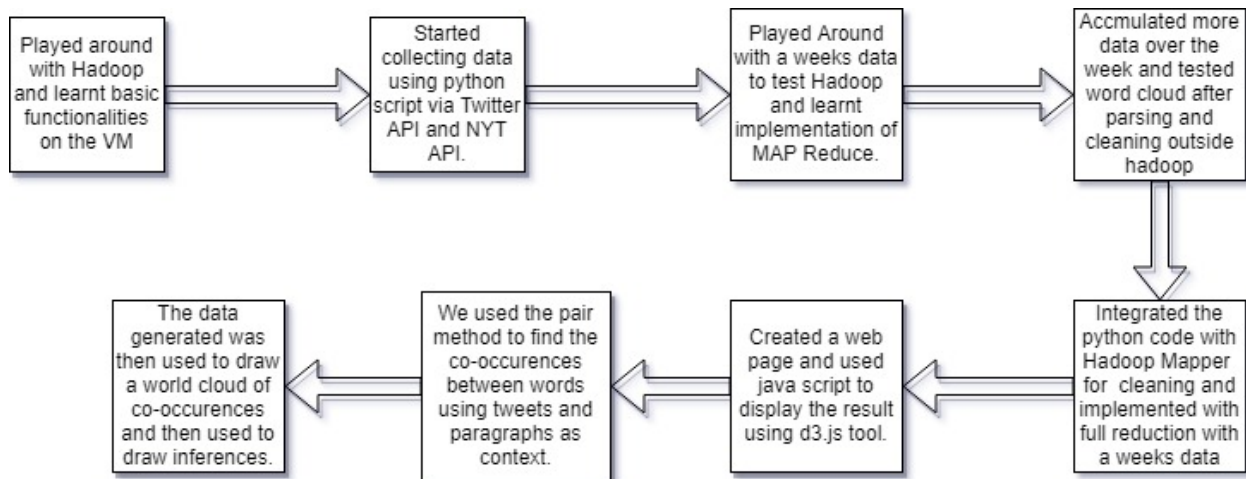


# Lab 2 Report

Suhit Datta

Sourav Ranu

The idea of this lab was to use Hadoop and experience how big data processing is done using Map Reduce on Hadoop infrastructure. The entire exercise can be suitably explained using the following block diagrams where each block symbolizes each step in the project implementation as described in the documentation.



This methodology can be easily reincorporated with any other data set and can be used to draw statistical inferences.

The procedure can be roughly divided into 4 steps:

- 1) Data Collection
- 2) Data Preprocessing
- 3) Data Post-Processing
- 4) Data Presentation

Note: The Data Post-Processing occurs after the output is received from the Map – Reduce phase.

Here we want to analyze the recent **Cambridge Analytica** data breaching incident that occurred.

### **Data Collection Phase:**

In this we explored two sources:

#### 1) New York Times

The data was collected using the NYT API using a relevant API Key. It returns a JSON object which needs to be processed to obtain the text. We collected a day of data and a week of data for the filter text: 'analytica' and by passing the relevant start date and end date in the URL.

You can find the following files in the same directory as this Report:

**NYTimesJsonExtract\_1DAY.pynb**  
**NYTimesJsonExtract\_FULL\_WEEK.pynb**

These files obtain the links of articles from the JSON object, scrape the data inside using BeautifulSoup and write to a file in the same directory.

**DATA**  
**NYTimesTextExtractFromBeautifulSoup.txt**  
**NYTTextExtractFullWeek.txt**

#### 2) Twitter feeds

A similar approach has been done to obtain the Twitter data. For obtaining a tweets for a week and a single day, we have referred to the Python script that has been used in Lab1.

The following codes have been used:

**Tweepy.ipynb** (to generate for a single day of tweets data)

**getTwitterJSONDataFromHashTag.pynb**  
**getTweetTextFromJSONData.pynb**

**DATA**  
**twitter1dayData.csv**  
**tweetTextExtract.txt** <- this comprises for the whole week

The data for the whole week was obtained as a JSON file namely **tweets.json**. This has been converted to **tweetTextExtract.txt** using the **getTweetTextFromJSONData.pynb** code.

### **Data Preprocessing Phase:**

In this step, we wrote the code for Preprocessing inside the Mapper as specified by the Instructor. It includes **Removing any symbols** such as @,# etc , **Removing Stop Words** and **Stemming** the words (for eg : converting spinning to spin)

This pre-processing has been done for the Mapper that generates the frequency of a single word and for the co-occurring words as well.

It can be found out in the following codes:

**mapper.py**

**mapper\_cooccurrence.py**

Likewise, the code for the Reducer is in the following file:

**reducer.py**

These two Python codes are to be run on the Hadoop Infrastructure.

An example could be as follows:

```
hdfs dfs -put $HOME/"location of input path or file " input
```

```
hadoop jar /home/hadoop/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.4.jar -  
mapper 'python3 mapper.py' -reducer 'python3 reducer.py' -input input -output output
```

For the sake of organization, the data can be placed in a directory such as NewsData and TwitterData in the Hadoop infrastructure.

The data obtained as a part of Step 1 needs to be mapped as the input for MapReduce Program to Run.

Once the output is generated, it needs to be written to a file or directory.

### **Data Post-Processing Phase :**

The output data obtained from the Reducer needs to be converted to a Javascript variable for it to be the input to the WordCloud in D3.js

A custom python code has been written for the same:

**genWCjsFile.ipynb**









For example, **TwitterFull.js** is generated when I provide an input of **TwitterFull\_part-00000** as obtained from the **Reducer**.

Thus the key value pairs are converted to relevant JS format file which serves as the input to the HTML code corresponding to running WordCloud using D3.js

All the HTML codes and the corresponding Javascript inputs can be found at:

\d3-wordcloud\wordclouds\_code directory.

You can run any of the following HTML codes to generate the WordCloud :

|  |                   |                      |      |
|--|-------------------|----------------------|------|
|  d3_WC_NYT_1DAY                     | 4/7/2018 12:23 AM | Chrome HTML Document | 1 KB |
|  d3_WC_NYT_FullData                 | 4/7/2018 9:24 PM  | Chrome HTML Document | 1 KB |
|  d3_WC_Twitter_1DAY                 | 4/7/2018 5:11 PM  | Chrome HTML Document | 1 KB |
|  d3_WC_Twitter_FullData             | 4/7/2018 7:15 PM  | Chrome HTML Document | 1 KB |
|  d3_WCCoOccur_NYT_1DAY              | 4/7/2018 11:37 PM | Chrome HTML Document | 1 KB |
|  d3_WCCoOccur_NYT_1DAY_FullData     | 4/7/2018 11:18 PM | Chrome HTML Document | 1 KB |
|  d3_WCCoOccur_Twitter_1DAY          | 4/7/2018 5:40 PM  | Chrome HTML Document | 1 KB |
|  d3_WCCoOccur_Twitter_1DAY_FullData | 4/7/2018 10:59 PM | Chrome HTML Document | 1 KB |

### Data Presentation Phase:







In Order to present our work, we created a portal: You need to open **d3-wordcloud\wordclouds\_code\portal.html**

As shown below:



These images have interactive hyperlinks.

You can actually view the obtained Wordcloud in the directory where this report is located:  
Here are the names:

|   |                   |          |        |
|---|-------------------|----------|--------|
|  OneDay_CA_NYT             | 4/7/2018 12:28 AM | PNG File | 284 KB |
|  OneDay_CA_NYT_CoOccur     | 4/7/2018 12:45 AM | PNG File | 298 KB |
|  OneDay_CA_Twitter         | 4/7/2018 5:13 PM  | PNG File | 313 KB |
|  OneDay_CA_Twitter_CoOccur | 4/8/2018 2:07 PM  | PNG File | 272 KB |
|  OneWeek_CA_NYT            | 4/7/2018 9:39 PM  | PNG File | 320 KB |
|  OneWeek_CA_Twitter        | 4/7/2018 7:17 PM  | PNG File | 313 KB |

A video has been made explaining the process:  
Can be found here :

<https://buffalo.box.com/s/uq0r9rztwzn33r7piocz8z63kwu31jkf>