Data Intensive Computing
Lab 2 Report

Submitted by

1. SAJID KHAN – 50248743
2. VINAY VARDHAMAN – 50248852

Part 2:

a. The topic we chose is about **Blockchain** and **Cryptocurrency**. Data is collected from twitter and New York Times Articles. We initially collected sample data for a day and later on proceeded to collect data over a week. We have also tweaked the keywords by using **bitcoin** to get better results.
   1. **Twitter** – Used "**twitteR**" library for data collection. Collected data of about 30,000 tweets over a week, filtered all the retweets and duplicates. We have a batch of 10,000 tweets for each keyword bitcoin, blockchain and cryptocurrency separately.
   2. **New York Times** – Collected data using the Articles API by fetching the URL's for the specific query keywords and used those URL's to scrap paragraphs from their web pages using **Beautiful Soap** python library. Data over 3 months was collected.

b. Used standard Hadoop VM appliance suggested on the piazza, tested basic commands on the sample data provided.

c. Data collection
   1. **Twitter** - Saved the tweets into a CSV file and wrote code to copy tweets into a separate text file which is then input to a Map Reduce Job.
   2. **NY Times** – Fetched the meta data for a query keyword using the New York Times Article API, parsed the meta data for URLs. Using Beautiful Soap python library, fetched the content from the URL's using web scraping. The article text is saved locally and then input to a Map reduce Job.

d. **Data processing** – Used Java for coding Map Reducer job. The Mapper takes input in the form of text files and outputs aggregate results in CSV file format.
   1. **Word Count MR**: The Mapper takes the input file and emits 1 for every word occurrence. These emitted words are later summed up to their frequency in the reducer and output in a .csv file with word matching its frequency. Did some cleaning operation by removing unwanted text using regular expressions and stop words. Tweaked the output for better results by changing the keywords. Used words like bitcoin etc.
   2. **Word co-occurrence MR** – The Mapper accepts the raw text input and emits pairs of words for the Top 10 frequent words. Other words are filtered from emitting since they are out of the scope. As in the Word Count MR, input data is cleansed using regular expressions and filtering the stop words. A custom Java class called Pair is created for emitting pair of words. The reducer sums the frequency of every pair and stores the pair in a Priority Queue. Only the top 10

pairs are saved in the queue. The less frequent pairs are automatically removed. At the end of Map Reducer Job, clean methods is called. In the clean method, we emit the top 10 pairs in the priority queue along with their frequency. The output is then written to a csv file.

e. **Data Visualization** -
The data processed in the above step is used to draw the word clouds. Basic HTML page is developed. Local .csv files are loaded using D3.js library functions and processed the words and corresponding counts in it. Upon the selection of **Search Word** and **Type** (word count or co-occurrence) the corresponding word clouds from NY times Data and Twitter Data are displayed on the page.

f. Repeated the steps above for larger sets of data collected over a week for Twitter, collected over the past months for NY Times Articles. The difference observed is the convergence of top 10 words with more data.

g. Designed web page with multiple options to select the type (word count or co-occurrence) and choose the keywords among bitcoin, blockchain and cryptocurrency. Related word clouds are shown. At any point on the web page, two word clouds are displayed, one for twitter and other for NY Times. Visual comparison can be made on the trends of the topic from both the sources of data.

h. Wrote another Map reducer code for co-occurrence of words(pairs) in a tweet from Twitter data and a paragraph from the Articles. Word co-occurrence is performed only on the top 10 words in the data, rest are ignored. The output of co-occurrence MR is displayed as word cloud like the word count with suitable options to select the topics.

i. This document.

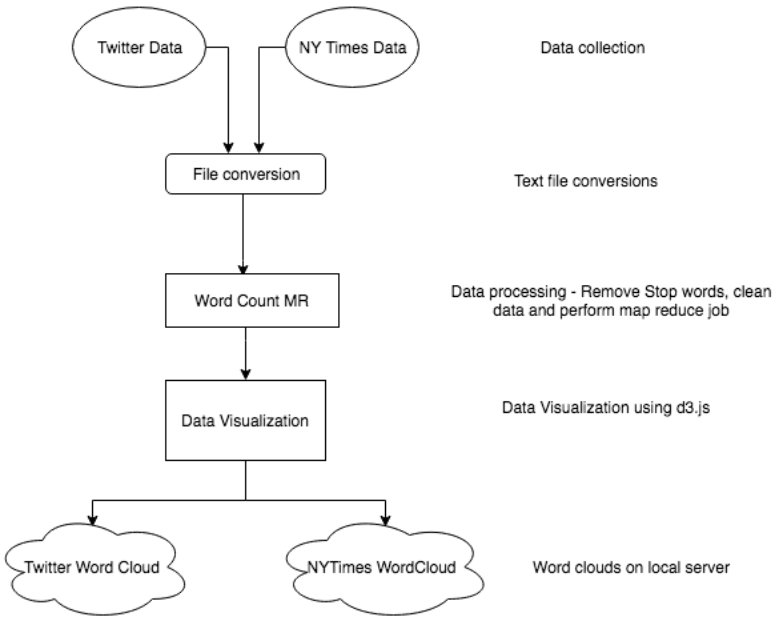
j. Reference to the video link
   i) https://youtu.be/MpJo43xLD6I
   ii) https://youtu.be/myAkX8e8ry4

Note: Flow-charts for word count and co-occurrence are attached at the end of the document.

References:

1. https://github.com/jiankaidang/RelativeFrequenciesHadoop/blob/master/src/Pairs.java
2. https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
3. https://developer.nytimes.com/
4. https://github.com/wvengen/d3-wordcloud

Word Count Flowchart



Word co-occurrence Flowchart