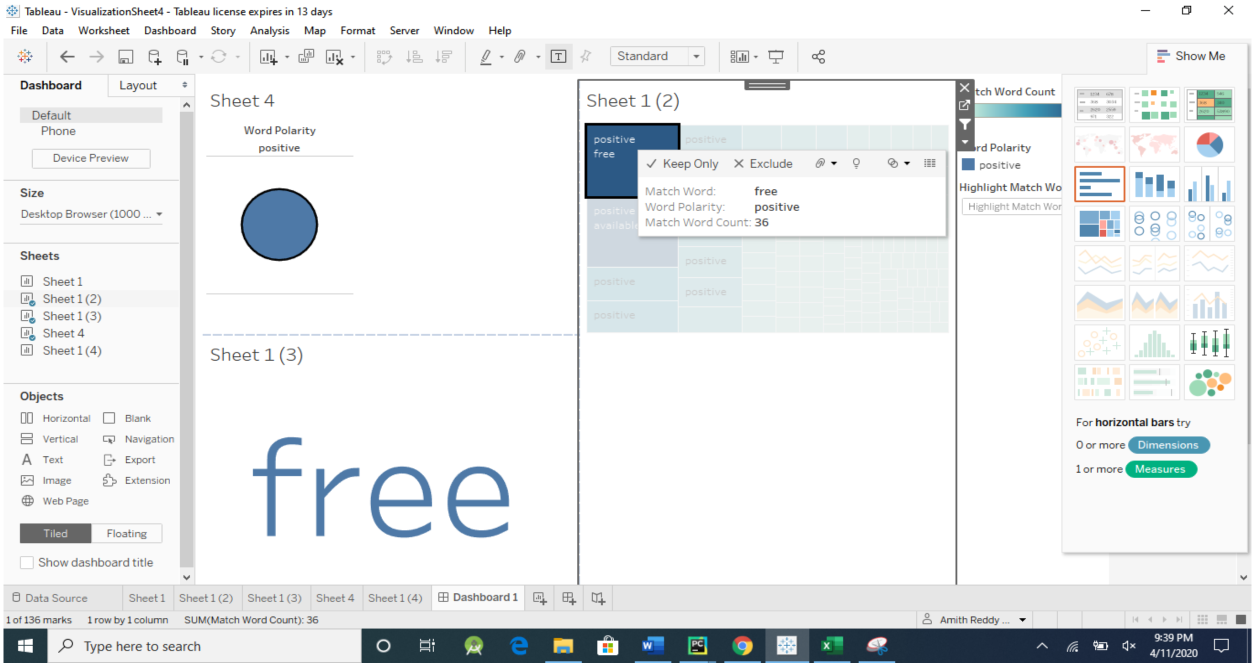


Tableau - VisualizationSheet4 - Tableau license expires in 13 days

Match Word: free
Word Polarity: positive
Match Word Count: 36

1 of 136 marks 1 row by 1 column SUM(Match Word Count): 36

The second visualization shows blocks of positive words and negative words. The size of the block tells us about the words which are most frequent and less frequent in the tweets. The highest occurring positive word is “free” with count 35 and highest occurring negative word is “break” with count 9.

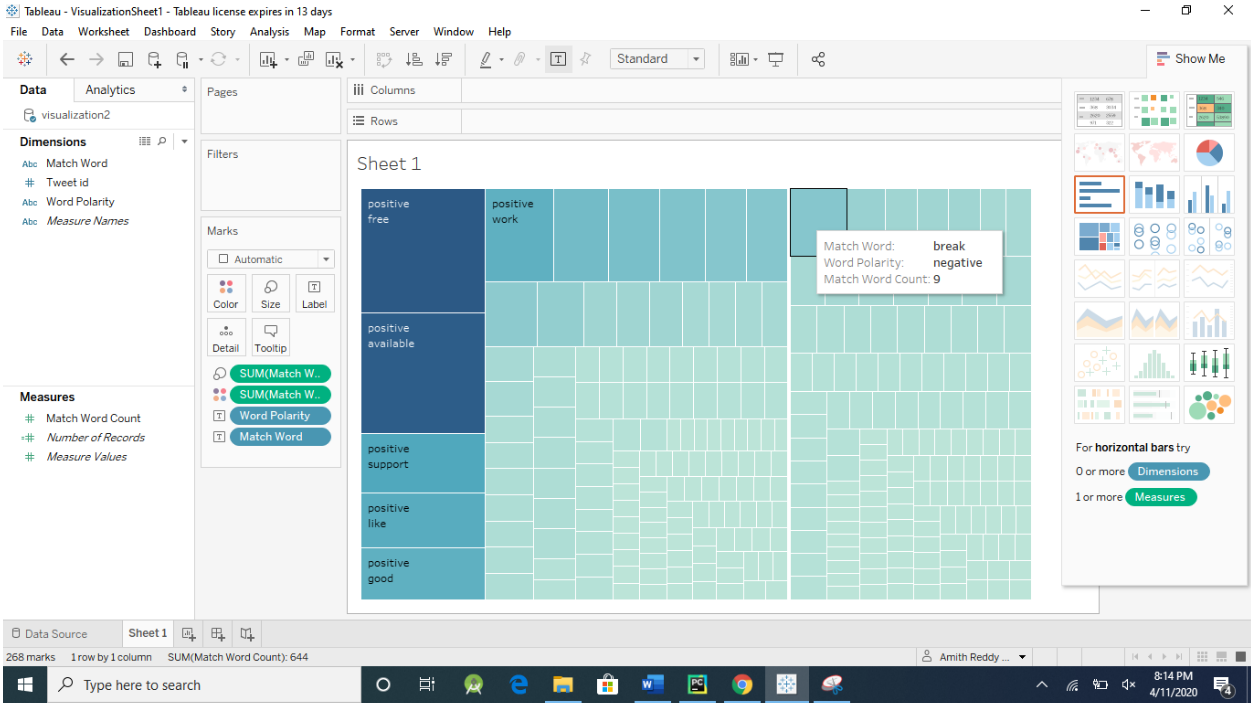


Tableau - VisualizationSheet1 - Tableau license expires in 13 days

Match Word: break
Word Polarity: negative
Match Word Count: 9

268 marks 1 row by 1 column SUM(Match Word Count): 644

The third visualization helps us in visualizing the most frequently occurring positive and negative words in form of a word cloud. When we hover over the words, it displays the word count of the word and word polarity which is extremely useful for analysing the tweets further.

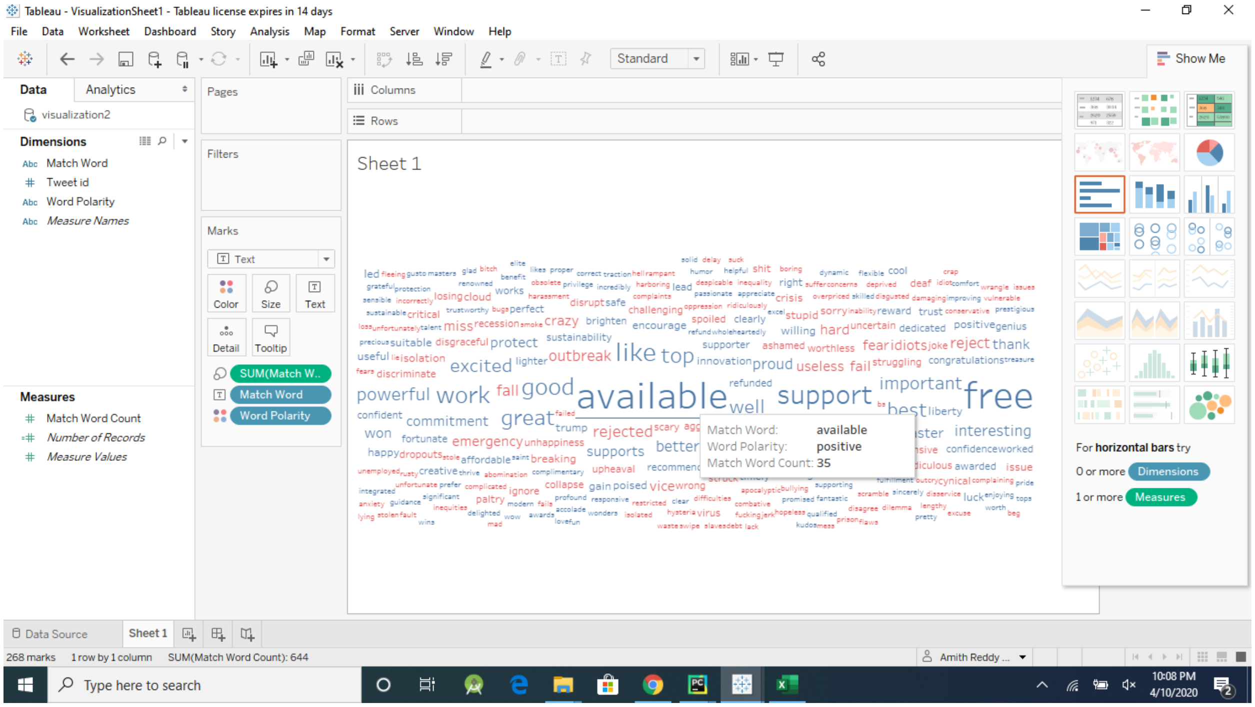
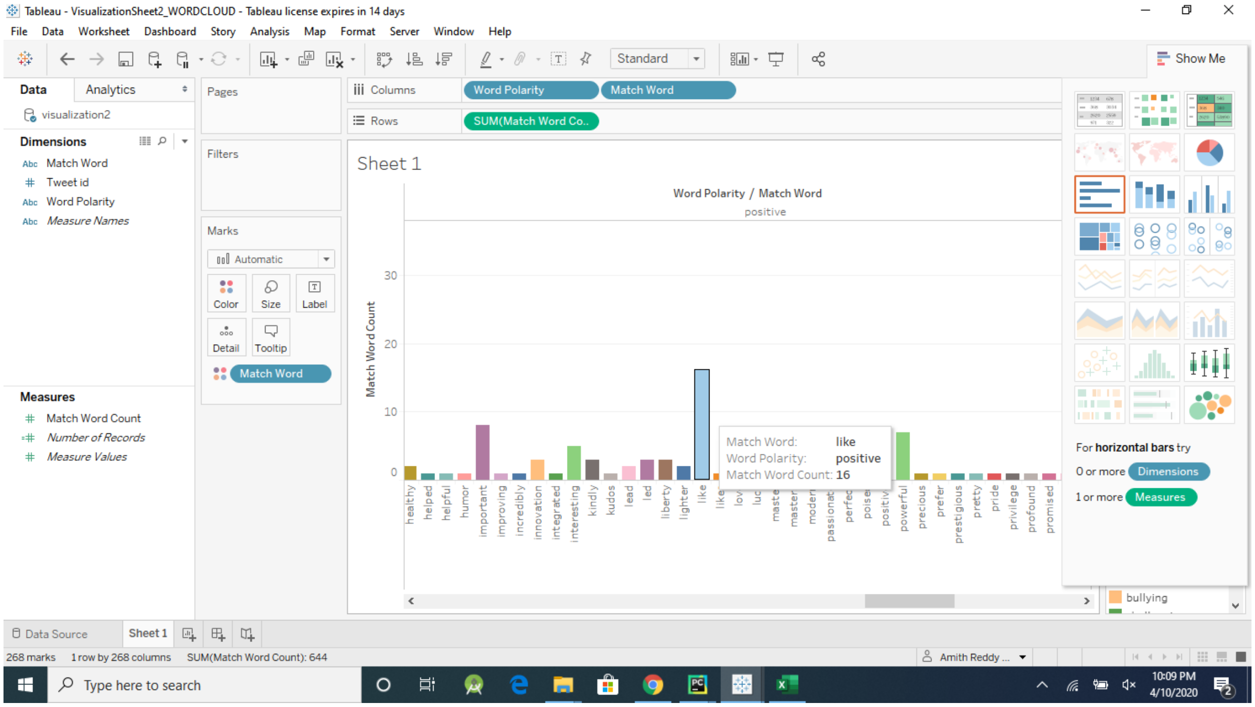


Tableau - VisualizationSheet1 - Tableau license expires in 14 days

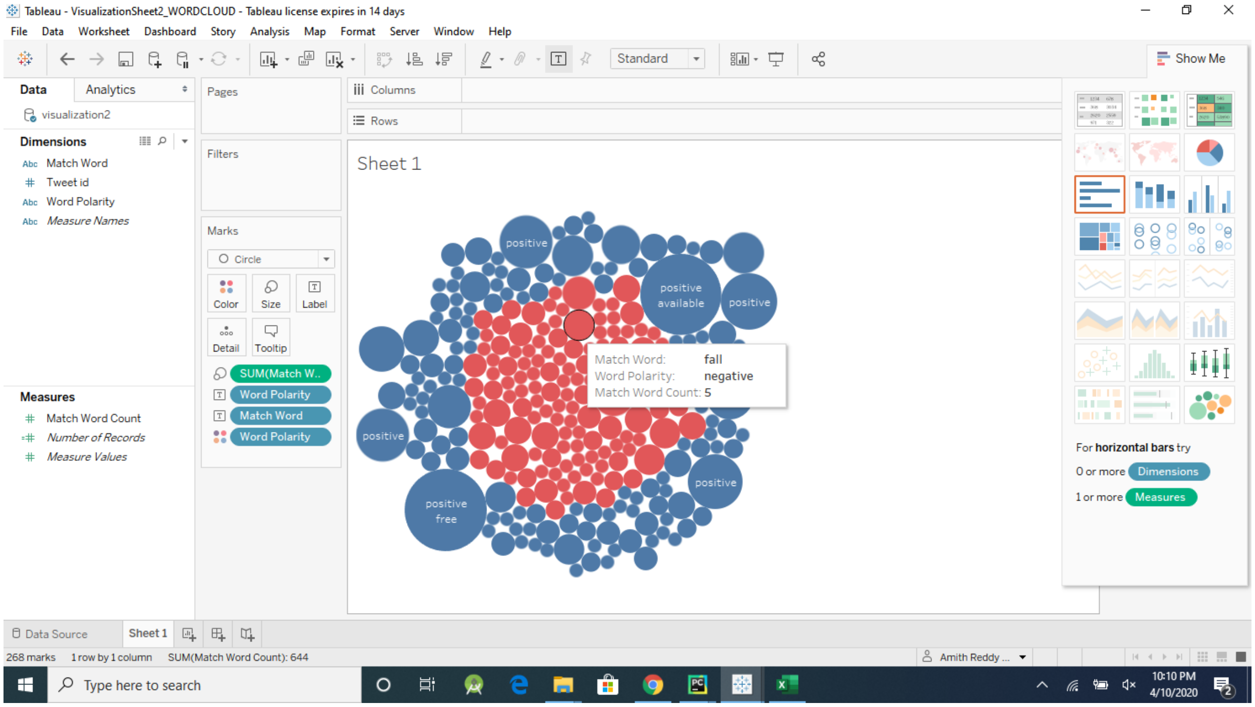
Match Word: available
Word Polarity: positive
Match Word Count: 35

268 marks 1 row by 1 column SUM(Match Word Count): 644

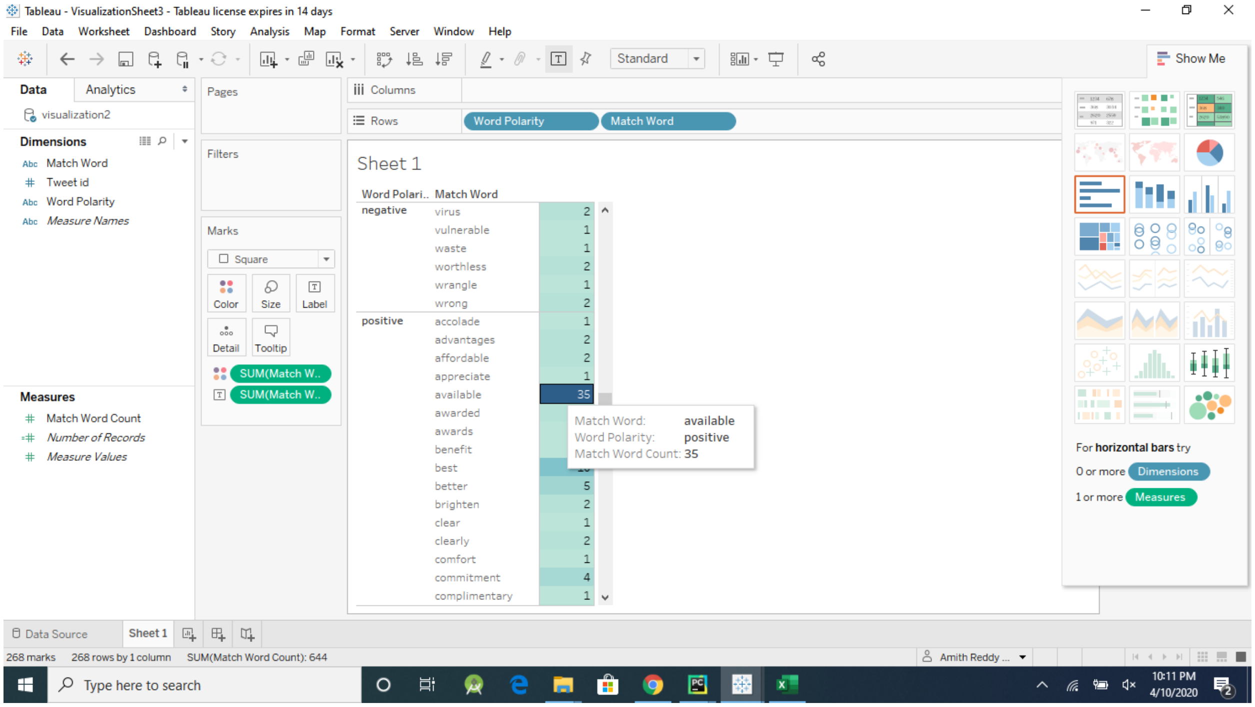
The fourth visualization is a bar graph which shows list of positive words and negative words along with their frequency count.



The fifth visualization is a bubble graph which clearly differentiates positive and negative words with blue and red colour. It also displays the count of each word.



The sixth visualization is a scroll sheet where you can scroll to check all the positive words and negative words along with their count.



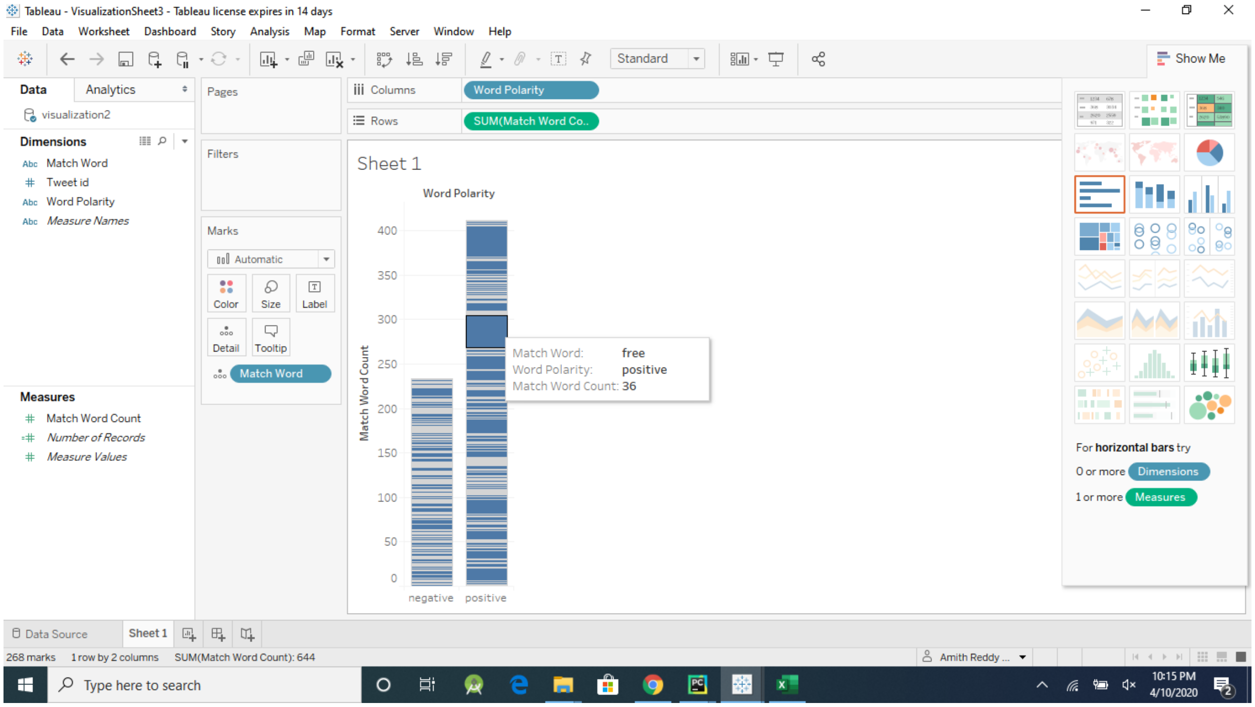
Sheet 1

Word Polari..	Match Word	
negative	virus	2
	vulnerable	1
	waste	1
	worthless	2
	wrangle	1
	wrong	2
positive	accolade	1
	advantages	2
	affordable	2
	appreciate	1
	available	35
	awarded	[illegible]
	awards	[illegible]
	benefit	[illegible]
	best	[illegible]
	better	5
	brighten	2
	clear	1
	clearly	2
	comfort	1
	commitment	4
	complimentary	1

Match Word: available
Word Polarity: positive
Match Word Count: 35

268 marks 268 rows by 1 column SUM(Match Word Count): 644

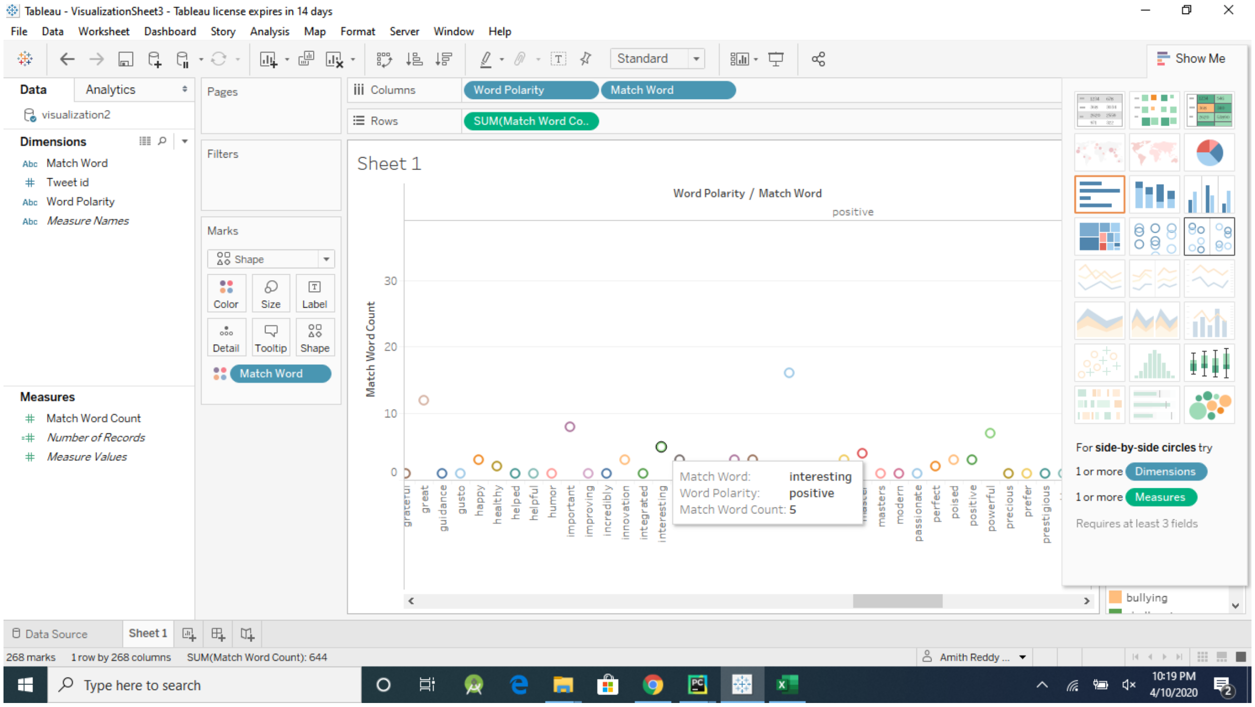
The seventh visualization shows the word count of all positive and negative words. It clearly shows that the total positive words count is nearly double the total negative words count.



Match Word: free
Word Polarity: positive
Match Word Count: 36

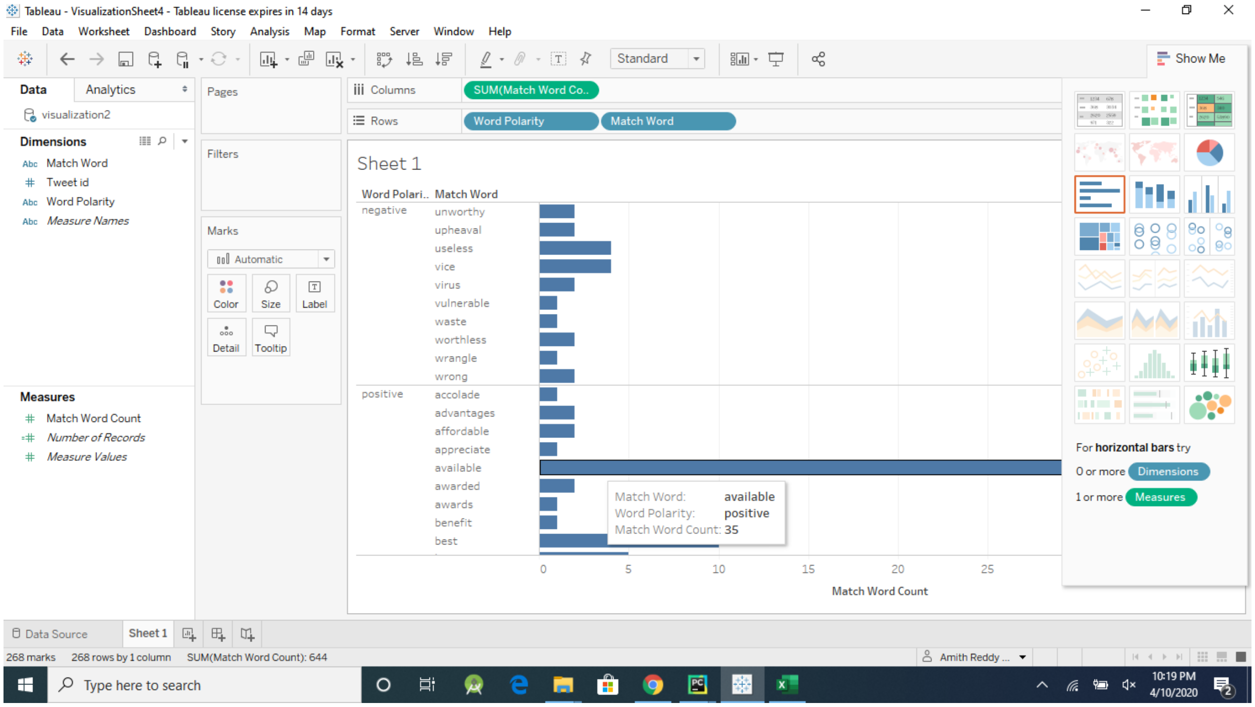
268 marks 1 row by 2 columns SUM(Match Word Count): 644

The eighth visualization is almost similar to a bar graph. The only difference is that it shows the count of positive words and negative words in form of a bubble. It looks very clean and precise.



Match Word: interesting
Word Polarity: positive
Match Word Count: 5

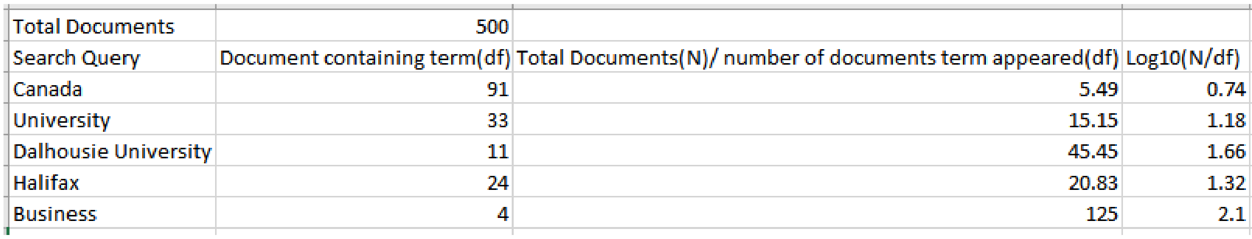
The nineth visualization shows a segregation between positive words and negative words in the form of a tilted bar graph.



Match Word: available
Word Polarity: positive
Match Word Count: 35

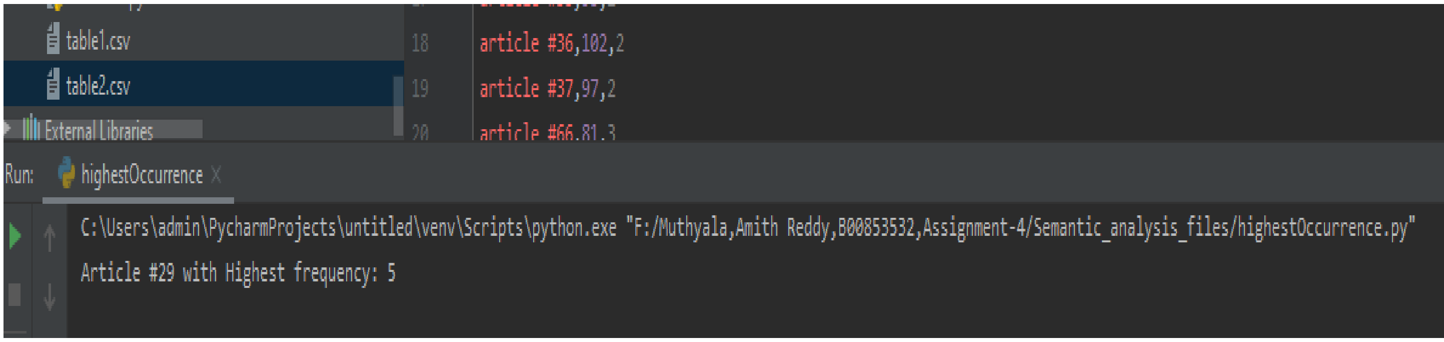
B. Semantic Analysis:

7. I wrote a well-formed script to clean and transform the news articles in python. I have stored these news articles in a file named ‘news.csv’. The script to generate this ‘news.csv’ is included in the Semantic Analysis files folder as ‘newsExtraction.py’. I have removed URLs, special characters, and emoticons.
8. I have considered each chunk of text as a document. I wrote a script ‘semantic.py’ to generate each news article as a separate csv document. All the news articles are named as ‘file’ followed by numbers that are in a sequence. These files only contain one news article per document.
9. Each news file only contains “title”, “description”, and the news “content”. The script to generate these required columns is included as ‘semantic.py’. The eighth and nineth steps are combinedly done to maintain the flow.
10. COMPUTE TF-IDF (term frequency-inverse document frequency)
    a. I have stored all the news articles in different csv files. I have included these files in the Semantic Analysis folder. In my case N=500. Now, I have used the search query “Canada”, “University”, “Dalhousie University”, “Halifax”, “Business”, and searched in how many documents these words have appeared. The python script for this searching is included as ‘search.py’. It takes the search query as specified and creates a table ‘table1.csv’ with columns “Search Query”, “Document containing df”, “N/df”, “log10(N/df)”. The below screenshot displays the number of documents which contain the search query. I have included the screenshot as ‘table-1’ and the csv file ‘table-1.csv’ is included in the Semantic Analysis folder.



Total Documents	500		
Search Query	Document containing term(df)	Total Documents(N)/ number of documents term appeared(df)	Log10(N/df)
Canada	91	5.49	0.74
University	33	15.15	1.18
Dalhousie University	11	45.45	1.66
Halifax	24	20.83	1.32
Business	4	125	2.1

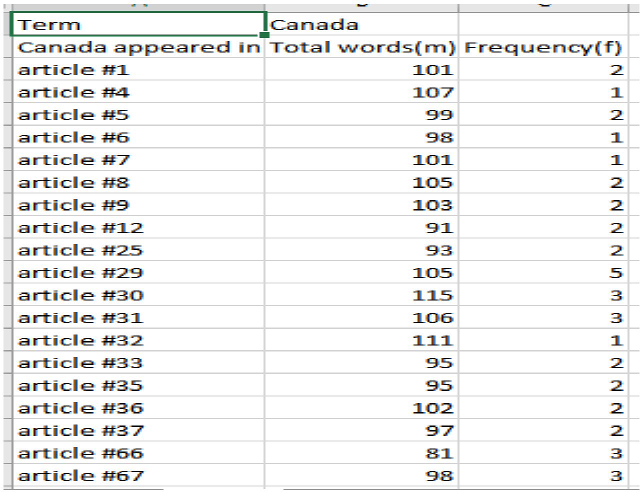
    b. The next step is to find the document which has the highest occurrence of the word “Canada”. This can be done by performing frequency count of the word per document. I wrote a python script ‘highestOccurrence.py’ to generate ‘table-2.csv’ and also displays the document which has the highest occurrence of the word “Canada”. The below screenshot displays the article with highest occurrence of the word “Canada”.



```
article #36,102,2
article #37,97,2
article #66,81,3
```

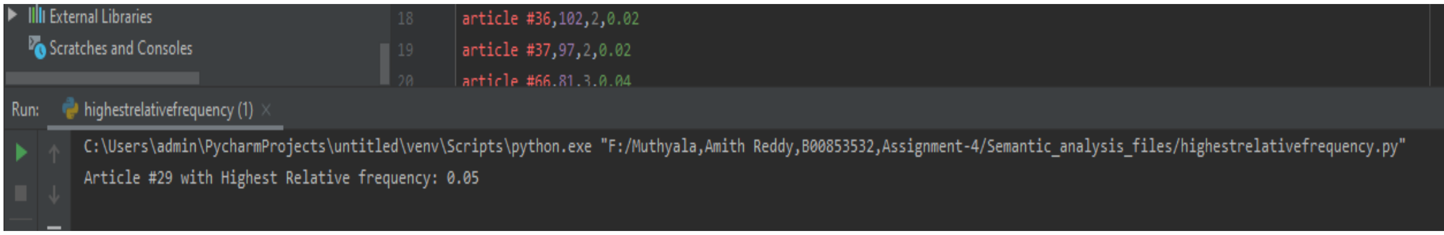
```
C:\Users\admin\PycharmProjects\untitled\venv\Scripts\python.exe "F:/Muthyala,Amith Reddy,B00853532,Assignment-4/Semantic_analysis_files/highestOccurrence.py"
Article #29 with Highest frequency: 5
```

The below table shows frequency count of the word “Canada” per document.



Term	Canada	
Canada appeared in	Total words(m)	Frequency(f)
article #1	101	2
article #4	107	1
article #5	99	2
article #6	98	1
article #7	101	1
article #8	105	2
article #9	103	2
article #12	91	2
article #25	93	2
article #29	105	5
article #30	115	3
article #31	106	3
article #32	111	1
article #33	95	2
article #35	95	2
article #36	102	2
article #37	97	2
article #66	81	3
article #67	98	3

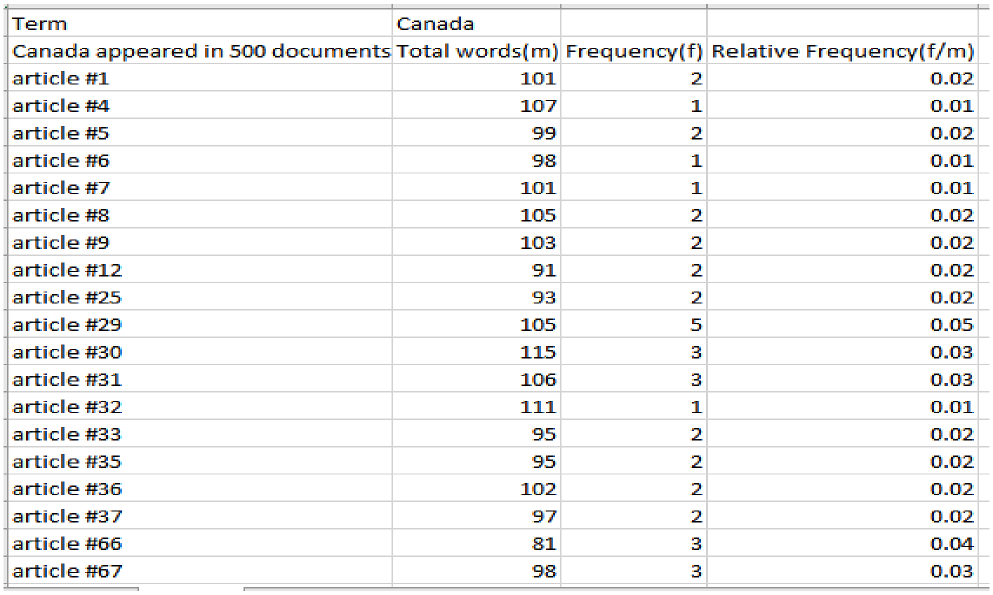
c. To find out the article with highest relative frequency, I wrote a well-formed python script ‘highestrelativefrequency.py’ which is included in the Semantic Analysis folder. This python file generates a table ‘table-3.csv’ which has relative frequency values per document. The below screenshot displays the article with highest relative frequency (f/m).



```
article #36,102,2,0.02
article #37,97,2,0.02
article #66,81,3,0.04
```

```
C:\Users\admin\PycharmProjects\untitled\venv\Scripts\python.exe "F:/Muthyala,Amith Reddy,B00853532,Assignment-4/Semantic_analysis_files/highestrelativefrequency.py"
Article #29 with Highest Relative frequency: 0.05
```

The below table shows the relative frequency of news articles.



Term	Canada		
Canada appeared in 500 documents	Total words(m)	Frequency(f)	Relative Frequency(f/m)
article #1	101	2	0.02
article #4	107	1	0.01
article #5	99	2	0.02
article #6	98	1	0.01
article #7	101	1	0.01
article #8	105	2	0.02
article #9	103	2	0.02
article #12	91	2	0.02
article #25	93	2	0.02
article #29	105	5	0.05
article #30	115	3	0.03
article #31	106	3	0.03
article #32	111	1	0.01
article #33	95	2	0.02
article #35	95	2	0.02
article #36	102	2	0.02
article #37	97	2	0.02
article #66	81	3	0.04
article #67	98	3	0.03

References:

[1] M. Hu and B. Liu, “Mining and summarizing customer reviews,” Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’04, 2004 [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1014052.1014073. [Accessed: 11-Apr-2020]

[2] B. Liu, M. Hu, and J. Cheng, “Opinion observer,” Proceedings of the 14th international conference on World Wide Web - WWW ’05, 2005 [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1060745.1060797. [Accessed: 11-Apr-2020]

[3] *Ptrckprry.com*, 2010. [Online]. Available: http://ptrckprry.com/course/ssd/data/positive-words.txt. [Accessed: 11-Apr-2020].

[4] *Ptrckprry.com*, 2010. [Online]. Available: http://ptrckprry.com/course/ssd/data/negative-words.txt. [Accessed: 11-Apr-2020].

[5] “Natural Language Processing - Semantic Analysis - Tutorialspoint,” *Tutorialspoint.com*, 2020. [Online]. Available: https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_semantic_analysis.htm. [Accessed: 11-Apr-2020].

[6] Abdullah Alsaeedi and Mohammad Zubair Khan, “A Study on Sentiment Analysis Techniques of Twitter Data,” *ResearchGate*, 28-Feb-2019. [Online]. Available: https://www.researchgate.net/publication/331411860_A_Study_on_Sentiment_Analysis_Techniques_of_Twitter_Data. [Accessed: 11-Apr-2020].