LAB2: DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION

Monica Vashu Kherajani (mkheraja)
Vaidehi Ajay Dharkar (vaidehia)
Data Intensive Computing
04/21/2019

Objective:

The purpose of this project is to collect the data from multiple sources using public APIs offered, to evaluate the data parallelly using MapReduce algorithm and thereby to compare the reliability of the data from different data sources.

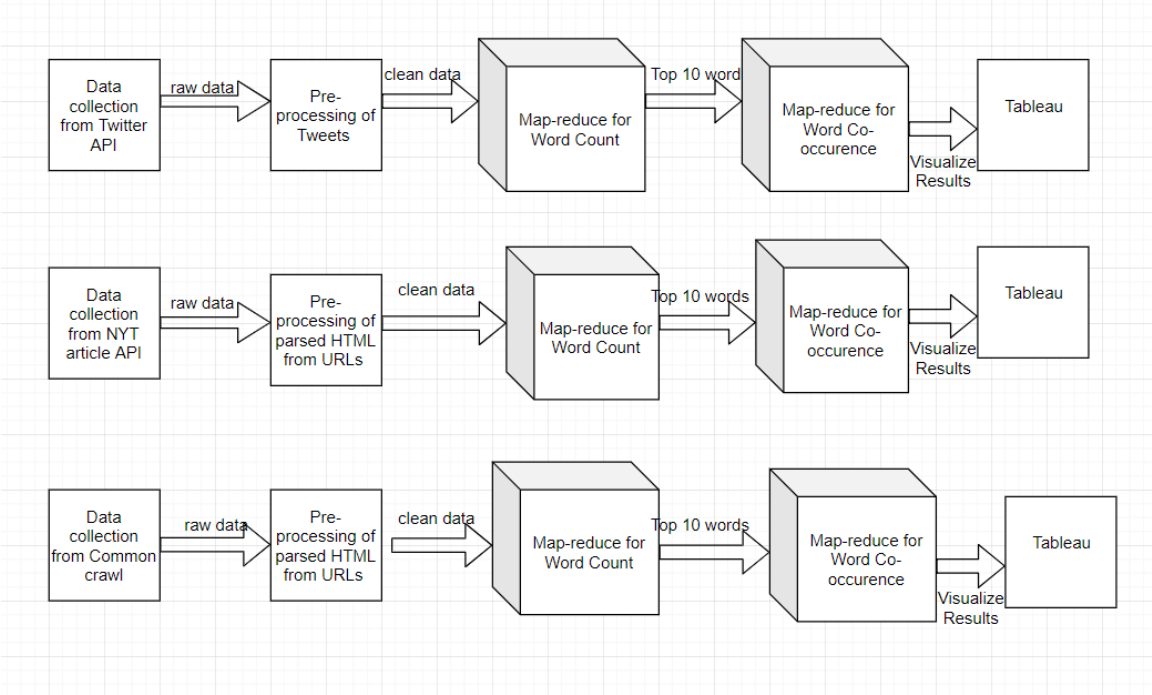
This is achieved in 3 stages:

- Data extraction and processing from
    - Social media source: Twitter
    - Reliable researched source: NYTimes
    - Common crawl
- Running MapReduce algorithm to evaluate the word count of top 10 words and their co-occuurrence with other words.
- Visualizing the outcome of MapReduce using word cloud.

The topic we have chosen for this project is 'Movies'. Topic is divided into 10 categories based on genres of movies:

- Action
- Comedy
- Crime
- Drama
- Fantasy
- Animated
- Horror
- Romance
- Social
- Thriller

The overall flow diagram:



Data Extraction and Processing

- As stated above, data is collected from 3 sources:

1. **Twitter data:**

- Tweepy API is used to collect twitter data. The User group considered for data collection is USA population.
- 23k tweets are collected by filtering user location as US, removing all retweets and repeated tweets.
- Keywords used for data collection:
    - Action movie
    - Comedy/ Humorous/ Parody/ Romantic comedy/ Comic fantasy/ Satire movie
    - Crime/ Suspense/ Courtroom drama/ Detective story/ Gangster/ Murder / mystery movie
    - Drama
    - Fantasy/ Fairy Tales/ Science fiction/ Sci-Fi/ Scifi
    - Animated/ Animation/ Anime
    - Horror/ Ghost/ Monster/ Vampire/ Slasher
    - Romance
    - Social
    - Thriller

2. **New York Times data:**

- Using articleAPI from nytimesarticle package
- performing API.search to get the required URLs
- parsing html data to get the relevant content using beautifulsoup

3. **Common crawl data:**

In the Common crawl code, we experimented with the various domains related to the movie keyword with index as 2019 in order to find out the recent articles. Webpages generated from valid responses are downloaded. Paragraph content for the webpage is searched for keywords related to 'movies'.

Data processing:

Raw data is generated from the above collection. This data has to be cleaned. Cleaning of raw data involves:

- Removing unnecessary tabs/ blank spaces
- Removing stopwords like numbers, 'a', 'an', 'the'. Stopwords are downloaded from nltk corpus.
- Removing special characters
- Lemmatization: which involves reducing the derived word to its original form. E.g. 'going' and 'goes' will be reduced to 'go'

Map Reduce

- Word-count parallel execution (Explained in video)
- Word-co-occurrence parallel execution (Explained in video)

Visualization

Visualizations of the obtained results are performed in Tableau and are published to its server. The following links are shared with professor and all the TAs:

Big data1:
https://us-east-1.online.tableau.com/t/momo/views/finalVisualization1/mainPage?iframeSizedToWindow=true&:embed=y&:showAppBanner=false&:display_count=no&:showVizHome=no

Big data2 – comparisons:
https://us-east-1.online.tableau.com/#/site/momo/views/finalVisualization1/ComparisonDashboard?:iid=11

Small data1:
https://us-east-1.online.tableau.com/#/site/momo/views/smallData/MainPage?:iid=1

Small data2 – comparisons:
https://us-east-1.online.tableau.com/#/site/momo/views/smallData/ComparisonDash?:iid=1

Comparison

Observation:

From the comparison of the word clouds for 3 data collection, some of the words like 'new' which are most talked about in all 3 domains.

ReadMe

- Directory structure explained:

  mkheraja.zip

- Report.pdf
- Video.mp4
- part1 (folder)
  - Code (folder) -> all scripts and codes for collecting data
  - Data (folder)
    - Twitter (folder) -> all tweets
    - NYT (folder) -> all NYT articles
    - Commoncrawl (folder) -> all CC articles
- part2 (folder)
  - mapper.py reducer.py .etc (project directory and dependencies required for running part2)
- part3 (folder)
  - Twitter (folder)
    - Code (folder) -> mapper.py, reducer.py
    - Images (folder) -> all visualizations
  - NYT (folder)

        - Code (folder) -> mapper.py, reducer.py
        - Images (folder) -> all visualizations
    - Commoncrawl
        - Code (folder) -> mapper.py, reducer.py
        - Images (folder) -> all visualizations
- Part2 reducer vs part3 reducer:
  In part2, reducer generates wordcount for all words emitted by mapper, whereas in part3, reducer generates wordcount for top 10 words so that their co-occurrence can be found out in the further steps.