Data Aggregation, Big Data Analysis and Visualization

Harshal Ganesh Jagtap
Compute Science and Engineering
University at Buffalo
UBIT: harshalg
Person #: 50290606
harshalg@buffalo.edu

Bhagyashri Thorat
Computer Science and Engineering
University at Buffalo
UBIT: bthorat
Person #: 50290581
bthorat@buffalo.edu

Abstract

The goal of the project is to collect data from different sources, use map reduce algorithms as analysis tool and visualize the analysis using Tableau. We have aggregated from data from three different data sources: Twitter, New York Times and Common crawl using various methods. The data is aggregated on various sub-topics of one main topic. The main and subtopic are as listed below:

- Movies and Series (Main topic)
  - Marvel cinematic universe
  - Game of Thrones
  - Star Wars
  - Disney
  - Drama Movies

1 Data Collection

We have collected data from three different sources namely Twitter, New York Times and Common Crawl. We have written separate python scripts for data collection from each sources. We have used APIs provided by source websites to fetch data. Data has been collected for each of the subtopics individually and then merged together to form one big dataset. This dataset is cleaned and provided as input the map reduce code on Google Cloud DataProc cluster.

2 Data Cleaning

The collected data cannot be used directly as input to map reduce due to following reasons:

- Data collected form all data sources contain stop words like ‘in, for, the, would’ etc.
- Data collected does not contain words in their natural format. That means the base word ‘rock’ may have different derived words like ‘rocking, rocks, rocker’.
- Data collected from sources like Twitter contains non- ASCII elements like emoticons. Presence of emoticons should not have any significance towards final word count.

Stop words removal:

We have removed stop words from collected text during the mapper phase. If the mapper sees a word that is present in the predefined list of stop words, then the mapper just ignores such word. The predefined list of keywords is taken from NLTK's own set of stop words. After multiple runs of map-reduce, we have added our own set words in the stop words list.

Lemmatization:

In order convert the collected words to their own natural format and take out context from collected words, we have used NLTK's WordNetLemmatizer(). When the data is collected, we have passed it through a separate phase of data cleaning which involves emoticon removal and lemmatization.

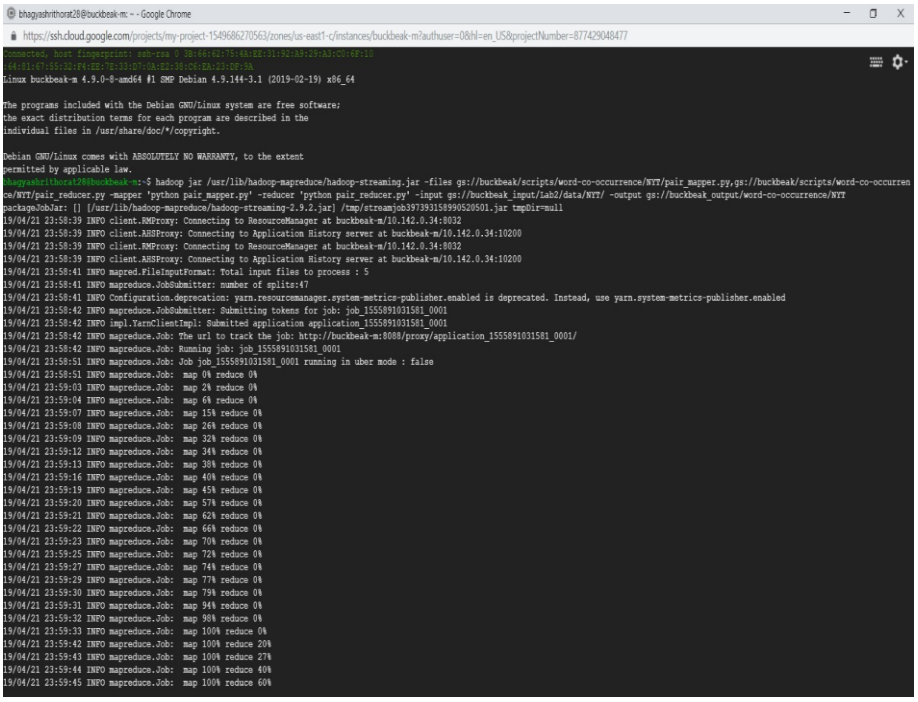
Emoticon removal:

Text form data sources like twitter contain a lots of emoticons which should not be counted for the final word counts and co-occurrences. To remove emoticons, we have a set of UTF-8 encodings of emoticons. We have ignored any words with UTF-8 encoding is present in this set of encodings. Hence, the final text generated before passing to map-reduce is free of any emoticons, stop words and the words are in their most natural form which is ideal for our word count and co-occurrence algorithms.

3 Running Map-Reduce on Google Cloud DataProc

To run MR on DataProc we have followed these steps:
1. Created a project
2. Set up billing account
3. Created a DataProc cluster and set the appropriate configuration as needed
4. Ssh to the master node. And run the command:

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -
files
gs://input_directory/mapper.py,gs://input_directory/reducer.py
-mapper 'python mapper.py' -reducer 'python reducer.py' -input
gs://data_input_directory/data4prac/ -output
gs://output_directory/Output
```



5. We have downloaded the generated output files of map reduce to our local file system.

4 Post Processing

The map reduce on DataProc generates multiple files which contains pairs of individual words or word pairs and their respective counts. We have to take top 10 words or word pairs across all these files based on their counts. We have written a separate python script to perform this task. This file produces three separate files for three different data sources containing top 10 counts or co-occurrences across all files of the data source. This output file contains the final top 10 word counts and co-occurrences and can be used for data visualization in tableau.

5 Visualization in Tableau

We have used Tableau as visualization tool to generate word cloud. We have created 6 different work sheets in tableau each for separate source and output type. We have published our work sheets using Tableau online and embedded our visualizations on a simple HTML webpage. The webpage file resides in the ‘webpage’ directory.