DIC Lab 2 Report

Name: Rajiv Ranjan Ubid : 50249099
Team -partner name: Pradeep Aitha

Part1: Complete the python code expositions discussed in Chapter 3-5 of your text book. Keep all the source code in a three directories: Lab2->Part1->Ch3, Lab2->Part1->Ch4, Lab2->Part1->Ch5.

Soln:

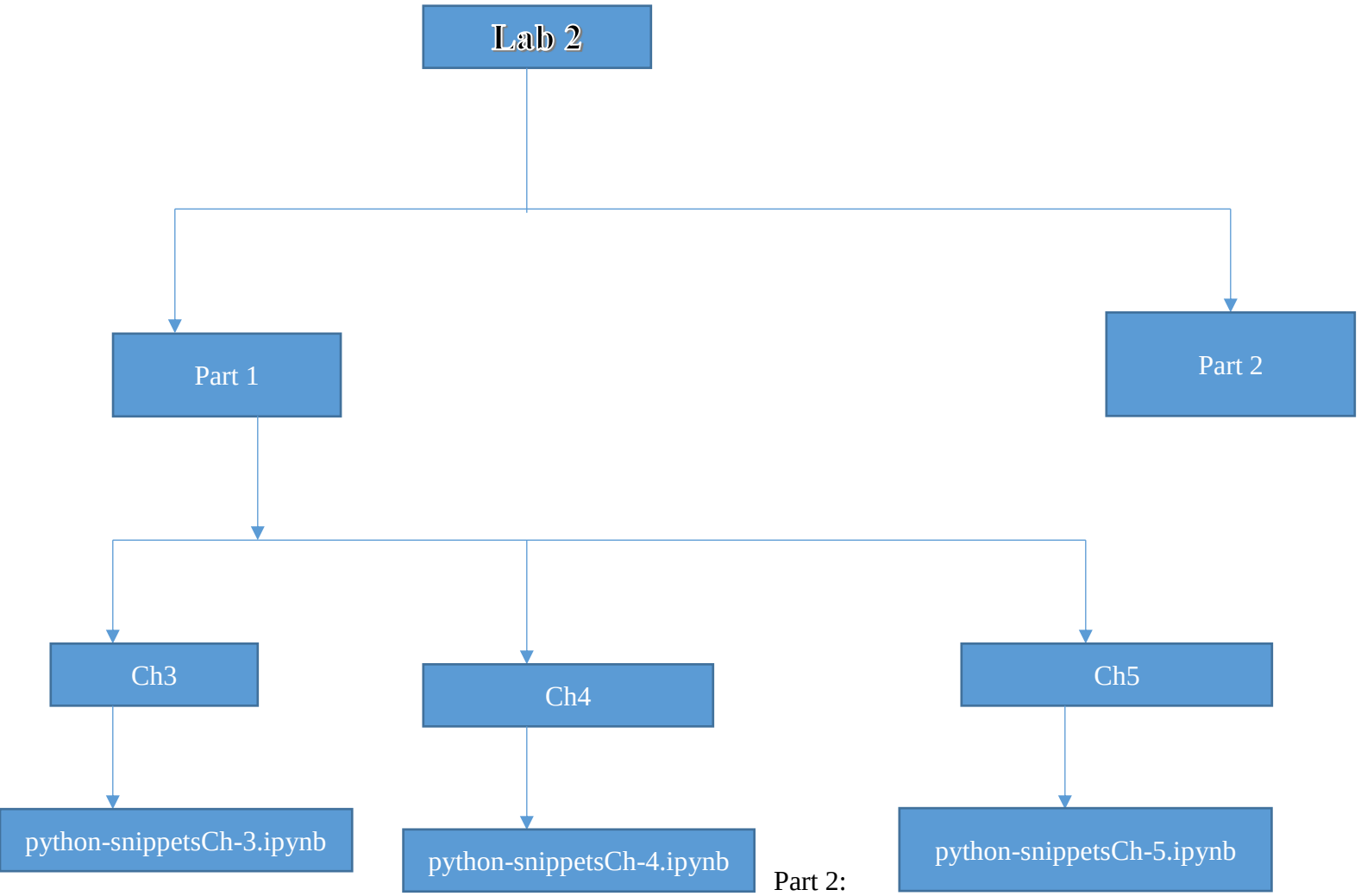
As per the requirement 3 Jupyter notebook files have been created one with codes and outputs of each chapter.

Chapter 3 – filename- python-snippetsCh-3.ipynb

Chapter 4 – filename- python-snippetsCh-4.ipynb , along with with other necessary input and output files

Chapter 5 – filename- python-snippetsCh-5.ipynb, along with with other necessary input and output files

Folder structure:



Part 2:

Now that you armed with the language to process your data, gather the data. The second part of project involves (i) aggregating data from multiple sources (ii) process using bug data methods and (iii) visual rending for review and decision making.

a. Choose a topic of current interest to people in the USA. Something that is in the news. Use the topic as the key word or phrase to aggregate tweets and news articles about the topic for the same period. For the initial prototype just use 1 day, later you can collect these two set of data for the same period from the different sources you have identified. You may have to tweak the phrase to get a good yield of tweets and news articles.
   Soln: The topic chosen for this analysis is **'Trump'** or **'Donald Trump'.**
   So, the data was collected for this over various days and for both of collection of tweets and News article data.
   These have been kept in following folder structure:
   Lab2->Part2->input_data->NewsData ->individual :: News article data collected over several days are kept inside this folder
   Lab2->Part2->input_data->NewsData ->combined:: data of all the individual files combined in one.File name: combined_article.txt
   Lab2->Part2->input_data->TwitterData->individual:: inside it various individual files containing tweets collected over several days are kept
   Lab2->Part2->input_data->TwitterData->combined :: data of all the individual files combined in one.File name: combined_tweet.txt

   The phrases were tweaked to collect a comprehensive list of tweets and news article data. Different phrases used are:trump,#trump,#donaldtrump, president, #trumptower

b. Now import the VM appliance for Hadoop infrastructure and test the basic commands with the sample data provided.

Soln: The VM appliance was imported on the system and the basic commands along with the sample data set provided was run. However that didn't work so, I had to install my own linux and Hadoop and I ran a sample data set for the commands and output could be found here at this location:
Lab2->Part2->sample:: two files are here sample_input.txt and the output in sample_output.txt

c. Load the data aggregated in step (a) into the VM, two directories: TwitterData and NewsData. Each directory can have many files of data.
   Soln: All the input files collected over various days are at below location:
   1) Lab2->Part2->input_data->TwitterData->individual-> *.txt
   2) Lab2->Part2->input_data->NewsData ->individual -> *.txt

d. Code and execute MapReduce word count on each of the data sets. Map will clean and parse the data sets into words, remove stop words, and reduce will count the useful words. Twitterdata◊TwitterWords and NewsData◊NewsWords. Review and visually compare the output for representative words about the topic. You may have to change the

search word, obtain new sets of data that may comparable sets of output words. You can use Python or java for your coding language.
Soln: The Mapreduce was coded and executed on the Python.
Imp: Only the top 20 outputs is being shown:
Let's see the execution for a few files individual files:
Twitter:
Input file: 1523158062.0526302_trump.txt
Output: Was stored in Json format initially and later converted into csv for web page creation:

{'building': 28, 'injured': 16, 'apartment': 9, 'trump': 54, 'fire': 71, 'there': 10, 'well': 11, 'dead': 11, 'realdonaldtrump': 35, 'four': 10, 'person': 10, 'fires': 11, 'trumptower': 139, 'year': 11, 'MAGA': 10, 'tower': 15, 'should': 10, 'sprinklers': 18, 'died': 21, 'built': 11}

Similarly all the files were and executed. Now let's see the the output all the tweets combined:
Input file: combined_tweet.txt
Output: The output of all the tweets combined is as follows:

{'building': 46, 'estamos': 38, 'trump': 619, 'wishlist': 43, 'fire': 146, 'message': 42, 'donald': 37, 'realdonaldtrump': 188, 'Mexicans': 37, 'Command': 37, 'trumptower': 181, 'UNIDOS': 39, 'alone': 37, 'president': 241, 'MAGA': 86, 'america': 49, 'POTUS': 84, 'tower': 75, 'QANON': 44, 'donaldtrump': 380}

News Article:
Input file: 1523172843.027815_article.txt
Output: the Json output

{'acumen': 1, 'insisted': 1, 'campaign': 1, 'just': 1, 'smoothly': 1, 'very': 2, 'going': 1, 'inauguration': 1, 'transition': 1, 'administration': 1, 'Trump': 1, 'business': 1, 'skills': 1, 'donald': 1, 'management': 1, 'extolled': 1, 'throughout': 1, 'presidential': 1, 'before': 1}

Similarly, all the files were and executed. Now let's see the the output all the news article combined combined:

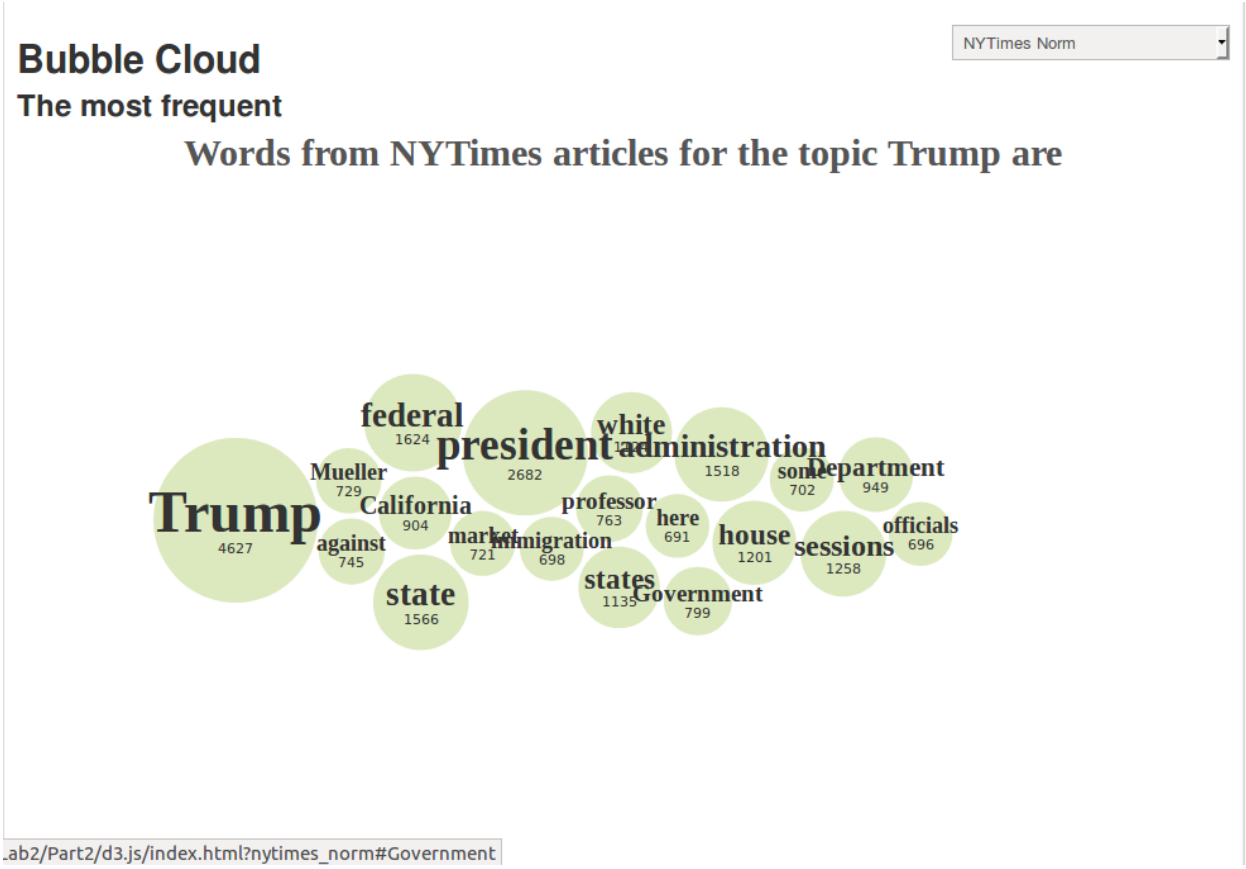
Input file: combined_article.txt
Output: the Json output

{'Mueller': 729, 'some': 702, 'Trump': 4627, 'sessions': 1258, 'federal': 1624, 'house': 1201, 'professor': 763, 'Government': 799, 'administration': 1518, 'immigration': 698, 'against': 745, 'states': 1135, 'state': 1566, 'officials': 696, 'California': 904, 'here': 691, 'Department': 949, 'president': 2682, 'white': 1124, 'market': 721}

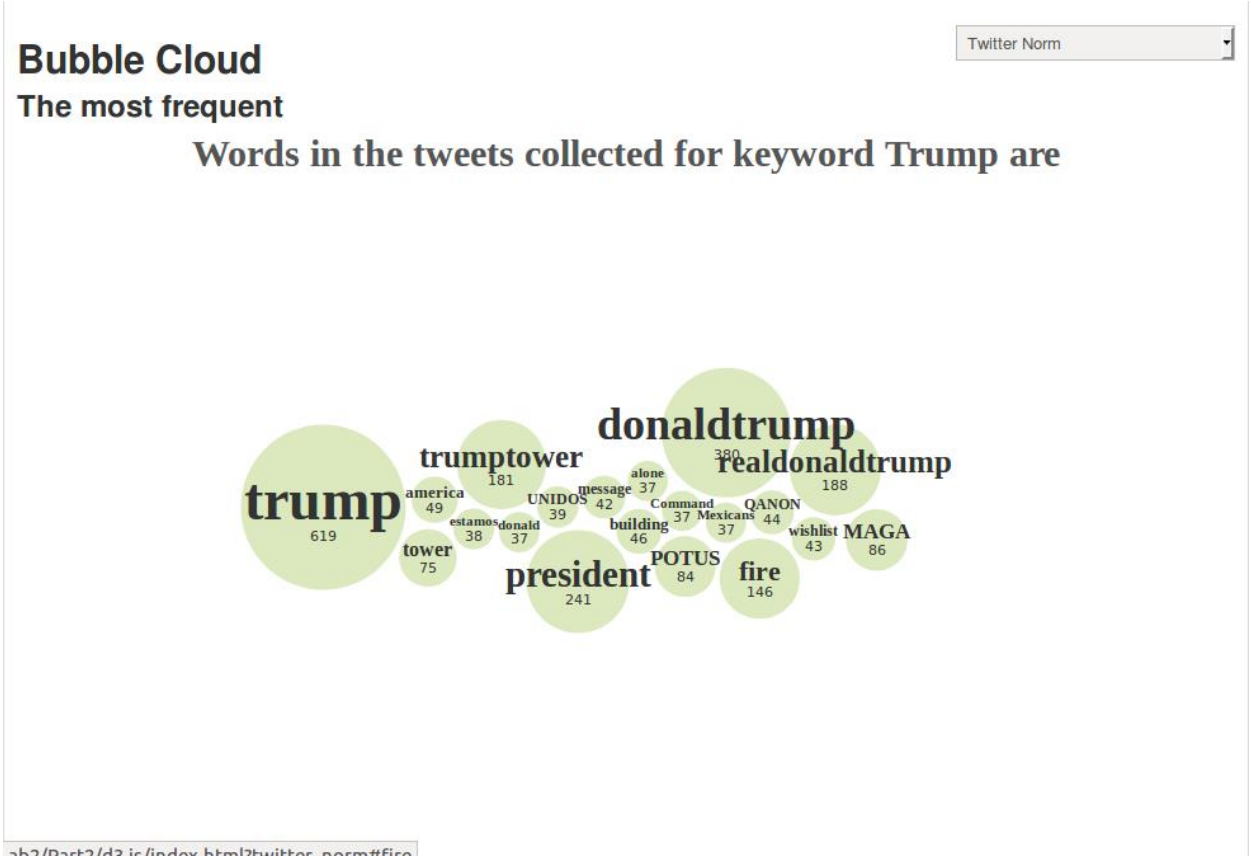
Execution Steps:
A)Twitter

1)Firstly all the code for this can be found in the following location: Lab2->Part2->code ->twitter
2) The tweets were collected over several days and also the keywords were tweeked so that there could be more data, for example: #trump,#donaltrump,#president,#trumptower etc. The code is in the file twitter_data_fetch.py
3)To run the combined file all the individual files were firstly combined. The code can be found at twitter_combine_file.py
4)The mapper code parses the data sets into words, remove stop words, and reduce will count the useful words. The code can be found at twitter_mapper_norm.py
5)The reducer code can be found at twitter_reducer_norm.py
The reducer code combines all the similar words and shows their count. The input is a key value pair generated by the mapper file. The output generated by the reducer code is also in form of key, value pair key is the word and value is the count of it.
It is to be noted that only top 10 words are being shown here.

e. Visualize each of the outputs using d3,js and on a simple web page that you create for this lab.
Soln: A output for a single file execution is below:
The output has been visualized using d3.js and on a simple web page:
It can be run from:
Lab2->Part2->d3.js->index.html
Select NYTimes Norm Single for dropdown menu.
Output is below for NYtimes:

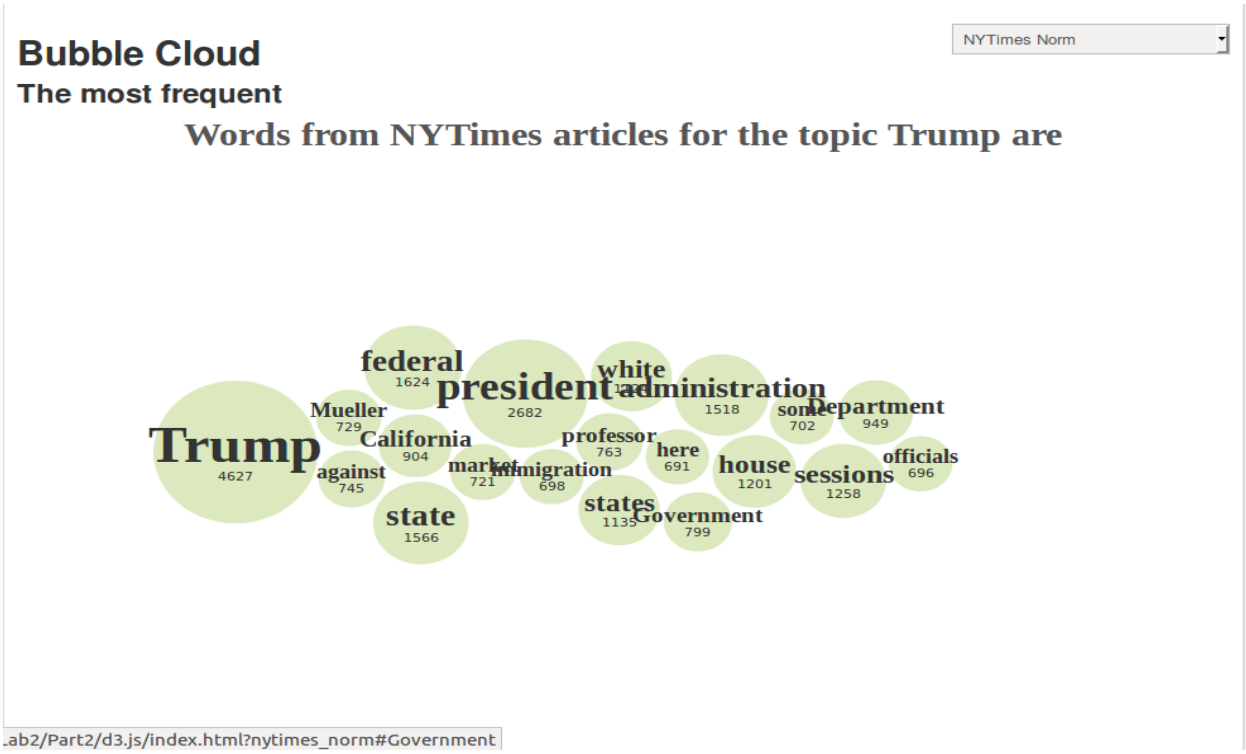


Output is below for Twitter:
Select twitter norm single from drop down menu

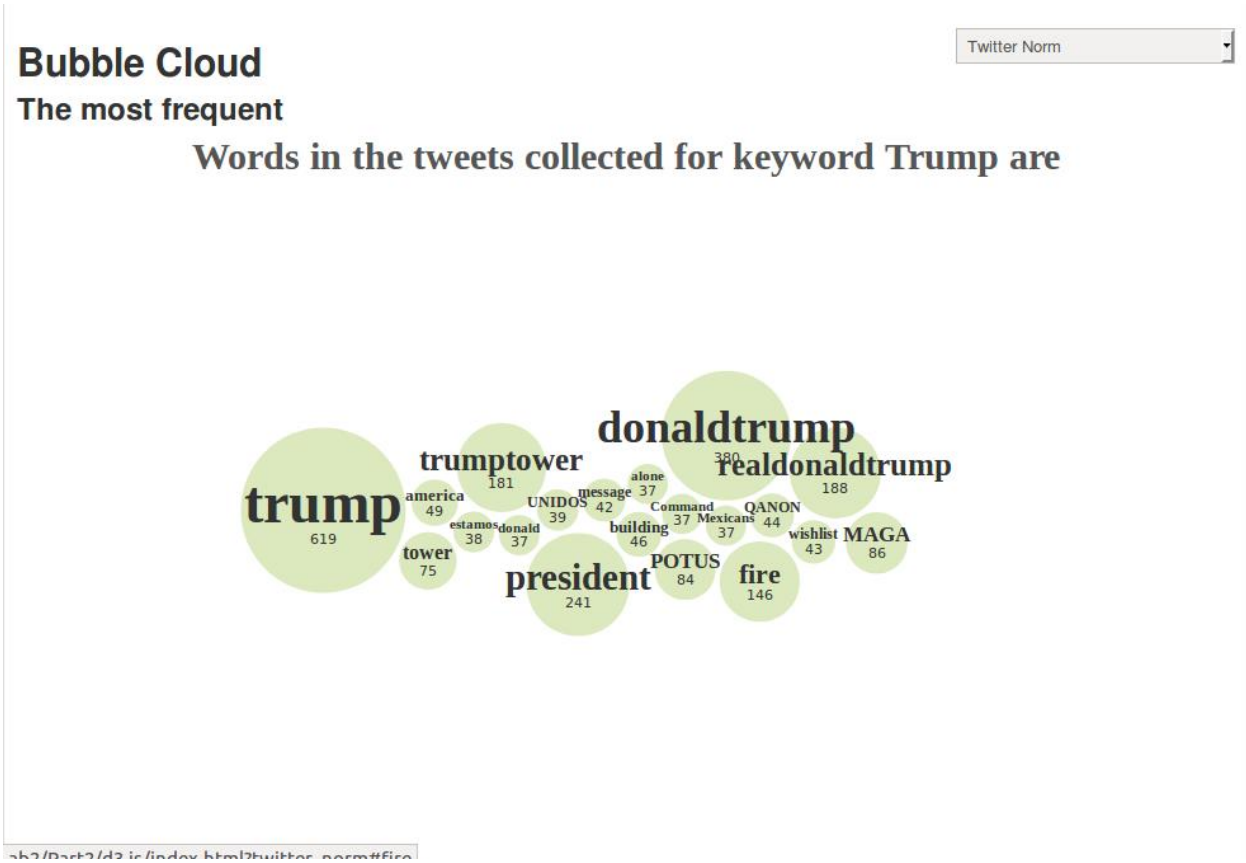


f.) Now at the steps c) to e) for larger data set collected over week. May be you will see some convergence in your output.
Soln: As mentioned above the whole process was repeated for the bigger combined dataset and shown below. Only top 20 words are shown below:

The output has been visualized using d3.js and on a simple web page:
It can be run from:
Lab2->Part2->d3.js->index.html
Select NYTimes Norm for dropdown menu.
Output is below for NYtimes:



Output is below for Twitter:
Select twitter norm from drop down menu



g.)Now design a web page and feed the results by embedding d3.js code (with replaceable worldclouds) in it, finalizing the display of results. In fact, you should be able to create interactive data product! Input a search topic, we will return the word cloud associated with that topic!

Soln: The same has been implemented with 8 different options to choose from:
It can be run from:
Lab2->Part2->d3.js->index.html

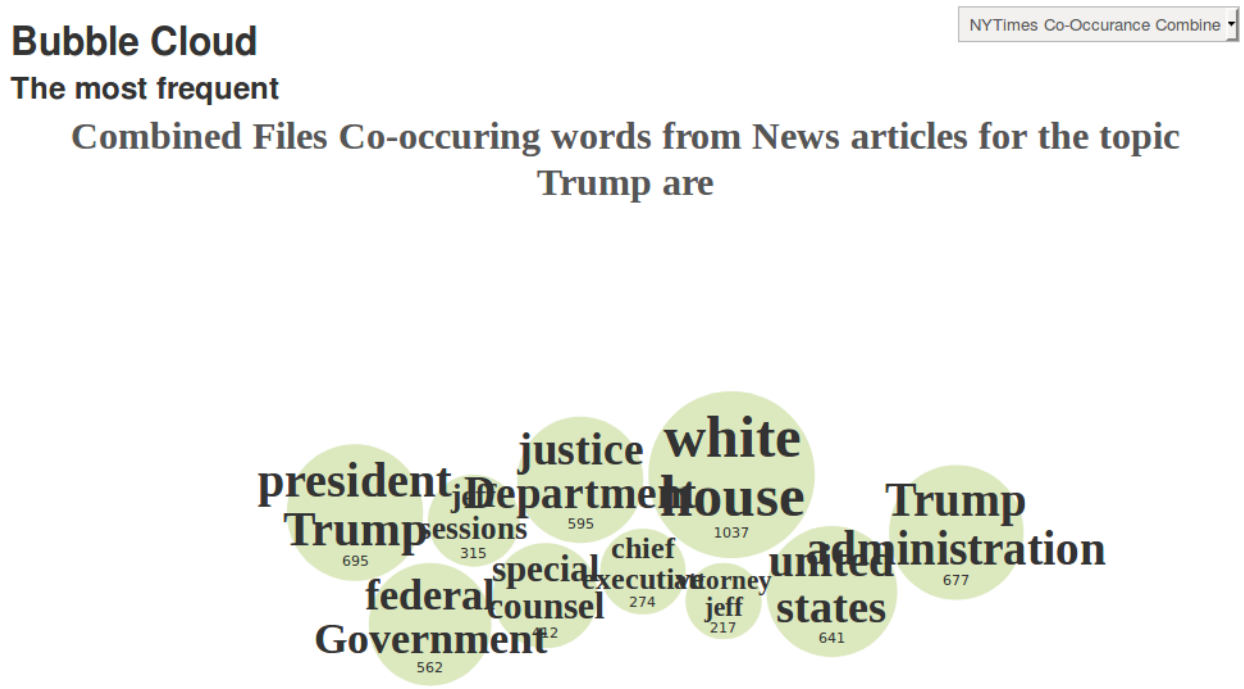
The different options and screenshot is below:

1) Twitter Norm Single : single input file wordcloud for twitter data
2) Twitter Norm Combined : all the combined files wordcloud for twitter data
3) Twitter Co-Occurance Single: single input showing the coocurance of words in twitter data
4) Twitter Co-Occurance Combined: combined input showing the coocurance of words in twitter data
5) NYTimes Norm Single: single input file wordcloud for NyTimes data
6) NYTimes Norm Combined: combined input file wordcloud for NyTimes data
7) NYTimes Norm : all the combined files wordcloud for NYTimes news data
8) NYTImes Co-Occurance: Showing the coocurance of words in Nytimes news data

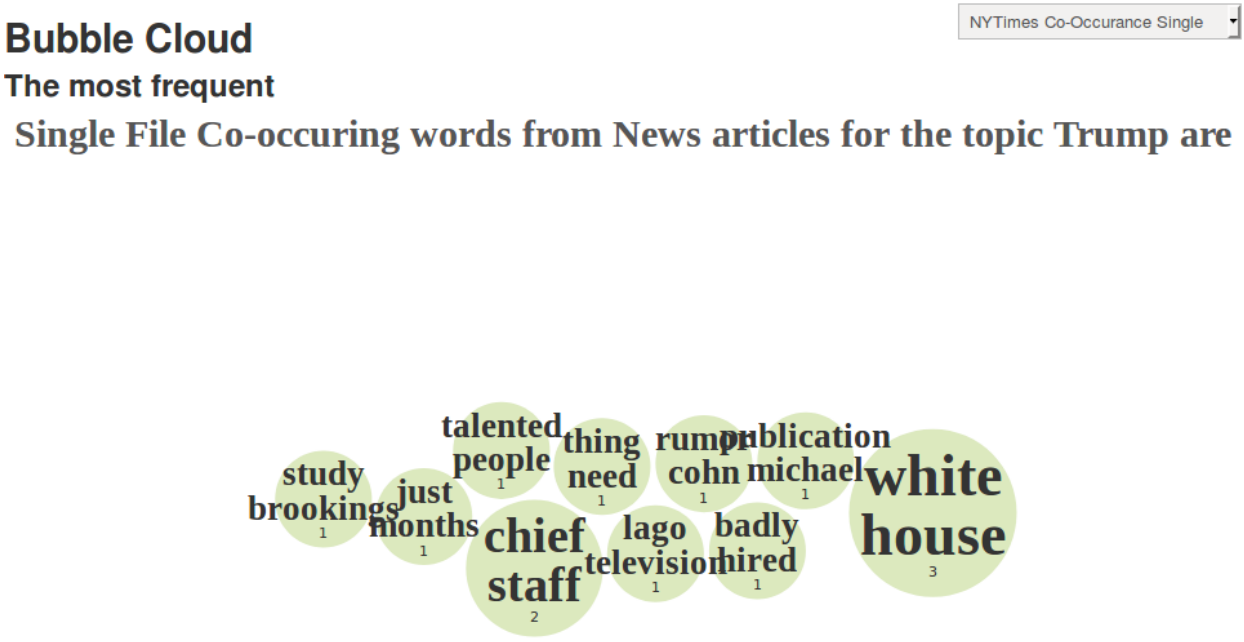
h.) We want to drill deeper into our analysis. Using the smallest data sets you collected in step a), analyze each set (Twitter and News) word co-occurrence for only the top ten words. Assume 4 context for co-occurrence is the “tweet” in the case of TwitterData, and the paragraph of the news article in the NewsData. Your “map” function emits and your “reduce” function should collate the co-occurrences for the top ten words and output them in a suitable format.
Soln: As per the requirement firstly we run it for smallest data sets that is on the a single file and then on a file which combines all the input data in one. 4 context for co-occurrence is the “tweet” in the case of TwitterData, and the paragraph of the news article in the NewsData has been done. Also “map” function emits and “reduce” function should collate the co-occurrences for the top ten words and has been outputted in both Json and Csv formats
Input code files: Lab2->Part2->code
Output code files: Lab2->Part2->output

The top ten co-occuring words are as follows:

For NYTIMES data for a combined file:



For NYTIMES data for a single file:



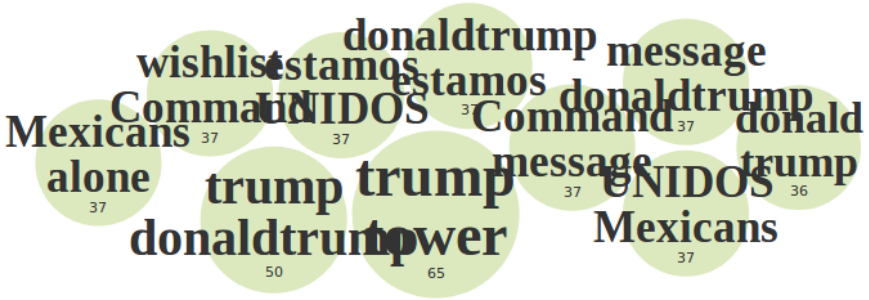
For Twitter data for a combined file:

Twitter Co-Occurance Combined

Bubble Cloud

The most frequent

Combined Files Co-occuring words in the tweets collected for keyword Trump are



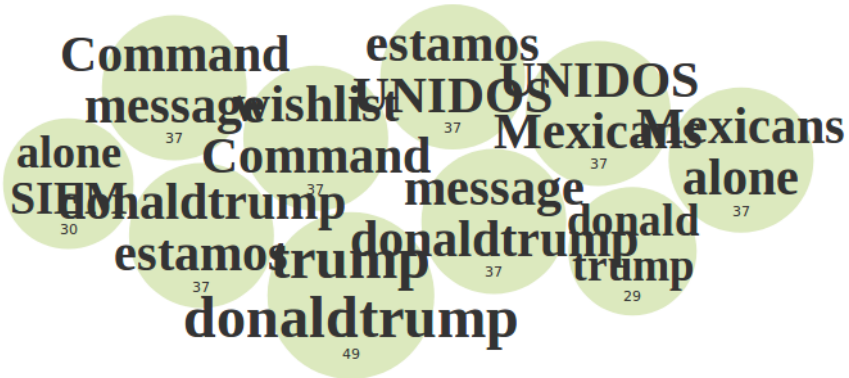
For Twitter data for a single file:

Twitter Co-Occurance Single

Bubble Cloud

The most frequent

Single File Co-occuring words in the tweets collected for keyword Trump are

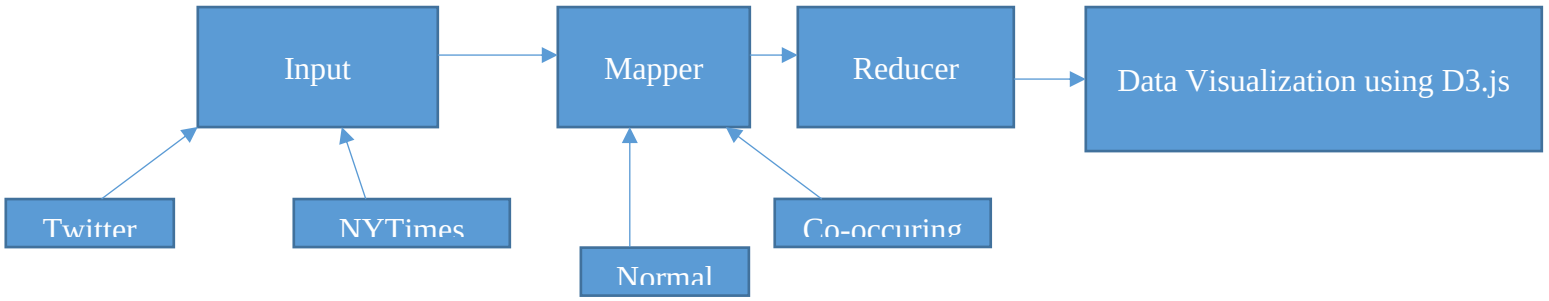


i. Document all the activities and how we can use your explorations and repeat them with some other data. Use block diagrams where needed. A well-organized directory structure is a requirement.

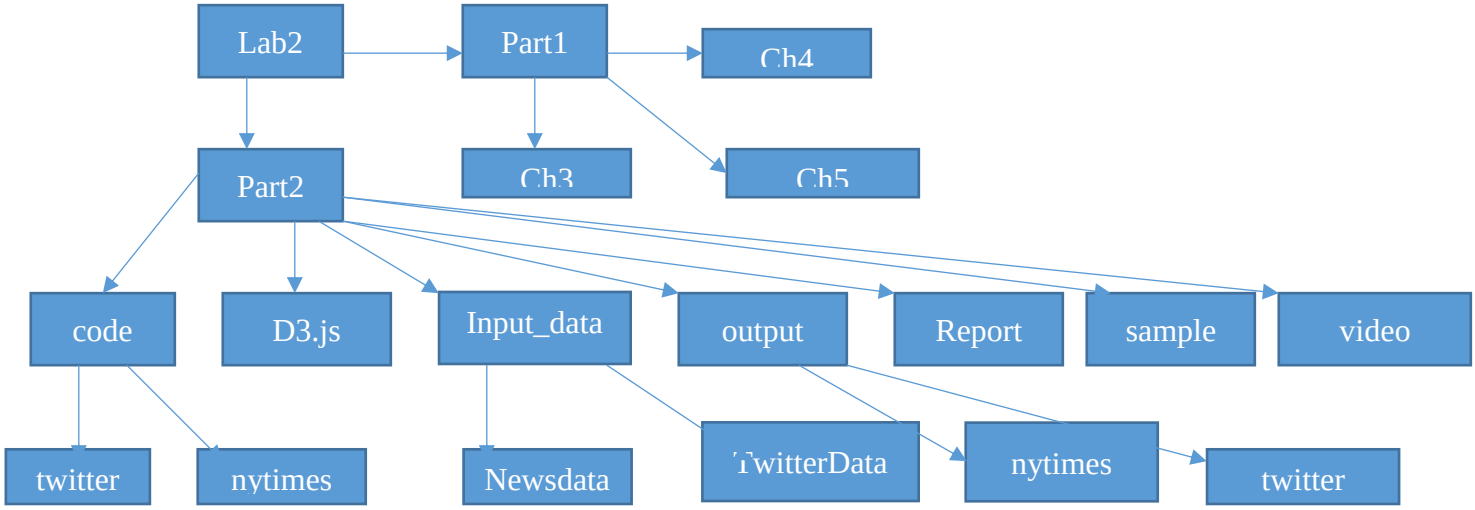
Soln:

The main processes are:

1) Data collection: from Twitter and NYTimes
2) Mapper: that filters out and removes unnecessary words from the data obtained this process is very important and most of the data is junk and for analysis we need clean data. It takes the input data cleans it, blocks the unnecessary data from going to the reducer phase and also produces output in key value pair which is required for the Reducer.
3) Reducer: It counts the occurrences of a particular word and outputs that word as key and count as value and the whole thing in a key value format.
   For all the data top 20 words with highest count were selected and in the co-occurances case top 10 was selected as stated in document.
4) Visualization using D3.js-> The output obtained was visualized in the form of wordcloud using D3.js, here greater the size of bubble means more is the count of that word.
   Here is a block diagram:



Here is the directory structure:



j.) A short video that explains your data analysis and visualization process.
Soln: The link to the video is below and in Lab2->Part2->video and goes by the name info.txt

https://buffalo.box.com/s/4bblnqhtwnw2soja1w9qdi4vymh69aj0