DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION

Team Members:

Pratibha Arjun Barsale (pbarsale@buffalo.edu) - 50247005

Venkata Krishnan Anantha Raman (va34@buffalo.edu) - 50246287

Abstract:

This project aims at analyzing Data scrapped from the Internet from sources like New York Times and Twitter using Word Count Map-Reduce Algorithm. The data is presented using D3 JS. The pipeline is built in a way so that it can analyze any topic of Interest. We picked the word "Facebook" and did the analysis.

The Word Count Map Reduce Algorithm was run on data scrapped from social media sources(Twitter) and News Sources (New York Times). Using the NY Times API, about 9K Articles were collected and about 1.5 Million tweets were gathered using the Twitter API. This data was scrapped to get text data from the web-pages and tweets. The data was fed into the Map Reduce Engine and the word-counts were collected. The Mapper employs stop words and regular expression filtering. The top words from the word-count is displayed using D3JS. Building on this, Another Map Reduce Engine was built that analyses word co-occurrence.

Approach:

Part 1:

Ran multiple python commands from Part3/Part4/Part5.

Part 2:

Data Collection:

1) API Key was created for NY Times. Using the Collect API, articles were collected.
2) Created twitter API Key and obtained tweets using twitteR API.
3) These URLs collected from the NY Times were scrapped and the contents were stored to a file. The tweet content is also stored to the file.

Word Count Mapper:

1) The Mapper employs regular expression filters and stop-words filter from ntlk library.
2) The data is fed on to the Mapper. The Mapper emits <key 1>

Word Count Reducer:

1) The <key 1> values are sorted and fed into the Reducer. The Reducer emits <key total-count>. This will help us see the most frequent words.

Word Count Visualization:

1) Used D3JS to display the frequent words obtained by the WordCount Map Reduce Algorithm.

Co-occurrence Mapper:

1) Using the previous knowledge of the frequent word counts. Emit <word1,word2 1>.

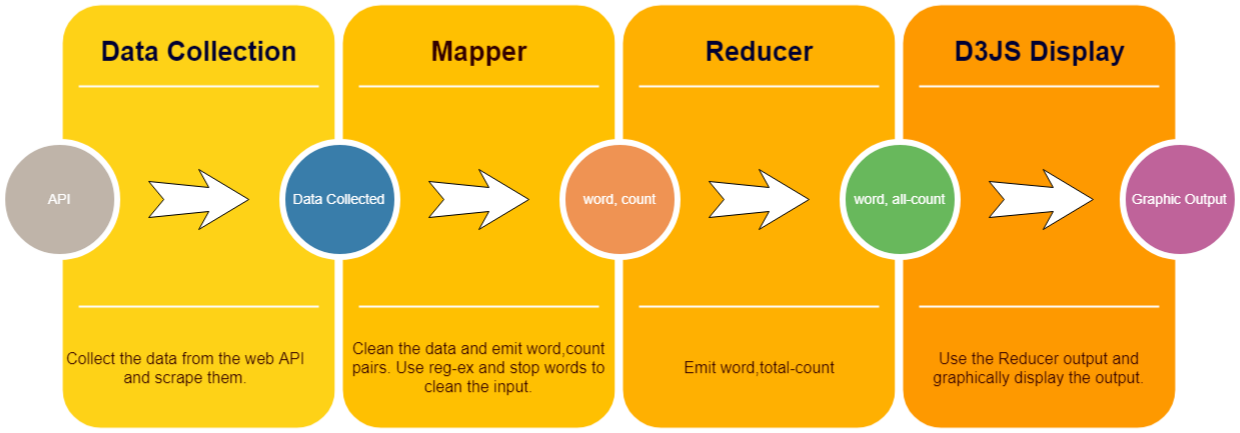
Co-occurrence Reducer:

1) Sort the <word1,word2> and feed it into the Reducer. Reducer emits the co-occurrence count. This will help give insight on the co-occurrence.

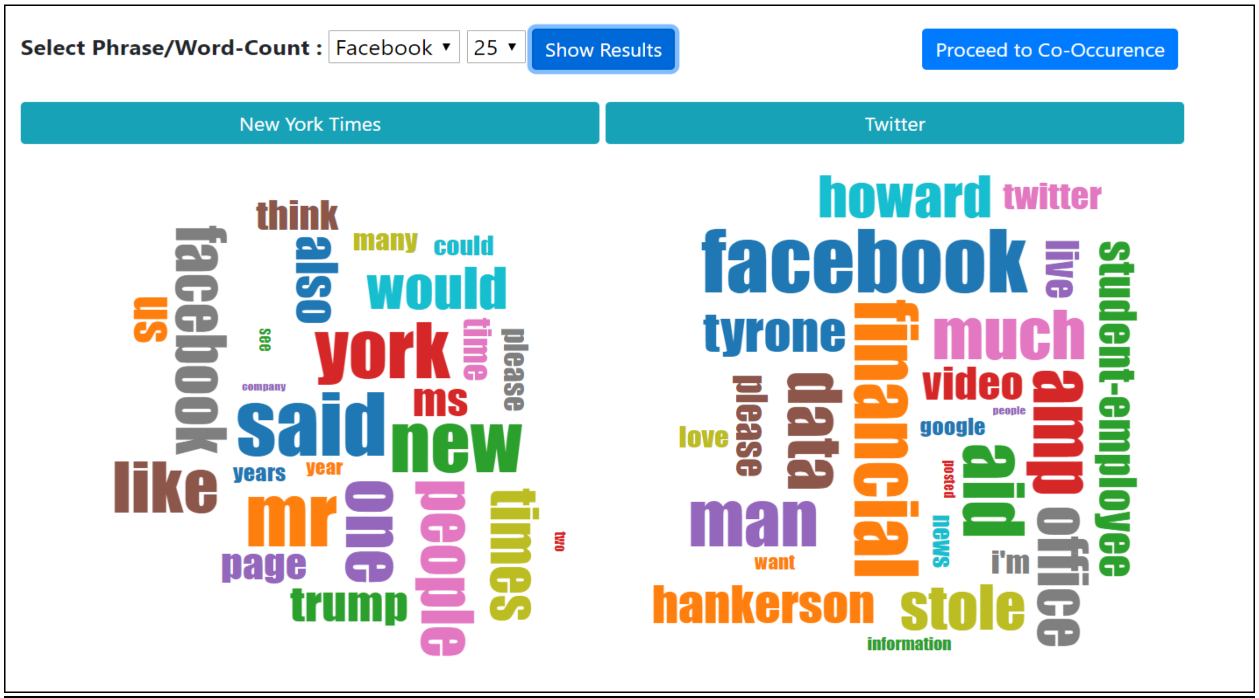
Co-occurrence Visualization:

1) The data obtained from the reducer is displayed using D3JS.

<u>Word-Count Map-Reduce Algorithm:</u>


Data Collection
Mapper
Reducer
D3JS Display
API
Data Collected
word, count
word, all-count
Graphic Output
Collect the data from the web API and scrape them.
Clean the data and emit word,count pairs. Use reg-ex and stop words to clean the input.
Emit word,total-count
Use the Reducer output and graphically display the output.

Word Cloud: Keyword: “facebook”


Select Phrase/Word-Count : Facebook 25
Show Results
Proceed to Co-Occurence
New York Times
Twitter
facebook like us think also many could would york time please see company said ms new years year mr one people times two page trump
howard twitter facebook live student-employee tyrone financial much video amp please data love google people posted aid man news i'm office want hankerson stole information

The Map Reduce word-count algorithm was run on two more key words “India” and “trump” just to see how the Algorithm behaves for new key-words.

Analysis of the word-count estimate:

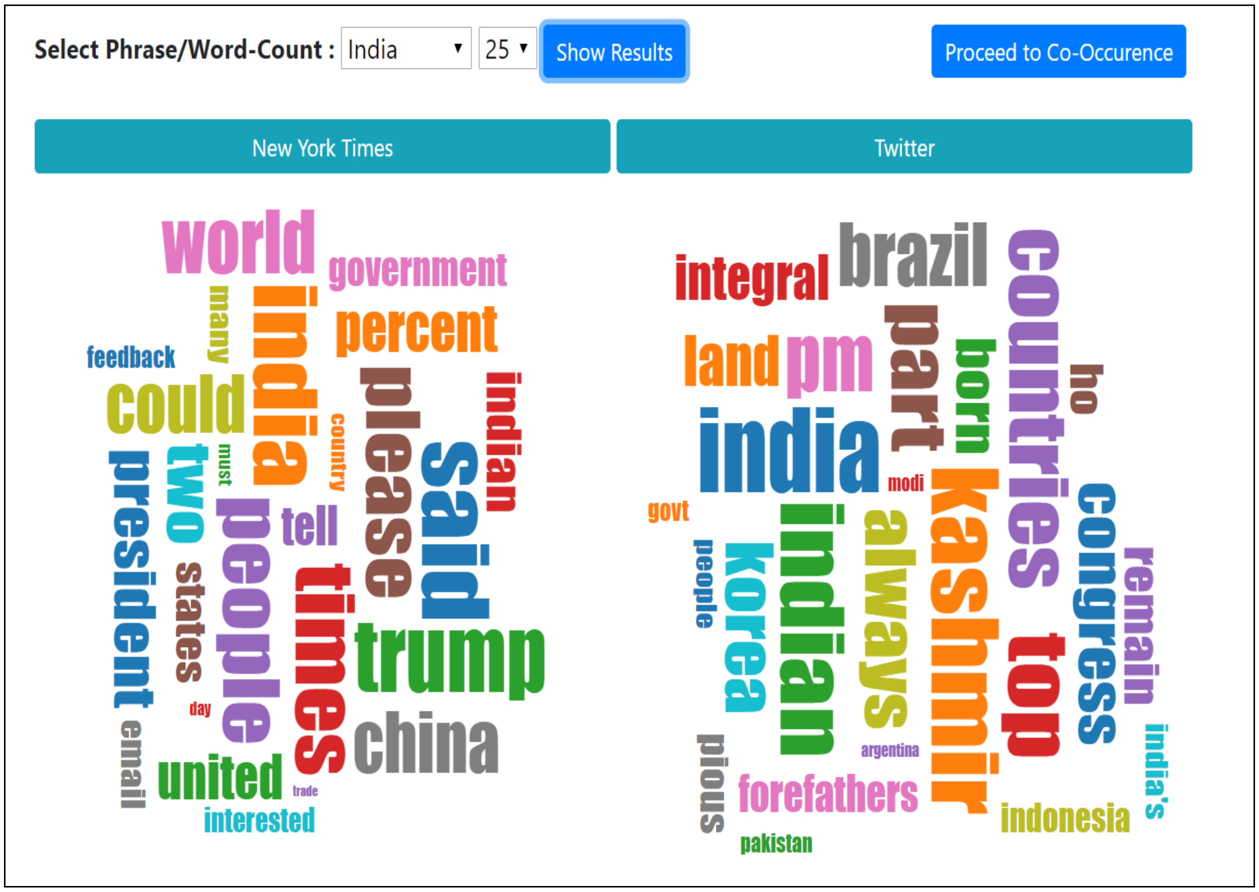
On Analysis of the frequent words on twitter and NY Times, we were able to establish some correlation between the frequent words on twitter and NY Times.

NY Times Data		Twitter Data	
Words	**Frequency**	**Words**	**Frequency**
facebook	15584	facebook	239873
People	16478	Financial	55052
trump	10593	Hankerson	27330
President	7475	News	17006
News	7051	People	14486
Media	5877	cambridge	14305
Social	5596	Media	5584

The words “facebook”,”people”,”Cambridge analytica”,”trump” are related to the issue of facebook data used to influence voters during the 2016 elections. The words “financial”,”Hankerson” are related to “Howard University” scandal which is more popular in twitter during the time we obtained the tweets. Due to nature of these issues, these are mostly localized to a place and did not get much traction in the news media. Based on the data correlation could be established between the Trump and NY Times data.

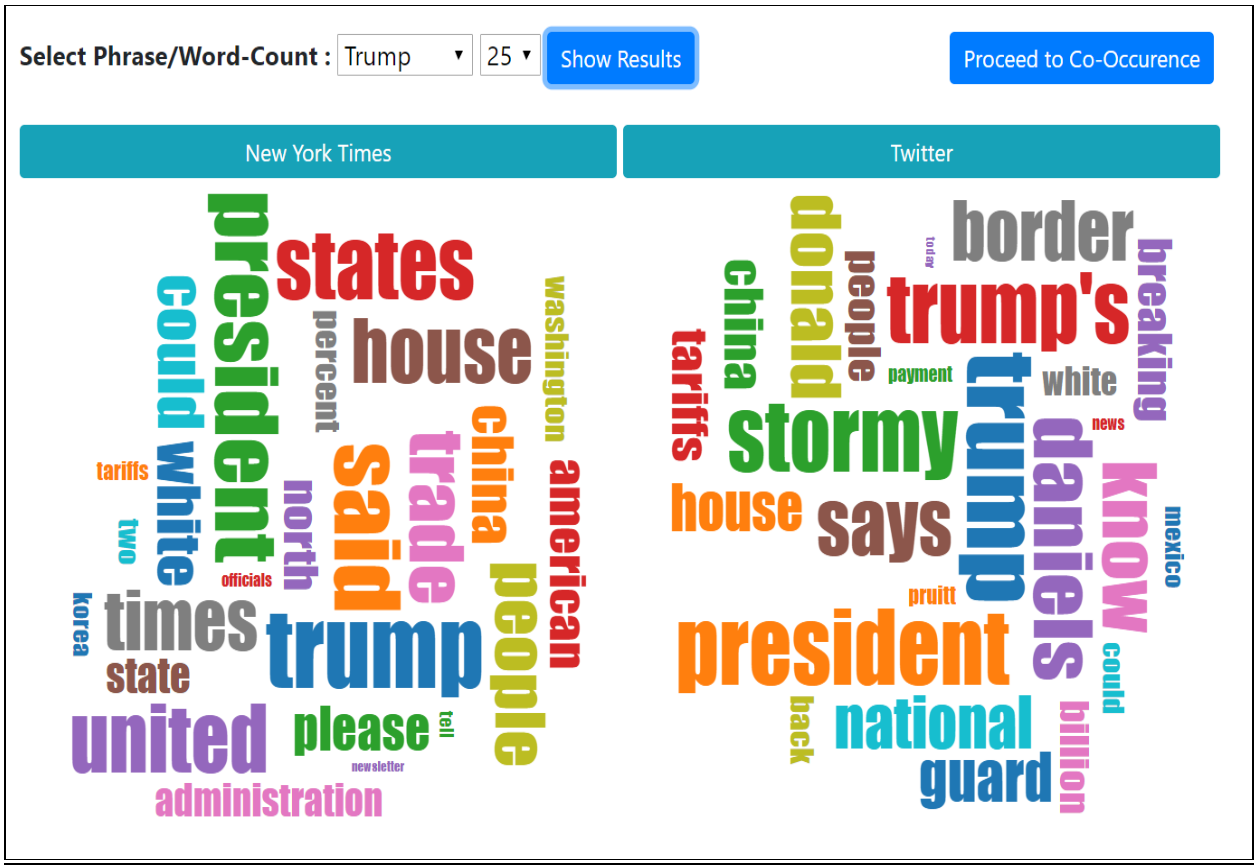
<u>Word Cloud for keyword “india”:</u>

Word Cloud for the keyword “india”, data was collected from the twitter and NYTimes. The data was fed into MapReduce Engine to give results.



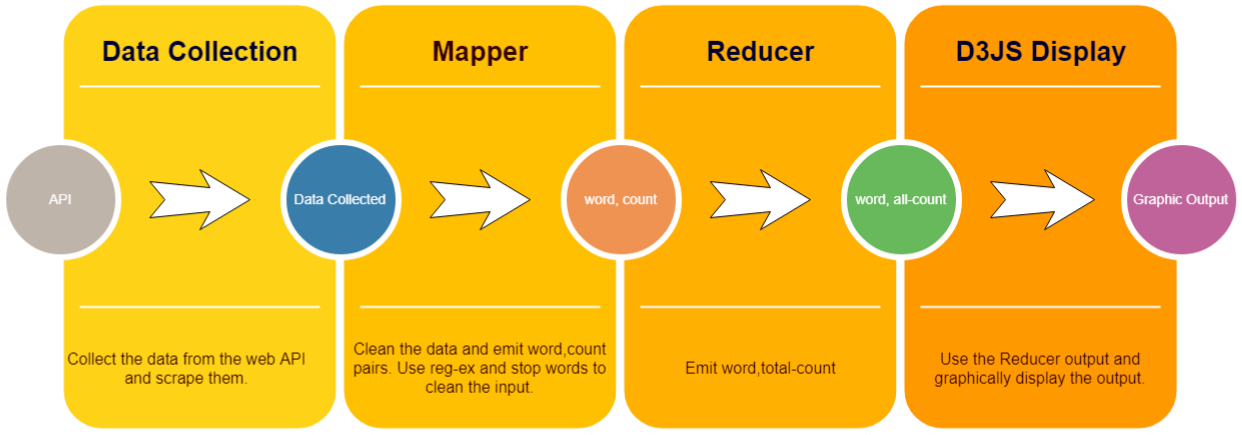
<u>Word Cloud for keyword "trump":</u>

Word Cloud for the keyword "trump", data was collected from the twitter and NYTimes. The data was fed into MapReduce Engine to give results.



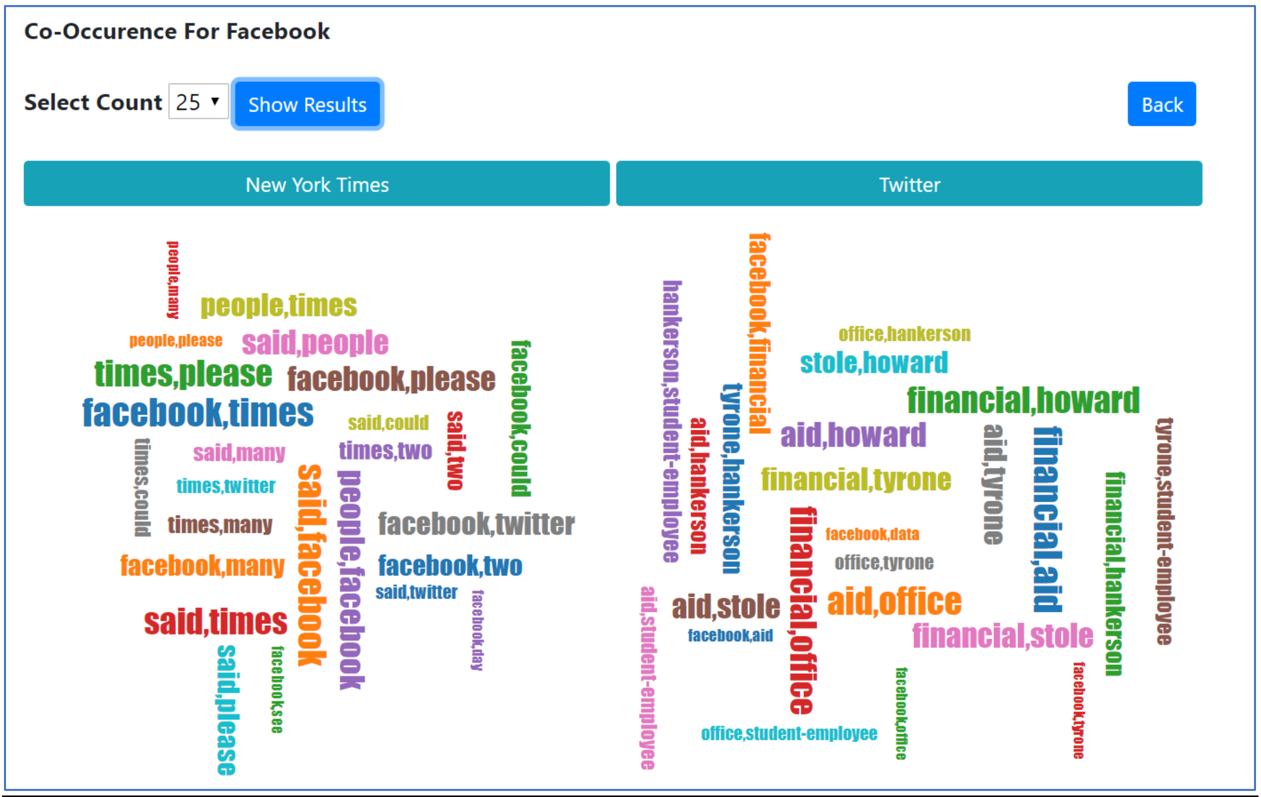
Word Co-occurrence Map-Reduce Algorithm:

Design of the word Co-occurrence map reduce pipeline.


Data Collection
Mapper
Reducer
D3JS Display
API
Data Collected
word, count
word, all-count
Graphic Output
Collect the data from the web API and scrape them.
Clean the data and emit word,count pairs. Use reg-ex and stop words to clean the input.
Emit word,total-count
Use the Reducer output and graphically display the output.

Visualization for Co-occurrence:

The following image shows the co-occurrence visualization for Word Co-occurrence.



<u>Word Co-occurrence:</u>

NY Times Data		**Twitter Data**	
Word Co-occurrence	**Frequency**	**Word Co-occurrence**	**Frequency**
said,facebook	4853	Financial,aid	47331
people,facebook	4168	Stole,howard	27314
People,times	3809	Facebook,stole	14826
facebook,president	2533	Facebook,data	26347
facebook,company	2182	Facebook,news	11773

This data helps us build a theory of our assumption of word-count. It could be noticed that the story about Hankerson misusing funds is getting lot of traction lately. The collected tweets also show us some insight on the Facebook Cambridge Analytica scandal.

<u>Directory Structure:</u>

```
Part1 - Contains the part-1 of the project.
|--- Ch3/Ch4/Ch5 - Contains the python code for Chapter3/4/5


Part2 - Contains the part-2 of the project
|--- MapReduce - Contains the MapReduce python files.
    |--- Mapper.py, Mapper_Co-occurence.py - Mapper code
    |--- Reduce.py, Reduce_Co-occurence.py - Reducer code
|--- MapReduce_output - Contains the output of the reducer.
|--- NYTimes - Contains the script for DataCollection and Scrapping for NYTimes.
    |--- datagrab_nytimes.PY – gets the article links from NY Times,output json.
    |--- GetAndDumpContent.py – reads the json and scrapes content of the article.
|--- Twitter - Contains twitter data collected using twitteR API
|--- Visualization - Contains the data and display for visualization.
```

<u>Learning:</u>

- Learnt to use WebAPIs and collect data from them.
- Learnt to clean and parse the data.
- Learnt to set up a Hadoop Infrastructure on a VM
- Learnt to code solutions in Mapper & Reducer for word-count and co-occurrence problems.
- Learnt to visualize the output of the Reducer using D3JS and derive inferences from the data.
- Learnt to document the analysis.
- Learnt to create a responsive web-tool to visualize the outcome of the analysis.