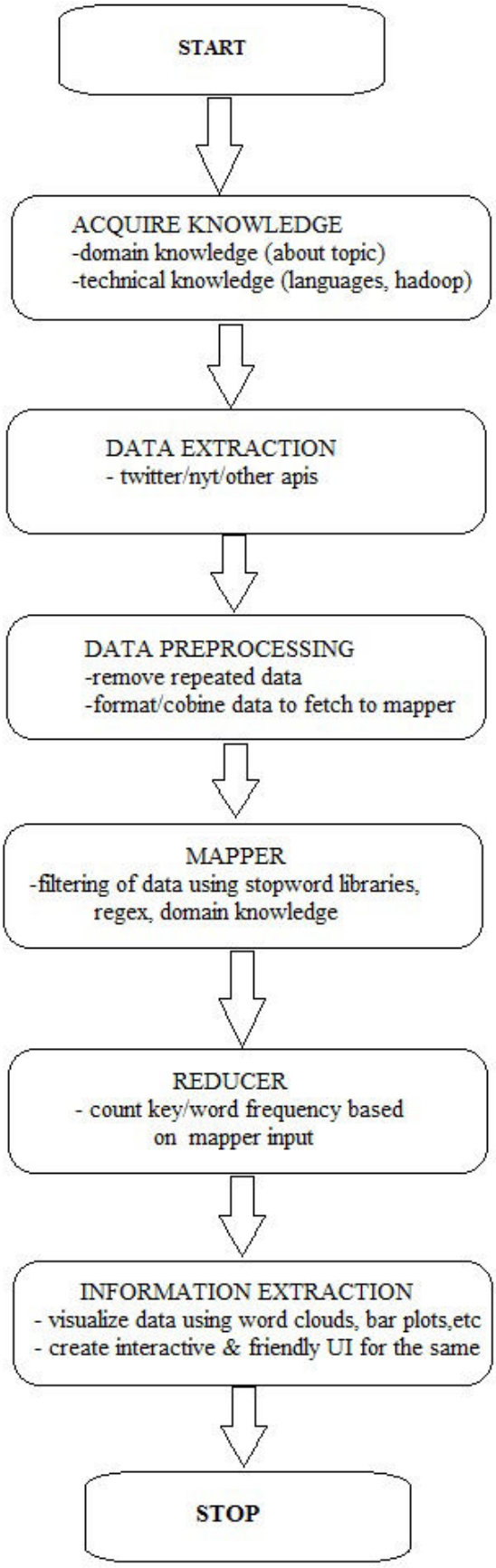
CSE 587 LAB 2: MARCH FOR OUR LIVES

- Gautam Shende (50245840)
- Sujit Singh (50247206)



ACTIVITES FLOW CHART

Details about how we achieved these steps in our project:

1) Building a strong foundation in Python:
We are using python from start to end so the only language we needed to master was python. Since we already had a decent hold over the language, doing the part 1 was sufficient to revise and brush up the concepts along with help on the fly from the official documentation and other online sources like stackoverflow.

2) Extraction of data from various sources such as Twitter, NY Times and other apis:
For tweets we used the twittersearch package in python to access the api and retrieve tweets. For articles, we started with 5 manually extracted articles for a prototype build and then moved on to the NY times api to extract articles. We were able to extract close to 50,000 tweets and 50 articles with close to 20% and 30% uniqueness respectively.

3) Data preprocessing and filtering without the mapper:
Now there was some basic filtering that needed to be done before we did the final processing and cleaning in the mapper. The only filtering done here is that we have removed the repeated articles and tweets.

4) Hadoop Mapper:
The mapper used regular expressions to extract words. From the extracted words, we removed stop words using the nltk library and seo word library. We also removed special words using domain knowledge like some words like amp were extracted due to html utf8 formatting so we added that to our set of stop words.
Logic for co-occurence:
We took at top 10 words at a time, once from articles and then from tweets and from pairs. These pairs from our keys for the mapper and it emits 1 only if both the words in the pair are present in a tweet/article paragraph.

5) Hadoop Reducer (Word Count):
The reducer just adds up all the occurrences for a key and then sorts based on frequency. It emits the count based on the frequency of a given key in decreasing order.

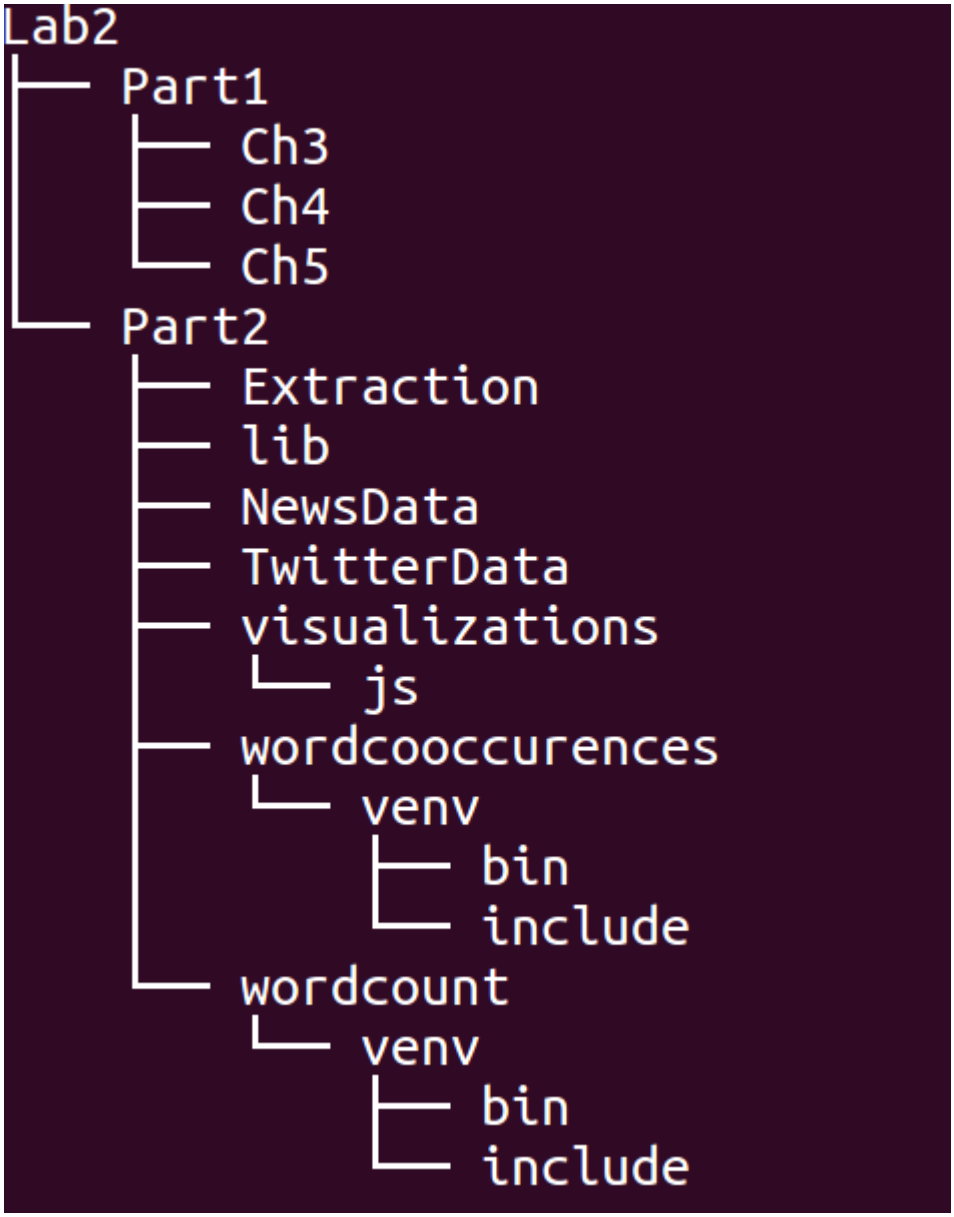
6) Data Visualization and information extraction on an interactive UI:
We have created an interactive website using html css javascript framework d3.js for visualization of the information.

It has 2 tabs:

- First tab is for wordcloud visualization which helps compare different information from different data sources by displaying word clouds side by side. User can choose different wordclouds to be compared.

- Second tab displays different bar plots for frequency of words obtained from different sources. User can select the various options for which they want to see the bar plot for frequency of words. The web page also allows user to type in a word and the frequency or count of that word will be dislayed.

Directory Structure:



The directory structure is divided into two main parts.

Part1:
The directory containing the implementation of python programs and snippets from the data-science handbook.

It has 3 sub folders – one for each chapter and its snippets in the form of jupyter notebooks.

Part2:

This folder is for the main project. It has various sub directories as explained below:

Extraction: This directory contains one extraction notebook. The language used is python3. The notebook has multiple snippets to extract tweets and nytimes articles.

NewsData: This directory consists of all the data (input / output) of the map-reduce for articles collected form New York Times API. The input contains the string “Extract” in it.

TwitterData: This directory consists of all the data (input / output) of the map-reduce for tweets collected form twitter API. The input contains the string sequence “Extract” in it.

Wordcount: This consists of the mapper and reducer files for word-count. Data cleansing operation is done by the mapper.

Wordcooccurences: This consists of the mapper and reducer files for finding co-occurences.

Visualizations: This folder consist of the d3.js codes to visualize outputs. The visualization is done using wordcloud and bargraphs. You can also search for the words with their associated counts using a search box. The output of this section is shown below:

#MarchForOurLives Visualization Word Cloud

Visualize Articles co-occurences for week 1

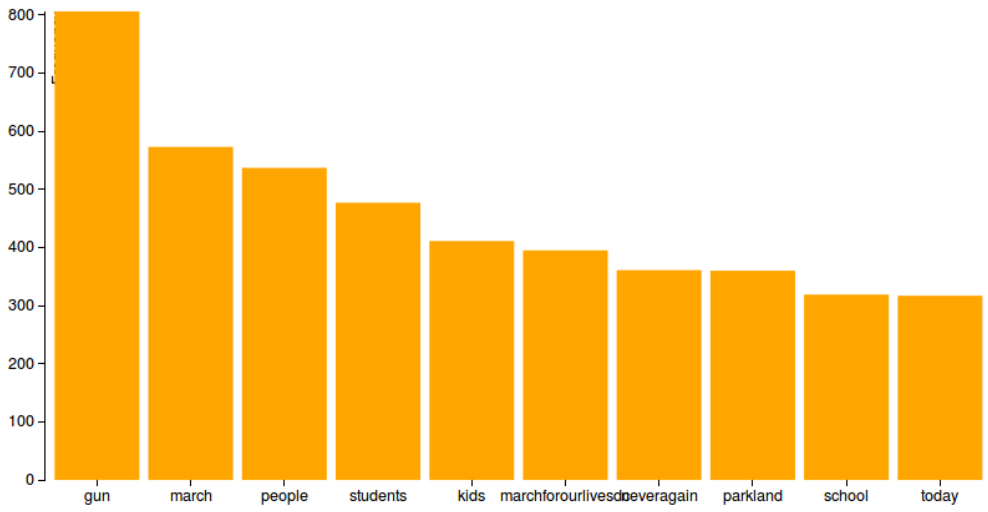
Word - Cloud A

Word - Cloud B

#MarchForOurLives Visualization Word Cloud

Select Query

input tags to search



We made some bash scripts to make our execution of the hadoop commands easier which are described in brief below:

namenode-format: To format name node
hadoop-start: To start hadoop
hadoop-stop: To stop hadoop
dfs-mkdir: To make a new directory in hadoop hdfs
dfs-put: To put files in the directory existing in hadoop hdfs
dfs-remove: To remove files from hadoop hdfs
save-output: To save the output in the desired location