DIC LAB – 2 REPORT: DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION

Miki Padhiary –mikipadh- 50286289
Venkatesh Viswanathan- vviswana-50290589

TOPIC:
"Politics in the United States"

INSTRUCTION FOR TESTING THE IMPLEMENTATION:

- We have used Tableau as the visualization tool and we have hosted the website using Tableau server. The website link is https://public.tableau.com/profile/venkatesh6373#!/vizhome/DIC_Lab2_6Clouds/Sheet1
- There are 6 sheets represented as Tabs and you can navigate it to see different word clouds.
- You need not need any extra setting up in your system to visualize the data as it is hosted in Tableau servers.
- We have submitted the video in UB box.The URL is https://buffalo.box.com/s/87fal3drv5axnl5qt3cibx30jvcdyir3
- Also, we have published in the video in youtube and the url is https://youtu.be/fOT5auVJRIU

Directory paths:

For testing data collection:

- */part1/Code/TweetCollection.**R*** -> to collect tweets and store them.
- */part1 /Code/NYArticleCollection.R* -> to collect article urls on the topic and store them.
- */part1 /Code/articleCrawler.R* -> to extract the content of the article from urls collected.
- */part1 /Code/Crawler.py -> to collect crawl articles and sent to processing.*
- */part1/Data/Twitter -> This will contain the csv files from which we extracted the tweets using $text*
- */part1/Data/Twitter/TwitterDataOutput* -> tweets collected and stored for Map Reduce.

- We have 3 files inside namely elections.txt, politics.txt, trump.txt that has text of 53K tweets.
- */part1/Data/NYT/* -> This will contain the csv files from which the URL's of the articles were extracted
- */part1/Data/NYT/newsDataOutput* -> Contains articles text which is used for Map Reduce.
- */part1/Data/CommonCrawl/* -> articles collected and stored for Map Reduce
- We have 2 files in it namely the crawl2_250.txt and crawl_284.txt which consists of 250 and 284 articles respectively.
- We have used the same format of appending "_filtered" after the file name once the preprocessing is done.

For Map Reduce:

- */Part2/wordcount* -> contains mapper and reducer files for Single Word Count for Twitter and NYTimes.
- /part2/wordcount/mapper_crawl.py -> contains the mapper for the common Crawl.
- */Part2/wordcooccurence_crawl*-> contains mapper and reducer files for Co-Occurring Word Count for data collected from Common Crawl.
- */Part2/wordcooccurence_nytimes*-> contains mapper and reducer files for Co-Occurring Word Count for data collected from NY Times
- */Part2/wordcooccurence_tweets*-> contains mapper and reducer files for Co-Occurring Word Count for data collected from Twitter
- */part2/results* -> Output of all the three sources for word count and word Co-occurrence
- We have maintained a pattern in which if the text file ends in "Co" then the file is for Co-occurence and if the file ends in "count" then the file is for word Count.

For Visualization:

- */Part3/NYT/Code* -> contains the processing code for Nytimes
- */Part3/NYT/Images* -> contain Images of word cloud and word co-occurrence for NY times
- *Similarly, we have maintained the same folder structure for Twitter and Common Crawl*
- */part3/twbx File* -> There is only one tableau file where we have 6 worksheets each having a single word cloud.

IMPLEMENTATION

Part - 1: Data Collection and Preprocessing of Data

- R is used as the language for data collection and cleaning tweets and NYTimes Articles.

TWITTER tweets:

- Total of 53,000 tweets is collected on the topic using the keyword described above and hashtags similar to the topic and the sub topic that we choose.
- **rtweet** package is used to extract tweets.
- Tweets are filtered using the text, tweet ids to remove duplicates.
- Data is cleaned by removing non-ASCII characters, symbols, hashtags, mentions, numbers in the preprocessing step.

NY TIMES articles:

- **Rtimes** package is used to extract the url of articles using keyword like *"politics", " administration", "government", "voting", "Trump" and "elections"*. Apparently politics was our main topic and the rest were our five topics.
- rtimes is the package used to collect data from NYTimes and **Rcrawler** package is used to crawl the webpages containing the url and extract the article content.
- A total of 2500 articles was collected but as our reducer was not supporting this huge data we decided to run our reducer on 600 articles.The data is collected after removing duplicated articles using article id.
- Articles are cleaned by removing non-ASCII characters, symbols, hashtags, mentions, numbers in the preprocessing step.

STEPS TO COLLECT DATA FROM COMMON CRAWL:

- A total of 550 articles related to the topics were collected from the common crawl.
- Go to index.commoncrawl.org
- It will list all the indices that it has. Choose the most recent one. In our case it is March 2019.
- Choose any one of the publicly available URL's which we feel like it will have more data. In our case we have chosen www.cnn.com/politics and www.nytimes.com/politics.

- Upon hitting the URL, we will be provided with a JSON file which will have many attributes but the one we are interested is the filename attribute. Append it to the URL https://commoncrawl.s3.amazonaws.com/ and replace warc with wet and .warc.gz with .warc.wet.gz which will give us a WET files we need.
- Upon obtaining the WET files which will have the data as well as the Web URI, we can extract the complete data or just the articles.
- We are having a count variable just to show the number of articles we collected as we are not appending the Web URI's to a file and hitting the web urls.
- We are iterating through the wet files downloaded and are writing only those articles which matches our keyword of politics, voting, trump, administration, elections and government.

Data Pre Processing and Lemmatization:

- We are cleaning the data once the data is collected. Each kind of source has its own kind of cleaning.
- Also apart from cleaning the data we are removing the most common English words known as the stop words.
- We are also **lemmatizing** the word so that we will be able to get the root form of the words.
- For twitter, we have to remove the retweets, Duplicated tweets, mentions, hashtags and mentions.
- For New York Times, we just have to remove the duplicates
- For Common crawl, from the WET files we have to filter out the data that are relevant to our topics and extract it.

Part – 2: Set up big data infrastructure

- We have used the docker image for setting up hadoop infrastructure.
- We downloaded **Docker for Desktop** and pulled the **Cloudera-Quickstart** docker Image.
- We used the below commands for creating a directory inside hadoop and transferring input/output to and back from hadoop system:
  hadoop fs -mkdir /user/miki/MR/input - For creating directory.

  hadoop fs -put *.txt /user/miki/MR/input - For moving the files from local to hadoop infrastructure.

  hadoop fs -ls /user/miki/MR/input - For viewing whether files were moved.

  hadoop /jar/usr/lib/hadoop-mapreduce
  /hadoop-streaming-2.6.0-cdh5.7.0.jar
  -file /src/mapper.py -mapper /src/mapper.py
  -file /src/reducer.py -reducer /src/reducer.py
  -input /user/miki/MR/input/*.txt
  -output /user/miki/MR/output - For running mapper and reducer.

hadoop fs -get /user/miki/MR/output/ /src/ - For retrieving the output from hadoop to local system.

- Python is used for creating the mapper and the reducer script.

- Separate mapper and reducer scripts are written for both single word count and co-occurring word count. So, for single word count there is a mapper and reducer. For common crawl wordcount, we have created another mapper and we wanted to remove few irrelevant words. For co-occurring words there are 3 mappers, one for each data source and the respective reducer.

mapper.py

- The mapper.py file outputs the words with count as 1 as <key, value> pair for single word count part and co-occurring words with count as 1 as <key, value> pair for co-occurring word count part.
- We have additionally added our own stop words so that we can restrict those words that are most commonly used words in English.
- Output of the mapper is sorted according to the key. The key and value (single word/ co-occurring words and '1') are tab separated.

reducer.py

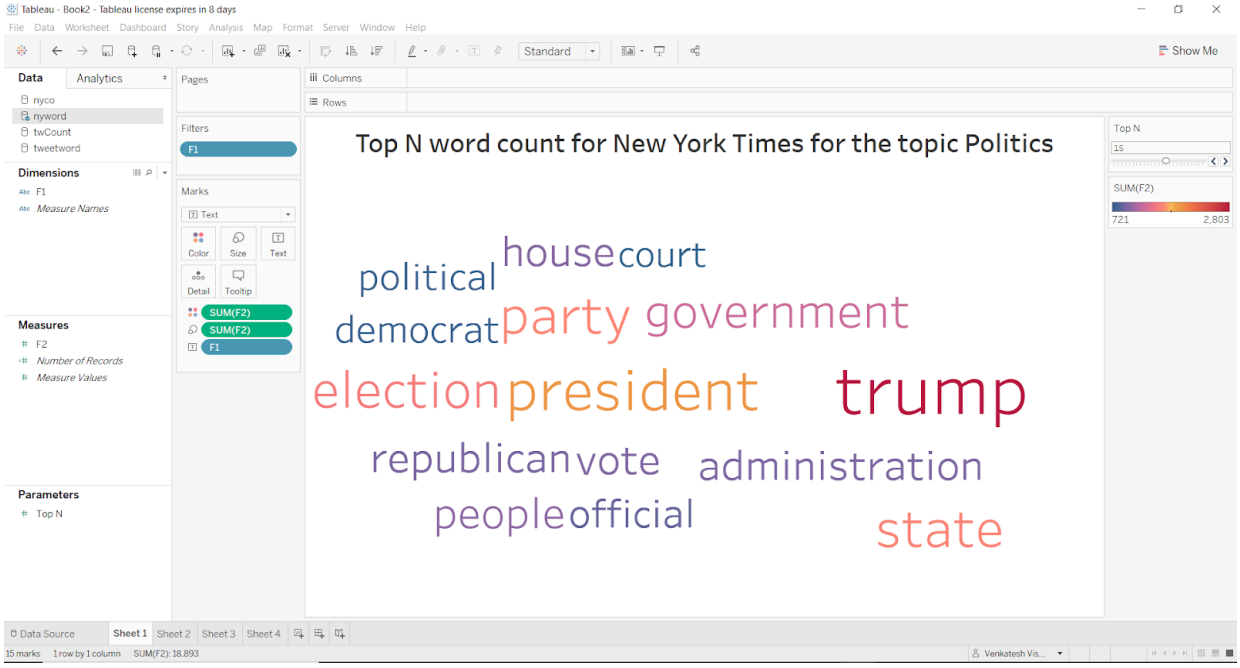
- The reducer splits the output of the mapper based on tab '\t'.
- It then checks that if the keys are same (single word/co-occurring words), it adds their corresponding values.
- The reducer outputs the word/co-occurring words with its actual count.

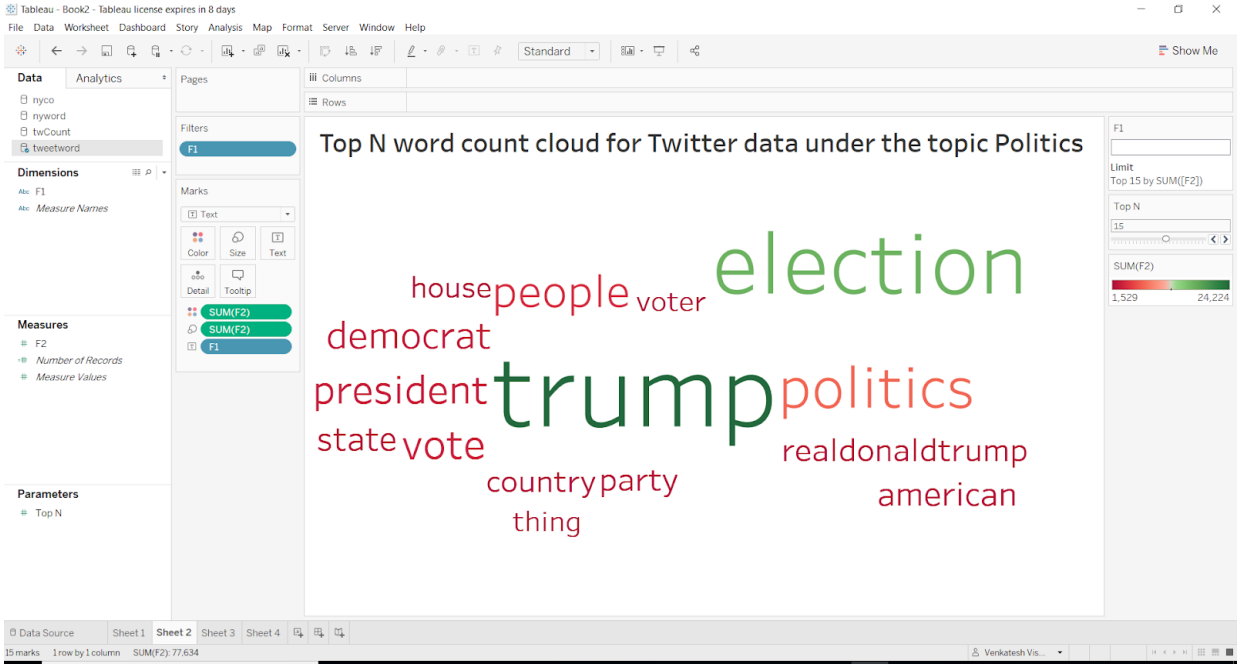
The only change for the Co-occurrence mapper is that we have 3 mappers for each source of data source i.e Twitter, NYTimes and Crawl because we are inputting the top 10 words from word count's reducer in the co-occurrence. Also, in the co-occurrence mapper we and storing the data in alphabetical order. For example if out top word is trump and the next occuring word is politics then we will store it as politics|trump, so that when we encounter politics and then trump it does not get stored as different word. Also, we are not storing the words like politics|politics.

Part – 3: Visualization of word count and word cooccurrence

- Used **Tableau** to visualizing the word cloud for the **top N** word count and co-occurring words.
- You can enter any number in the range of 0 to 25 and the respective word cloud will show up.

- We have made it interactive by adding a slider so that you can enter it to visualize the word cloud.
- The word cloud images are as follows





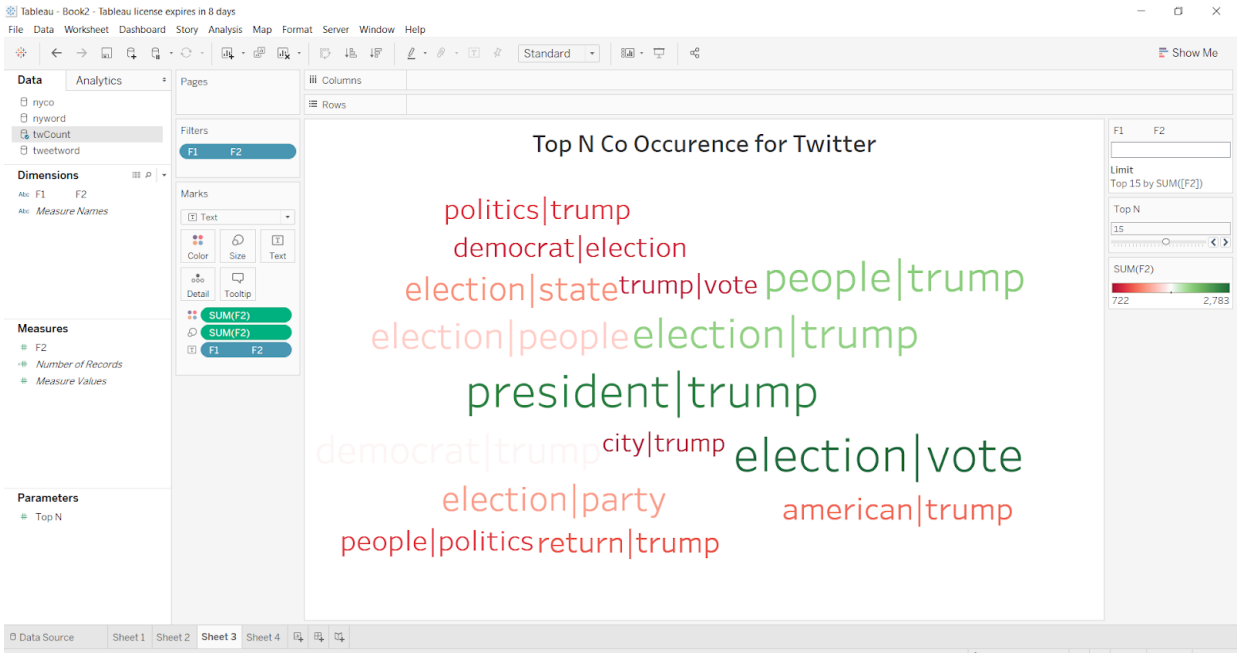


Tableau - Book2 - Tableau license expires in 8 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data
Analytics
nyco
nyword
twCount
tweetword
Dimensions
F1
F2
Measure Names
Measures
F2
Number of Records
Measure Values
Parameters
Top N
Pages
Filters
F1
F2
Marks
Text
Color
Size
Text
Detail
Tooltip
SUM(F2)
SUM(F2)
F1
F2
Columns
Rows
Top N Co Occurence for Twitter
politics|trump
democrat|election
election|state
trump|vote
people|trump
election|people
election|trump
president|trump
democrat|trump
city|trump
election|vote
election|party
american|trump
people|politics
return|trump
F1
F2
Limit
Top 15 by SUM([F2])
Top N
15
SUM(F2)
722
2,783
Data Source
Sheet 1
Sheet 2
Sheet 3
Sheet 4

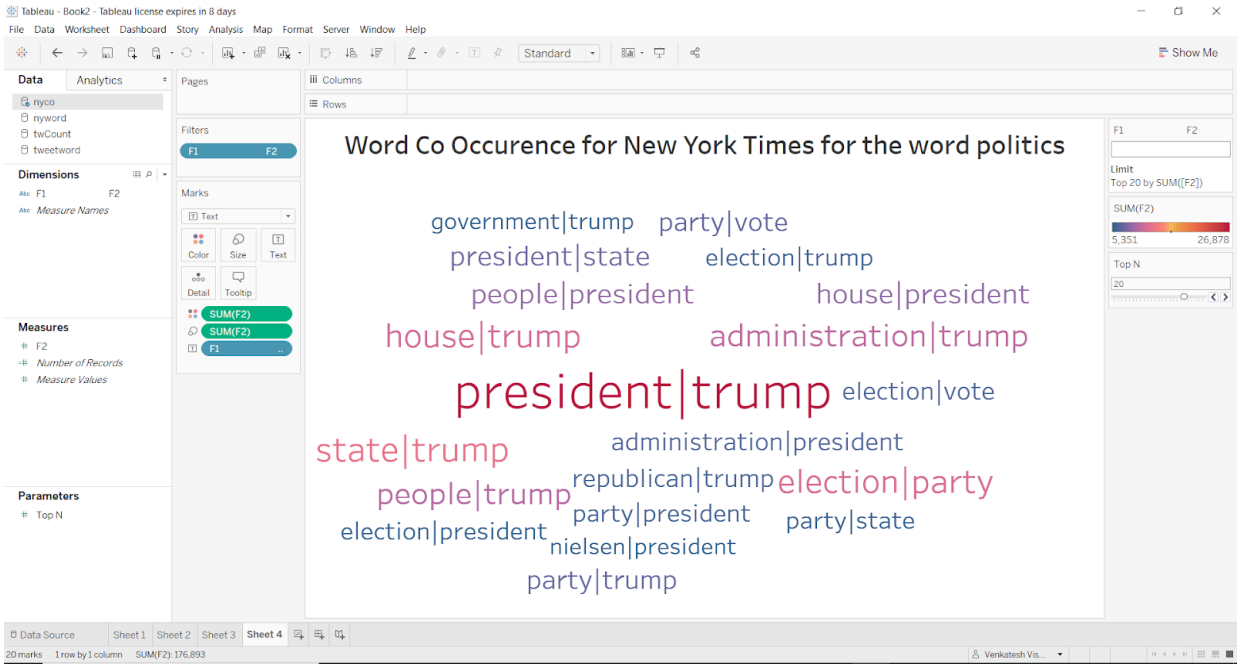
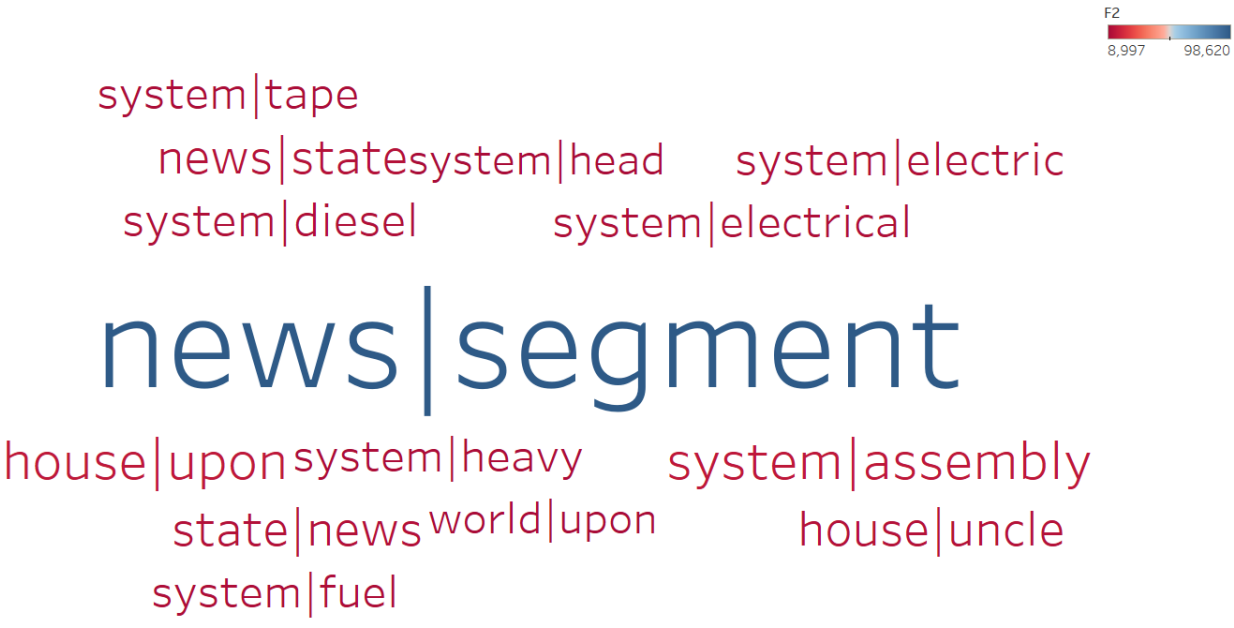


Tableau - Book2 - Tableau license expires in 8 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data
Analytics
nyco
nyword
twCount
tweetword
Dimensions
F1
F2
Measure Names
Measures
F2
Number of Records
Measure Values
Parameters
Top N
Pages
Filters
F1
F2
Marks
Text
Color
Size
Text
Detail
Tooltip
SUM(F2)
SUM(F2)
F1
Columns
Rows
Word Co Occurence for New York Times for the word politics
government|trump
party|vote
president|state
election|trump
people|president
house|president
house|trump
administration|trump
president|trump
election|vote
state|trump
administration|president
republican|trump
election|party
people|trump
party|president
party|state
election|president
nielsen|president
party|trump
F1
F2
Limit
Top 20 by SUM([F2])
SUM(F2)
5,351
26,878
Top N
20
Data Source
Sheet 1
Sheet 2
Sheet 3
Sheet 4
20 marks
1 row by 1 column
SUM(F2): 176,893

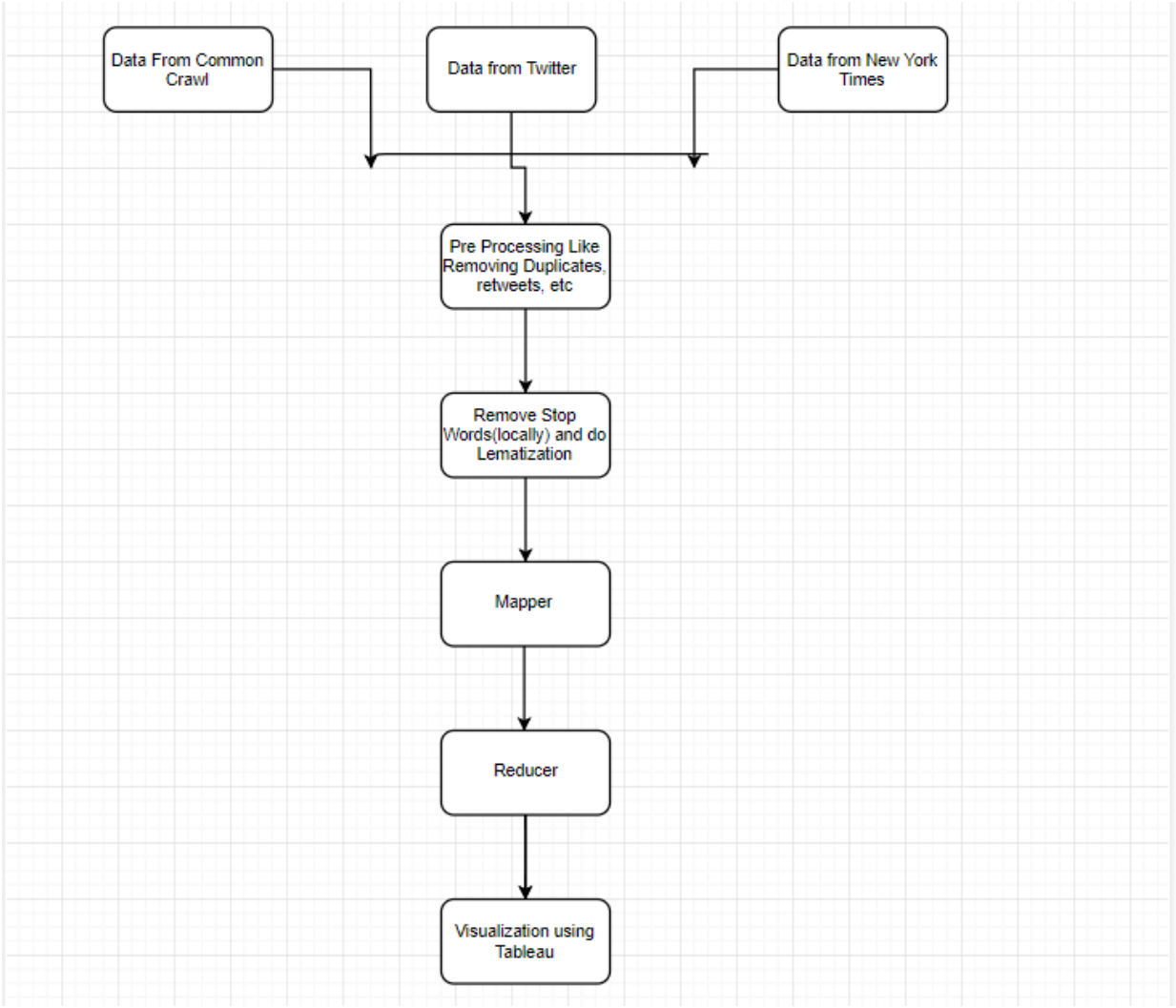
Top N word count for the topic Politics visualized in Word Cloud using Common Crawl

F1. Color shows sum of F2. Size shows sum of F2. The view is filtered on F1, which has multiple members selected.

Top N Co Occurence for the topic Politics visualized using Word Cloud using Common Crawl



F1. Color shows sum of F2. Size shows sum of F2. The view is filtered on F1, which has multiple members selected.



Analysis

- Words like “trump”, “election”, “president”, “politics”, “vote” were having very high frequency and were found to be common in both tweets and ny times articles data.
- The word count on tweets shows “trump”, “election”, “politics”, ”president” – we presume the reason for this to be the common usage of these words in many tweets and they represent the idea on topic mostly.
- Tweets also contain a lot of commonly used/ colloquial words that people use in their day to day life while the words in NYTimes articles are more refined.
- “President | Trump” is the most co-occurring pair. This is true because it is Trump who has been the president of the US for the past 4 years.
- For common crawl word count, we got frequent words like news, trump , people , state.

- Common crawl word occurrence gave us words like news|segment, system|tape, house|upon as most occurring as the context of the articles were not correct. We looked for our keywords and extracted the articles even if it was not matching the context of our topics and subtopics.

References:

[1] https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop- mapreduce-client-core/MapReduceTutorial.html – Reducer

[2] http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program- in-python/

[3] The template of the report was taken from the link https://github.com/akshay993/Data-Integration-Big-Data-Analysis-And-Visualization-Apache-Hadoop

[4] https://rushter.com/blog/python-fast-html-parser/

[5] Common crawl (open data). http://commoncrawl.org/.

[6] Tableau, https://www.tableau.com/, last viewed 2019.