DIC – Lab 2

Niranjan Mirashi **nmirashi(50289408)**
Aniket Raghuvir Rane **arane2(50290850)**

- The main aim of this project is to execute Word Count and Word Co-Occurrence programs using MapReduce
- We have executed this project in the following stages – Data acquisition, Data Cleaning, MapReduce Execution and finally, Visualization of results.
- In this project, we collected approximately 30,000 tweets (excluding duplicates and retweets), 683 articles from New York Times and a considerable amount of data from Common Crawl on the topic 'Sports'.
- We divided the main topic into five subtopics to narrow our search and to have an anticipated result.
- Our subtopics are –
    1. Soccer
    2. Tennis
    3. Golf
    4. Swimming
    5. Basketball

- As a result, we gathered a considerable amount of data as each of the above-mentioned subtopics are always trending in the United States.

PHASE 1 – Data Acquisition

- Twitter data:
  - We used the 'tweepy' package in python to extract tweets. In the mentioned package, we used 'tweepy.cursor' to specify the start date and end date of the required tweets. Also, we specified our word to be searched for, in this case, each sub-topic. To be precise, we collected a total of 29,425 tweets, which includes all non-duplicates and no retweets.
  - The tweets are collected in .csv format. We convert all files into one large .csv file and then convert the large file to a text file for simplicity.
  - As a result, we have a large text file of tweet-data with html tags, punctuation marks, spaces, etc. which we clean in the next phase.

- NYT data:
  - We obtained API keys from the developer NYT site.
  - We used the following packages – 'nytimesarticle','bs4','requests' and 'time'.
  - We mention the start and end dates of the query search in 'api.search'.
  - We obtain data from various nyt pages and then use an html parser to obtain data using 'Beautiful Soup'.
  - We use 'prettify()' to match indentation levels.
  - We then append the data to an array which is then written into separate text files for sub-topics.

- Lastly, we merge all text files resulting into one big text file containing NYT articles which is ‘dirty’. We clean this data in the next phase.

- Common Crawl data:
  - Common Crawl data set is huge, we first tackle the warc file CC-Main-2019-19/wet.paths.gz So here we have around 56000 files in this zip.
  - We need Warc Module to read CC-MAIN-20190318132220-20190318153617-00012.warc.wet.gz and Similar files.
  - So we import warc module, but first we need warc in python 3 to tackle the __built__ module error.
  - After this we need langdetect and islice, to trace the urls and find our keywords for sports, soccer, golf, tennis, swimming, basketball.
  - More than 550 articles that are matched with the keywords are then saved in respective text files, sports, soccer, golf, tennis, swimming, basketball.
  - Later we merge those files to get One file with all the data together and then we run the Mapper and Reducer on this file to get the word count and word co-occurences which we later use on the Tableau.

PHASE 2 – Data Cleaning

- This phase is common for all data sources.
- The aim of this phase is to make the data ready for further processing (MapReduce).
- This phase includes three sub-phases:
  - **Removing HTML tags, punctuation marks and non-English words/phrases:**
    - We use the 'regex' python library.
    - We also convert all words to lower case for precision.
    - The function 're.sub' is used to remove unwanted characters.
    - In the latter phase of data cleaning, we also use the function 'isalpha()' to be sure that there are no unwanted words.
  - **Lemmatization:**
    - We use the natural language processing python library – 'nltk'.
    - In this phase, we basically convert all word forms to their root words.
  - **Removing Stop-Words:**
    - We import the stop-words package from nltk. We also add a few of our own stop-words as this library doesn't contain all stop-words.
- As a result, we end up obtaining clean files from each of the three sources.

PHASE 3 – MapReduce

- We use Virtual Box and a VM image to deploy our Hadoop ecosystem.
- We import all the clean data files from our host machine and place them in new folders in the Hadoop file system.
- **Word Count:**
  - We use an existing jar file to execute this program.
  - In the mapper phase, we emit a count for every word in the data file which is stored on the local buffer for the reducer to combine.
  - The reducer then combines all the counts of matching words and emits total count for each word in the file.
- **Word Co-occurrence:**
  - Similar to word count, in this case, we find the count of the pairs of words.
  - However, we don't calculate the counts for each word. We use the top ten words from Word Count and consider the next and previous words forming pairs.
  - The reducer phase is similar to Word Count where we combine the counts for each pair, and emit the total counts for matching pairs.
- To get top ten words from each of the above programs, we use pandas to create data frames and sort them in ascending order according to their counts.

- We then obtain the top ten words with highest counts by doing ‘df.tail(10)’.
- We the write these data frames to their respective csv files.
- As a result, after this phase is completed, we obtain 6 different csv files – 3 for each source for word count and 3 for word co-occurrence.

PHASE 4 – Visualization

- We use Tableau for data visualization.
- We create word clouds for each of the above obtained csv files.
- We also create buttons to toggle between the 6 results.

References

- https://lintool.github.io/MapReduceAlgorithms/MapReduce-book-final.pdf
- https://dzone.com/articles/configuring-memory-for-mapreduce-running-on-yarn
- https://data-flair.training/blogs/top-hadoop-hdfs-commands-tutorial/
- https://docs.python.org/3/library/re.html
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- https://pypi.org/project/nytimesarticle/
- https://pythonspot.com/category/nltk/
- https://tweepy.readthedocs.io/en/v3.5.0/
- https://pandas.pydata.org/
- https://data-flair.training/forums/topic/explain-the-process-of-spilling-in-mapreduce/