Lab 2

DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION

By

Priyanka Manoj Naik

50248591

Goals:

1. Data aggregation from more than one source using the APIs exposed by data sources,
2. Applying classical big data analytic method of MapReduce to the unstructured data collected, and
3. Building a visualization data product

Abstract:

Through the use of various tools, we select a trending topic, in this case “Immigration”, and collect relevant data that we subsequently explore.

I plan on implementing this project in three parts. First and foremost, I need a data wrangling technique for which I will use python, Tweepy, and the Twitter API to collect data from Twitter. In order to collect data from NYTimes, I will be using nytimes provided API. I will refactor the data into a predefined format and make sure to leave only relevant information within, removing stop words, links, and other unnecessary parts. I will then proceed onto the map reduce portion of the project where I shall implement the word and its frequency as well as word count, co-occurrence word and their frequency. Lastly, I will represent this data in a word cloud with commonly used words larger and slightly faded in colour while less common words smaller and darker.

Project Objectives:

1. **Explore** a high-level programming language for data collection and analysis. This will be Python with its popular libraries such as “pandas”.
2. **Choose** a topic of interest to you. Here in this case, I have selected “Immigration” as my topic of analysis
3. **Aggregate** data from multiple sources to corroborate any findings and outcomes of data analysis.
4. **Install** a virtual machine (VM) image for data storage in HDFS and Hadoop infrastructure.
5. **Familiarize** yourself with **MapReduce (MR)** model and programming using this model.
6. **Code** solutions in Java or Python to process data in <key,value> format using MR model
7. **Visualize** the outcome of the MR analysis using “wordcloud” visualization tool.
8. **Compare** the outcomes of the same analysis for at least two different sources: first an opinion social media source such a twitter and the second one, a reliable researched source such as NYTimes.
9. **Create** a responsive web interface (web tool) for visualizing the outcome of your analysis. I will be using d3.js for representing data in word cloud.

Project Approach:

Part 1:

Exploring high-level programming language for data collection and analysis. I have been using Java for around 4+ years and wanted to learn Python. So, I started with the chapters 3,4 and 5 from the book. Here, I learnt on data collection, using APIs, storing of data in files, reading files for data, using pandas for handling data frames, plotting graphs, etc.

Chapter 3 - NaikLab2/Part1/Ch3
Chapter 4 - NaikLab2/Part1/Ch4
Chapter 5 - NaikLab2/Part1/Ch5

Part 2:

Choosing Topic of current interest to people in USA
Topic selected – ***"Immigration"***

VM Installation and check for the working of Hadoop on the machine
Step1: I started with VM installation and installing Hadoop on the machine.
Step2: Tested basic commands with the sample data provided

Scripts for collecting data
Step1: Check how the NYTimes API works
Step2: Wrote code in Python to collect data from twitter and nytimes APIs
Step3: The data is aggregated over a period of one week and placed in directories TwitterData and NewsData. The data here is stored in json format.

Twitter Data – NaikLab2/Part2/PythonCode/collect-twitter-data.py
News Data – NaikLab2/Part2/PythonCode/collect-news-data.py

Result:
News Data – NaikLab2/Part2/NewsData.zip
Twitter Data – NaikLab2/Part2/TwitterData.zip

Code for MapReduce
Step1: Wrote code for map in python whose primary work was to parse and clean the data collected from both the APIs, remove stop words. The map emits word and count 1.
Step2: Reduce counts the number of useful words and combines the count. The output of this is json file with word and its count stating the occurrence of the word in the tweets and news data.

Mapper for Twitter Data – NaikLab2/Part2/PythonCode/mapper.py
Mapper for News Data – NaikLab2/Part2/PythonCode/mapper-news.py
Reducer for Twitter and News Data – NaikLab2/Part2/PythonCode/reducer.py

Result:
News Data – NaikLab2/Part2/D3Project/newsdata.js
Twitter Data – NaikLab2/Part2/D3Project/twitterdata.js

Visualize data
Step1: See the code for d3.js and how the word list can be plugged and played in the html for representation
Step2: Tested the word cloud with sample data

D3.js Project – NaikLab2/Part2/D3Project

Code for MapReduce for word co-occurrence

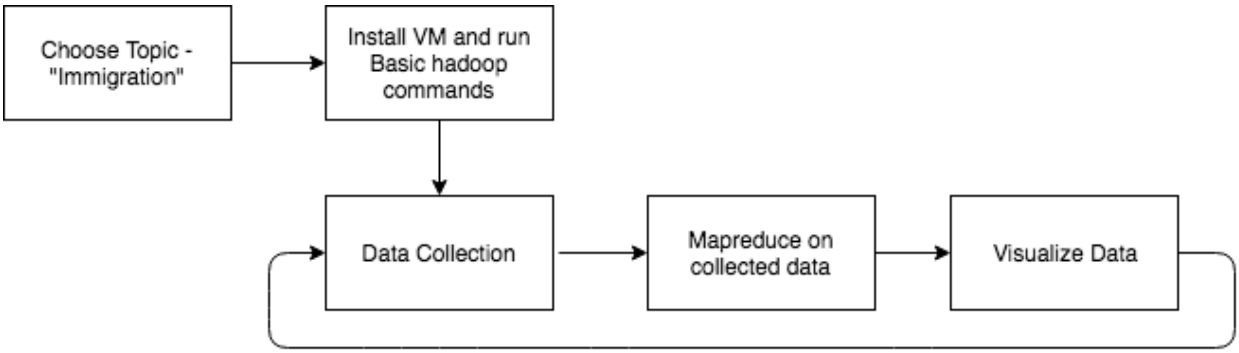
Step1: Wrote code for map in python whose primary work was to parse and clean the data collected from both the APIs, remove stopwords. The map emits word, its co-occurring word and count 1.

Step2: Reduce counts the number of useful words and combines the count. The output of this is json file with word and its co-occurring word and count stating the how the terms where related to each other and their occurrence in the tweets and news data.

Mapper for Twitter Data – NaikLab2/Part2/PythonCode/mapper-co.py
Mapper for News Data – NaikLab2/Part2/PythonCode/mapper-news-co.py
Reducer for Twitter and News Data – NaikLab2/Part2/PythonCode/reducer-co.py

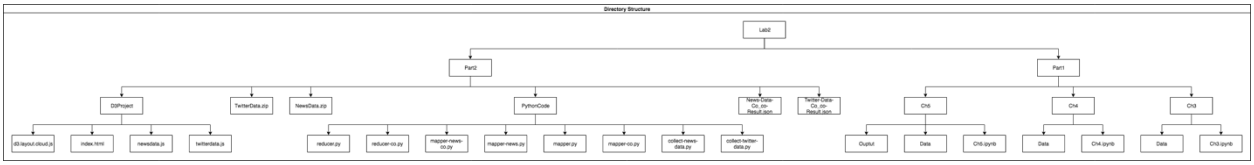
Result:
Mapper for News Data – NaikLab2/Part2/News-Data-Co_oc-Result.json
Mapper for Twitter Data – NaikLab2/Part2/Twitter-Data-Co_oc-Result.json

Flow of the project:



Directory Structure for the Project:

https://drive.google.com/open?id=1tx1KuWSO0RtFti1NCeSmd6_0Ff4S2lHW



Amount of Data Collected:

NYTimes Data collected – **250 news reports**
Twitter Data collected – **710343 tweets**

Sample Outputs:

Map reduce for word and its count - TwitterData


```
[
  {
    "term": "immigration",
    "freq": 47
  },
  {
    "term": "trump",
    "freq": 18
  },
  {
    "term": "ice",
    "freq": 14
  },
  {
    "term": "illegal",
    "freq": 8
  },
  {
    "term": "mexico",
    "freq": 8
  },
  {
    "term": "president",
    "freq": 4
  },
  {
    "term": "migration",
    "freq": 6
  },
  {
    "term": "uses",
    "freq": 6
  },
  {
    "term": "wall",
    "freq": 7
  },
  {
    "term": "oakland",
    "freq": 5
  }
```

```
]
```


Map reduce for word and its count - NewsData


```
[
  {
    "text": "immigration",
    "size": 38
  },
  {
    "text": "trump",
    "size": 53
  },
  {
    "text": "president",
    "size": 46
  },
  {
    "text": "donald",
    "size": 28
  },
  {
    "text": "immigrants",
    "size": 19
  },
  {
    "text": "border",
    "size": 17
  },
  {
    "text": "administration",
    "size": 17
  },
  {
    "text": "government",
    "size": 18
  },
  {
    "text": "said",
    "size": 31
  },
  {
    "text": "new",
    "size": 21
  }
]
```


Map reduce for word, co-occurring word and its count - TwitterData

```
[
  {
    "co_term": "trump",
    "term": "immigration",
    "freq": 1093
  },
  {
    "co_term": "trump",
    "term": "president",
    "freq": 878
  },
  {
    "co_term": "visa",
    "term": "donald",
    "freq": 563
  },
  {
    "co_term": "trump",
    "term": "h1b",
    "freq": 326
  },
  {
    "co_term": "trump",
    "term": "donald",
    "freq": 204
  },
  {
    "co_term": "president",
    "term": "immigration",
    "freq": 198
  },
  {
    "co_term": "deport",
    "term": "immigration",
    "freq": 167
  },
  {
    "co_term": "trump",
    "term": "policies",
    "freq": 143
  },
  {
    "co_term": "trump",
    "term": "administration",
    "freq": 78
  },
  {
```

```
    "co_term": "president",
    "term": "border",
    "freq": 34
  }
]
```

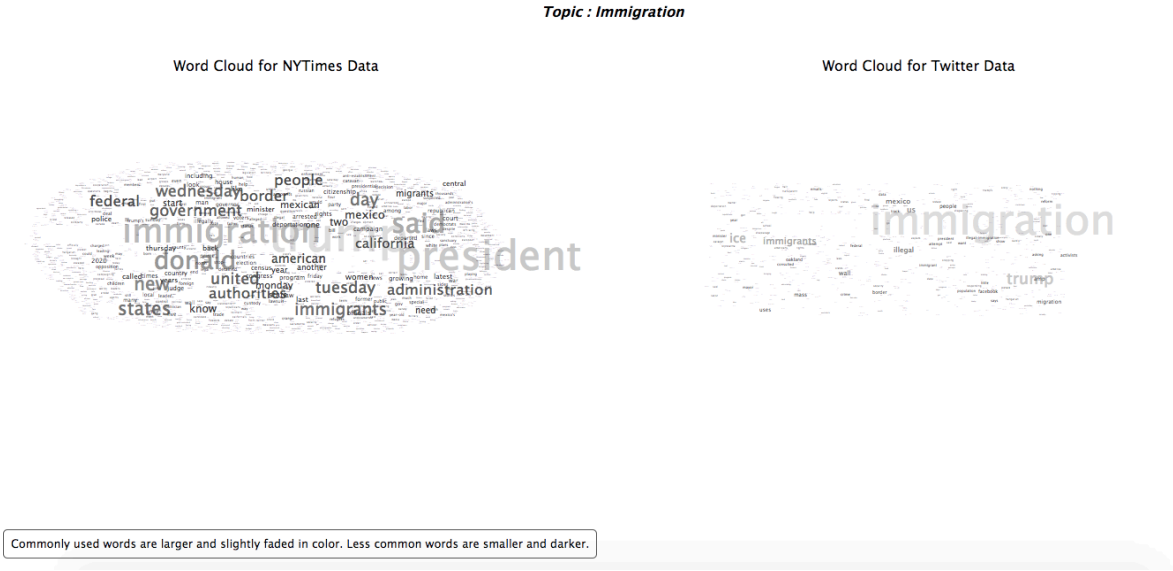

Map reduce for word, co-occurring word and its count - NewsData


```
[
  {
    "co_term": "trump",
    "term": "president",
    "freq": 130
  },
  {
    "co_term": "trump",
    "term": "immigration",
    "freq": 113
  },
  {
    "co_term": "president",
    "term": "donald",
    "freq": 98
  },
  {
    "co_term": "trump",
    "term": "donald",
    "freq": 92
  },
  {
    "co_term": "trump",
    "term": "said",
    "freq": 80
  },
  {
    "co_term": "said",
    "term": "immigration",
    "freq": 75
  },
  {
    "co_term": "president",
    "term": "immigration",
    "freq": 71
  },
  {
    "co_term": "trump",
    "term": "border",
    "freq": 65
```

```
  },
  {
    "co_term": "trump",
    "term": "administration",
    "freq": 60
  },
  {
    "co_term": "president",
    "term": "border",
    "freq": 58
  }
]
```


Data Visualization:

Word Cloud



Ease of Plug and Play:

The scripts written for data collection can be used for different topics just by changing the search term in the code.

The map reduce output for word and its count is a json which can be directly used in the index.html file to display the word cloud for that term.

Demo Video:

https://drive.google.com/open?id=1Fias0QJrElZCoK5h6luzz28NDzc8qIYc