LAB2: DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION

JAYANTH KADALIPURA SHIVALINGAIAH
50290276

ANIRUDH MANJUNATH
50289832

Summary

In this project, we will explore big data analytics by combining the skills of data exploration. We use Hadoop, AWS, Tableau to implement map reduce and visualize the output. We collect data from various sources like Twitter, New York Times and data from Common Crawl.

Common Crawl crawls through millions of web pages, stores them in AWS. It usually completes the crawl every month. The data collected is stored in the form of archives which is available to the public for free. The indexing helps us to get the data of the domain that we require.

We use Amazon EMR to run map reduce functionalities for this project. Amazon EMR is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data.

Project Goals

We have to choose a category of our interest and five subtopics in that category. We then collect all the data relevant to those five subtopics. The collected data is passed to map reduce to obtain word count and word co-occurrences. This is visualized using Tableau.

Procedure

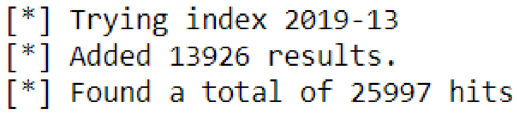
We chose Technology as our category. The five subtopics we chose were Amazon, Microsoft, Uber, Google and Facebook.

Step 1:
We collected data from Twitter using the subtopics as our keywords. We collected about 20,000 tweets, roughly 4,000 from each subtopic. We used searchTwitter method in TwitteR library in R to obtain the tweets. To do this, we have created **Datacollection_Twitter.ipynb** program in R, which collects and stores the data locally.

Step 2:
We proceeded to collect data from New York Times. We used the NYT API to obtain about 1400 articles. The API cannot stream the articles continuously and hence a sleep module, which halts the program for about 6 seconds is used. We have created **datacollection_nytimes.ipynb** in Python 2 to execute this.

Step 3:
In this step, we collect data from common crawl. We chose the Wall Street Journal as our domain since our subtopics are mostly related to business. To collect the data, we first have to choose from which index we want the data to be. We only get the data till March 2019.

We wrote a program which iterates through the archive of the three months of 2019 and return all the links belonging to the Wall Street Journal domain. We store the links in a text file. Our program iterates through the list of websites, visits each site and checks the headlines. If the headlines contain any of our subtopics, we collect the entire article and store it in text files. The data of each subtopic is stored in different files. We also have another text file which contains all the data combined. The Python 2 program **Datacollection_CommonCrawl.ipynb** executes this.



```
[*] Trying index 2019-13
[*] Added 13926 results.
[*] Found a total of 25997 hits
```

Collecting the links of WSJ domain

Step 4:
To apply map reduce, we first have to pre process the data. In the preprocessing stage, we remove all the stop words from the collected data. We also use regular expression to remove non alphabet characters. We also stemmed the data using Snowball Stemmer in NLTK. The result in the end will have all the important words which are relevant to our project. **Data_clean_stemming.ipynb** performs all these operations.

Step 5:
We use Amazon EMR to apply map reduce to get word count and word co-occurrence. We write different mappers and reducers for word count and cooccurrence.
We create Hadoop clusters by manually configuring the input file mapper and reducer functions. The outputs of these functions are stored in S3 buckets which can be downloaded as CSV files. There are **mapper.py** and **reducer.py** for word count and **mapperocc.py** for word cooccurrence in Python 3 to be executed by Amazon EMR.

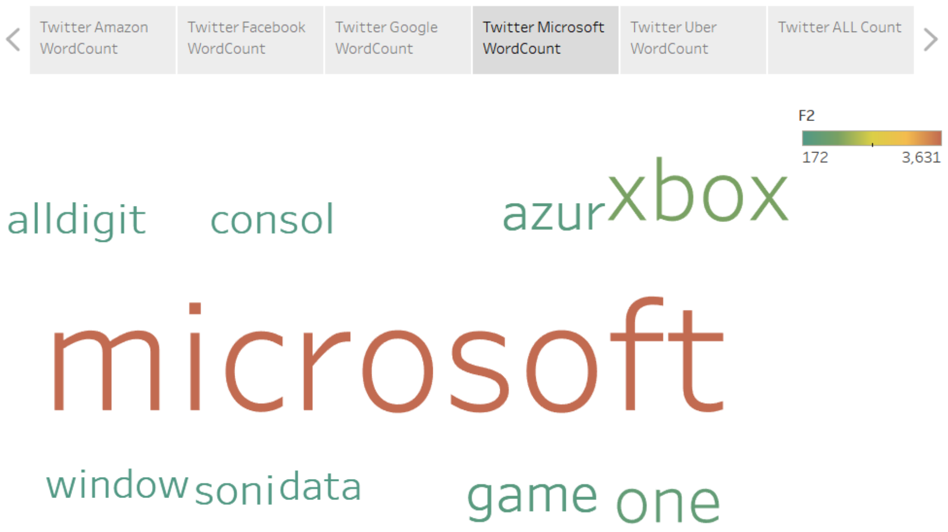
Step 6:
The results are visualized using Tableau. We pass the Excel files and visualize the top ten words in word count of each of the file in the form of Word cloud. We will have 6 stories, each containing the results of all the subtopics and the combined data of all three sources.

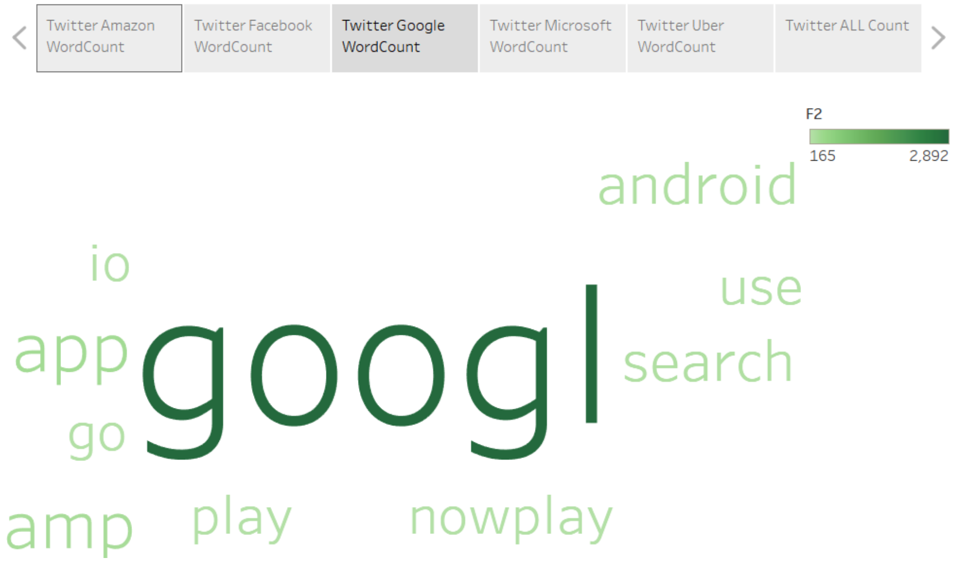
Outputs

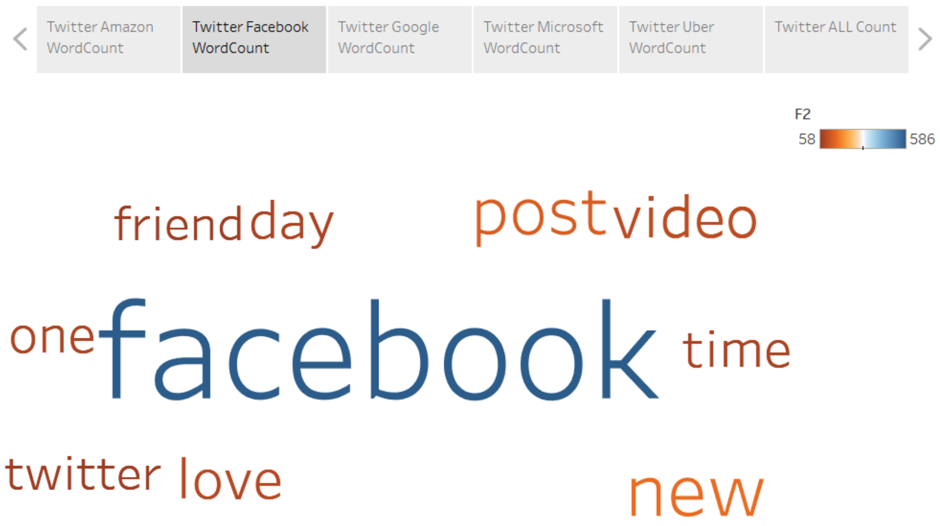
We obtain text files which contain different sets of data. We create a new set of text files when we preprocess it which are passed to map reduce. When we visualize all the data, we get a different wordclouds, each representing the type of data and the type of map reduce function performed on it.

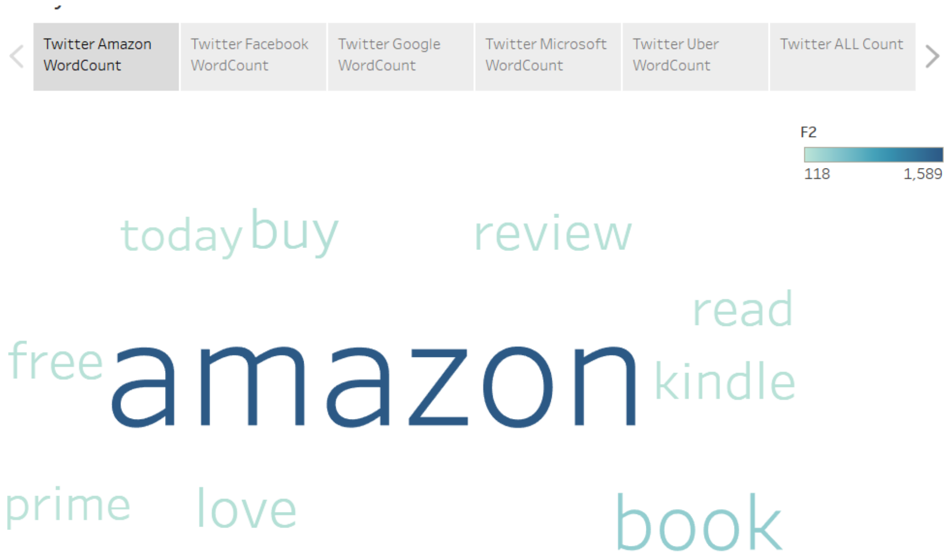
We have published the visualizations/workbook on Tableau online. We did not find the email ids of TAs to share with.

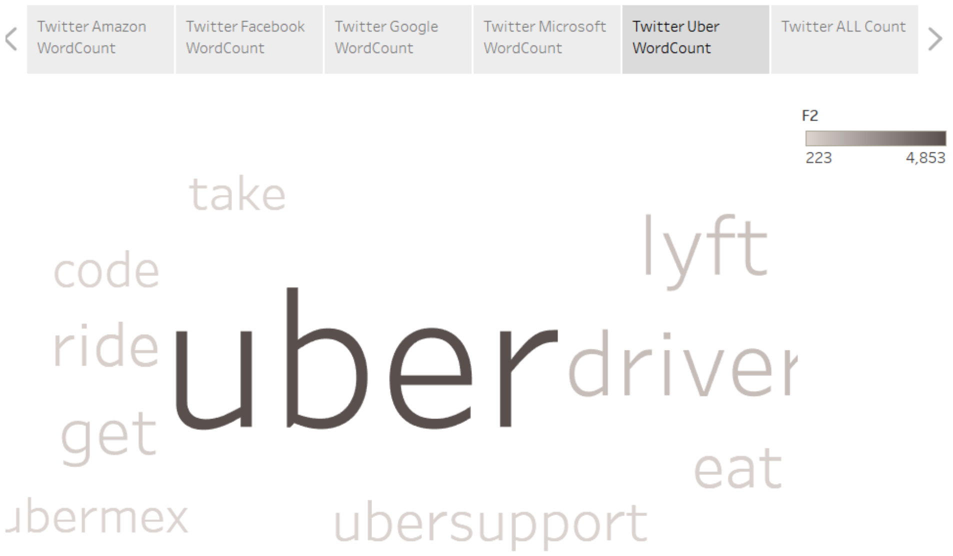
Below are the outputs of word cloud for Twitter data word count.individual subtopic


Twitter Amazon WordCount
Twitter Facebook WordCount
Twitter Google WordCount
Twitter Microsoft WordCount
Twitter Uber WordCount
Twitter ALL Count
F2
172
3,631
alldigit
consol
azur
xbox
microsoft
window
sonidata
game
one

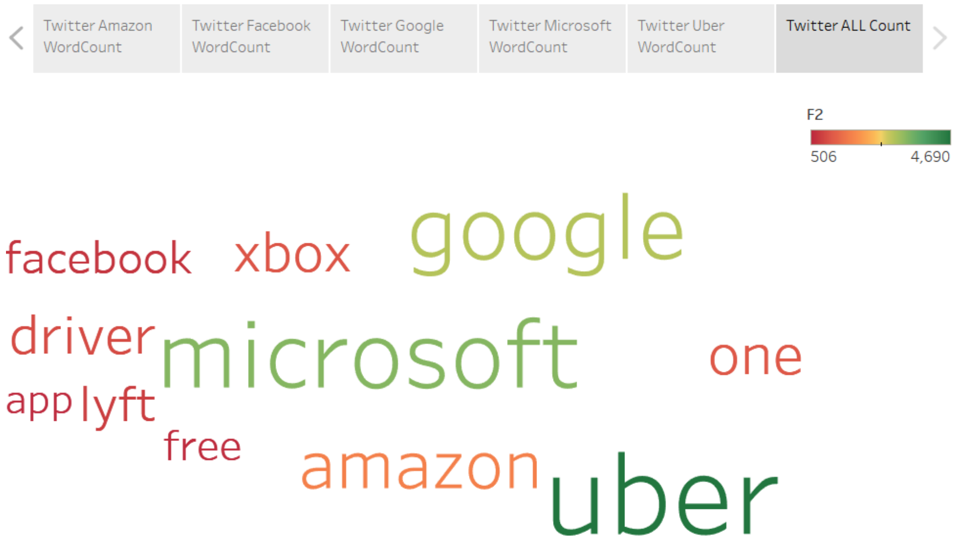

Twitter Amazon WordCount
Twitter Facebook WordCount
Twitter Google WordCount
Twitter Microsoft WordCount
Twitter Uber WordCount
Twitter ALL Count
F2
165
2,892
android
io
use
app
googl
search
go
amp
play
nowplay


Twitter Amazon WordCount
Twitter Facebook WordCount
Twitter Google WordCount
Twitter Microsoft WordCount
Twitter Uber WordCount
Twitter ALL Count
F2
58
586
friend
day
post
video
one
facebook
time
twitter
love
new


Twitter Amazon WordCount
Twitter Facebook WordCount
Twitter Google WordCount
Twitter Microsoft WordCount
Twitter Uber WordCount
Twitter ALL Count
F2
118
1,589
today
buy
review
read
free
amazon
kindle
prime
love
book



The below image is the visualization of all topics combined for word count


Twitter Amazon WordCount
Twitter Facebook WordCount
Twitter Google WordCount
Twitter Microsoft WordCount
Twitter Uber WordCount
Twitter ALL Count
F2
506
4,690
facebook
xbox
google
driver
microsoft
one
app
lyft
free
amazon
uber