README

DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION

Aditya Subramanian Muralidaran

Project Description

To develop an Exploratory Data Analysis project, that can be used to get word-count and word co-occurrence in any document and visualize this information using d3.js for further analysis.

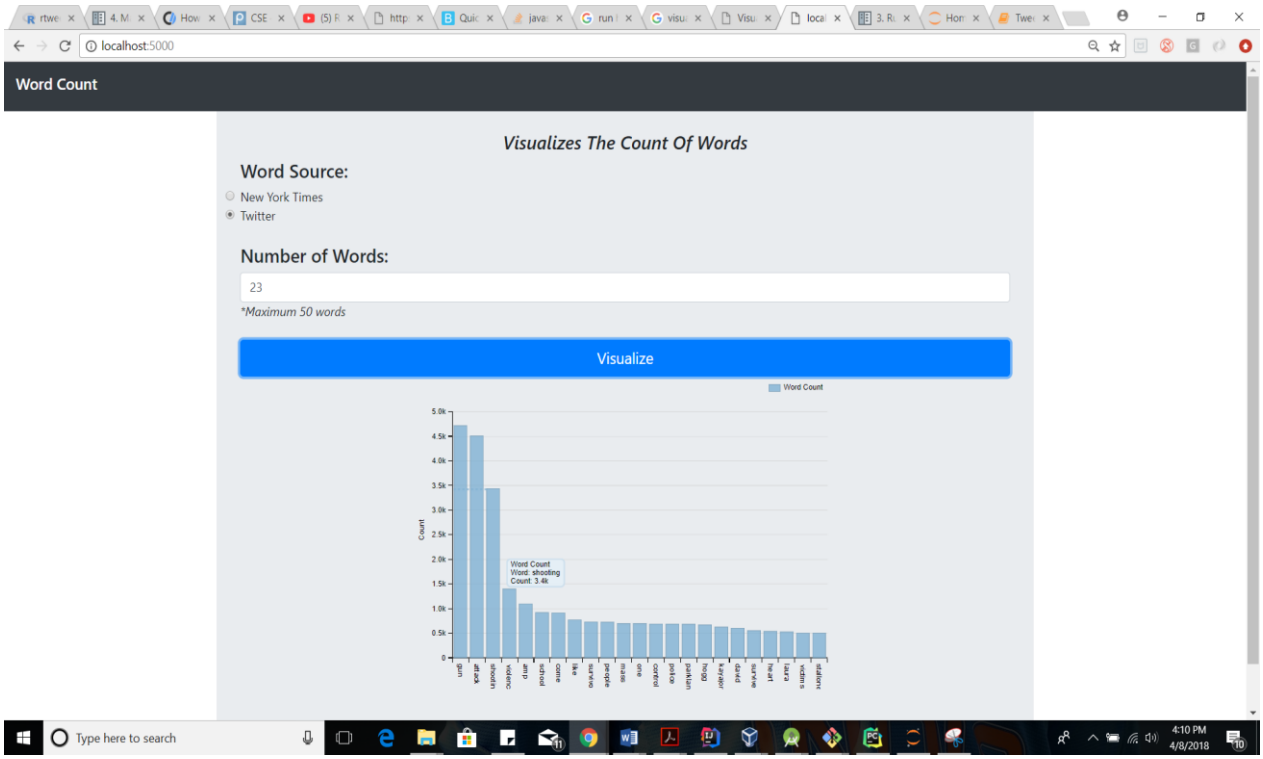
Stacks / Technology Used:

- Python programming language.
- Data aggregation from more than one source using the APIs (Application programming interface) exposed by data sources – Twitter and New York Times(Automated data collection from multiple sources using the APIs offered by the businesses and python/R scripts.)
- MapReduce from Hadoop environment.
- Hadoop 2.x, HDFS and process the data using big data algorithms.
- d3.js to learn modern visualization methods and disseminate results using the web/mobile interface

Project Steps:

- Topic for data collection was "Mass Shooting in US".
- Data sources used for collecting data:
    - Twitter
    - New York Times
- The key words used for collecting data:
    - Shooting
    - Gun
    - Attack
- Same three words for collecting data from both the data sources - New York Times and Twitter.
- Steps Followed:
    - Initially the data was gathered from Twitter using the Twitter API and 'rtweet' package in R using the script file - 'TweetData_Program.ipynb'
    - Then the data from New York Times was gathered using New York Times API as using the script file – 'GetNewsData.py'
    - Data collected for the period between - 30-March-2018 and 06-April-2018

- The data was cleaned, stop words and punctuations were removed in the word count mapper function. The output of the mapper function is (word \t 1).
- In the Reducer, the value part of each word from all the mappers were grouped together to get the actual word count as (word \t count).
- We then use this output to generate the d3.js interactive visualization where we use a Bar Plot to depict the data.
- Screenshot of word count virtualization:



- From our visualization it was noticed that the count of words in the bar plot using Twitter data are similar to the count of words from the News data.
- Then the script ‘Sorting_WordCount.py’ was used to sort the data based on word count, and used the top 10 words from the sorted data to find the co-occurrence words.
- The co-occurrence words by using a Map Reduce method for Twitter and New York Times data separately.
- In the Mapper, the top 10 words from the sorted collection was used for both the Twitter and New York Times data to generate a pair as a key and a value (which will be 1) and emitted it.
- So the output of the mapper will be of the form ({word, co-occurrence},1)
- In the Reducer, the output from the Mapper was reduced to get the count of the co-occurrence word.
- The output of the Reducer will be of the form ({word, co-occurrence}, count).
- For visualizing the co-occurrence, the word cloud from d3.js was used.

- Screenshot of word cloud: