Lab 2 Report

Data Aggregation, Big Data Analysis and Visualization

Kiran Prabhakar | UBIT: kprabhak | Person No: 50287403

Shishir Suvarna | UBIT:shishirs | Person No: 50290573

Introduction:

The current project focus on creating a big data pipeline to perform word count and word co-occurrence on the data set collected from Twitter, New York Times and Common Crawl.

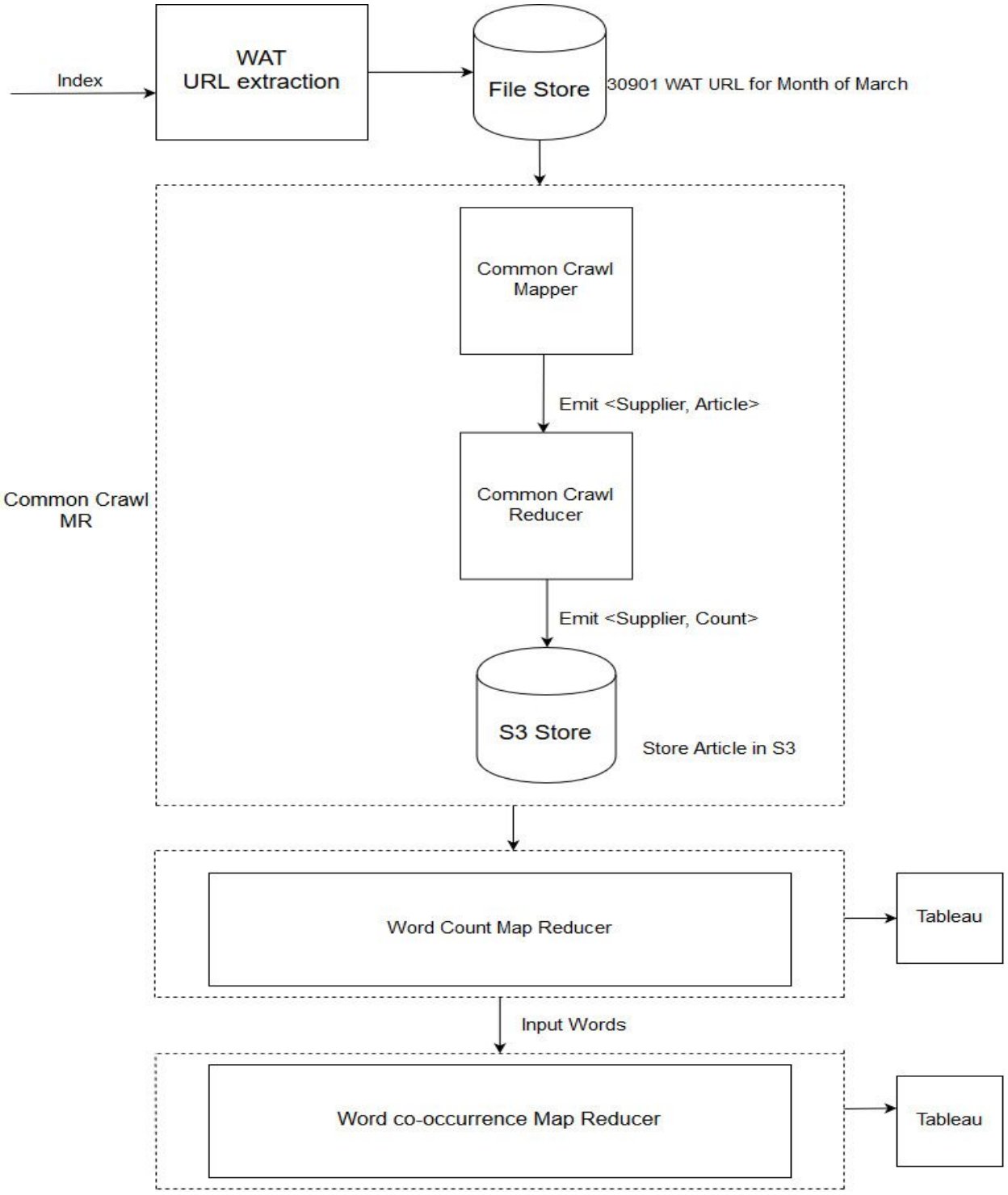
The topic used is Tech Companies and the sub topics are Amazon, Apple, Google, Microsoft and Facebook. Three different big data pipeline are constructed on Amazon's EMR to perform big data analysis. Each of the big data pipeline will be explained in detail in the article.

Common Crawl Pipeline:

The different components in the pipe are classified in to two - Components with in MR and Components outside. The key feature of the pipeline is, **the Common Crawl data is collected with in MR.** As the size of each file is around 1 GB, it's not ideal to download each file and collect the data locally. So, the data is crawled in the EMR pipeline.

The common crawl data is filtered filtered using the **WAT files**. The relevant article url is picked by matching the title, description of the article with the sub topics.

The detailed description of the entire pipeline is described below with a block diagram.


Index
WAT
URL extraction
File Store
30901 WAT URL for Month of March
Common Crawl
MR
Common Crawl
Mapper
Emit <Supplier, Article>
Common Crawl
Reducer
Emit <Supplier, Count>
S3 Store
Store Article in S3
Word Count Map Reducer
Tableau
Input Words
Word co-occurrence Map Reducer
Tableau

Components outside in MR

The component collects the common crawl url of using the index link available in the Common Crawl website. The index url will return the details of the warc file relevant to the the domain. The warc file path is modified to give the relevant WAT file path.

The list of WAT file paths are then given to MR, 100 at a time, to crawl and perform big data analysis. As the WAT files contains only the metadata information, it is highly efficient to process, filter and extract the relevant web urls for the article. Processing WAT file is more than 50% faster than processing WARC file.

Components inside MR:

The most efficient feature of Common Crawl piple is collecting the data with in MR by leveraging the computational power of the EMR systems. The details of the current component are as below.

The input file containing relevant WAT paths are given as input to the mapper. The WAT file paths are split across the mappers to efficiently collect data.

In the data collection process, each article in the WAT file is analysed to identify if the article is relevant to the subtopic by verifying the header and **HTML-Metadata and Metas** attributes in the json. If the subtopic is present in the title or description of the Metas attribute, then the article is considered as relevant for data pipeline.

Once an relevant article is identified the url of the article is extracted from the header tag of the WAT file. The extracted url is then passed to **jsoup get** to scrap the contents in the web page relevant to the article. Only the **p tags** with length greater than 100 is selected.

The urls related to www.amazon.com and www.facebook.com are ignored to avoid articles which are irrelevant to technology. Also, as the subtopics like facebook, amazon and apple can give articles which are not relevant to tech companies, a more aggressive filter which check for the subtopics in title case is applied to get relevant articles.

The relevant article is emitted from the mapper as **<Subtopic, Article>.** So the input of the reducer will be **<Subtopic, List<Article>>.** Here the Subtopic and Article are of Text type.

The collected article is persistently stored in the **S3 bucket**. The reducer connects to the S3 bucket using the **Amazon S3 client, S3 access key, secret** and the bucket name. While storing the articles, the reducer calculates the count of articles collected.

The collected articles are then passed to the word count and word co occurrence MR.

Visualizations:

Word Count

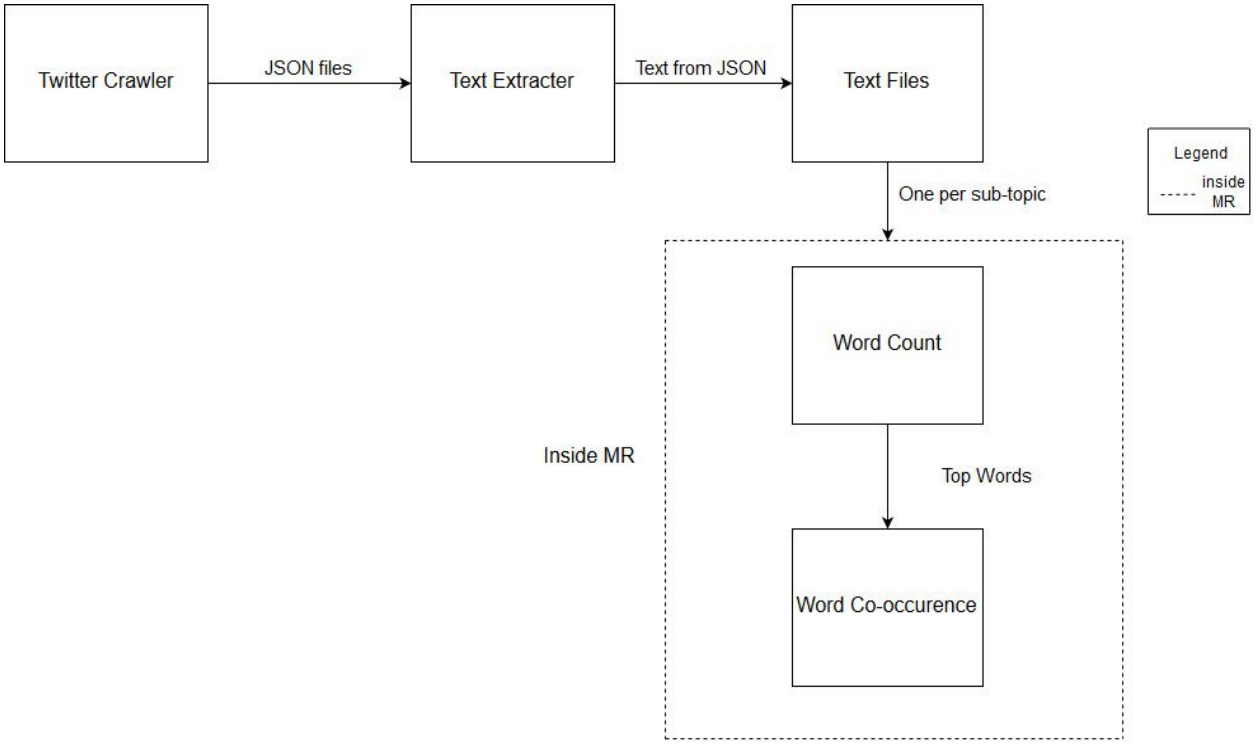
Word Co-occurrence:

Twitter Pipeline:

The articles relevant to subtopics are collected using the twitter crawler component built as part of LAB1. The tweets are collected from the **first week of March**. Around **6000** tweets are used per topic, summing up to **30000** tweets over all across the topics.

The tweets are stored in json format, all json files are then passed to text extractor component to extract the relevant text. Inside the component, the text is filtered out to remove, emoji, urls, mentions, punctuations and tweets in languages other than english. Only alphanumeric tweets are taken into consideration. Also to avoid irrelevant tweets, only tweets containing Title cased subtopics are filtered out.

The filtered tweets are curated in separate files based on the subtopic and uploaded to **tw** folder in s3. This folder location is given to word count and word co-occurrence MR. The high level block diagram for the above process is explained shown below.



Visualizations:

Word Count

Word Co-occurrence:

New York Times Pipeline:

The data fors New York times is collected by using the article search api. The filter query (fq) used are Technology, Organization, Science and Education. The data is collected from 01-01-2019 to 04-05-2019. To meet the project requirements, a total of **528** is collected across all subtopics.

The json response received from api is passed to the Url extractor to filter out the urls of the related articles.

The list of urls are passed to the Beautiful soup python module to extract out the paragraphs in the article. The scraping is done in python to learn and understand the working of web scraping.

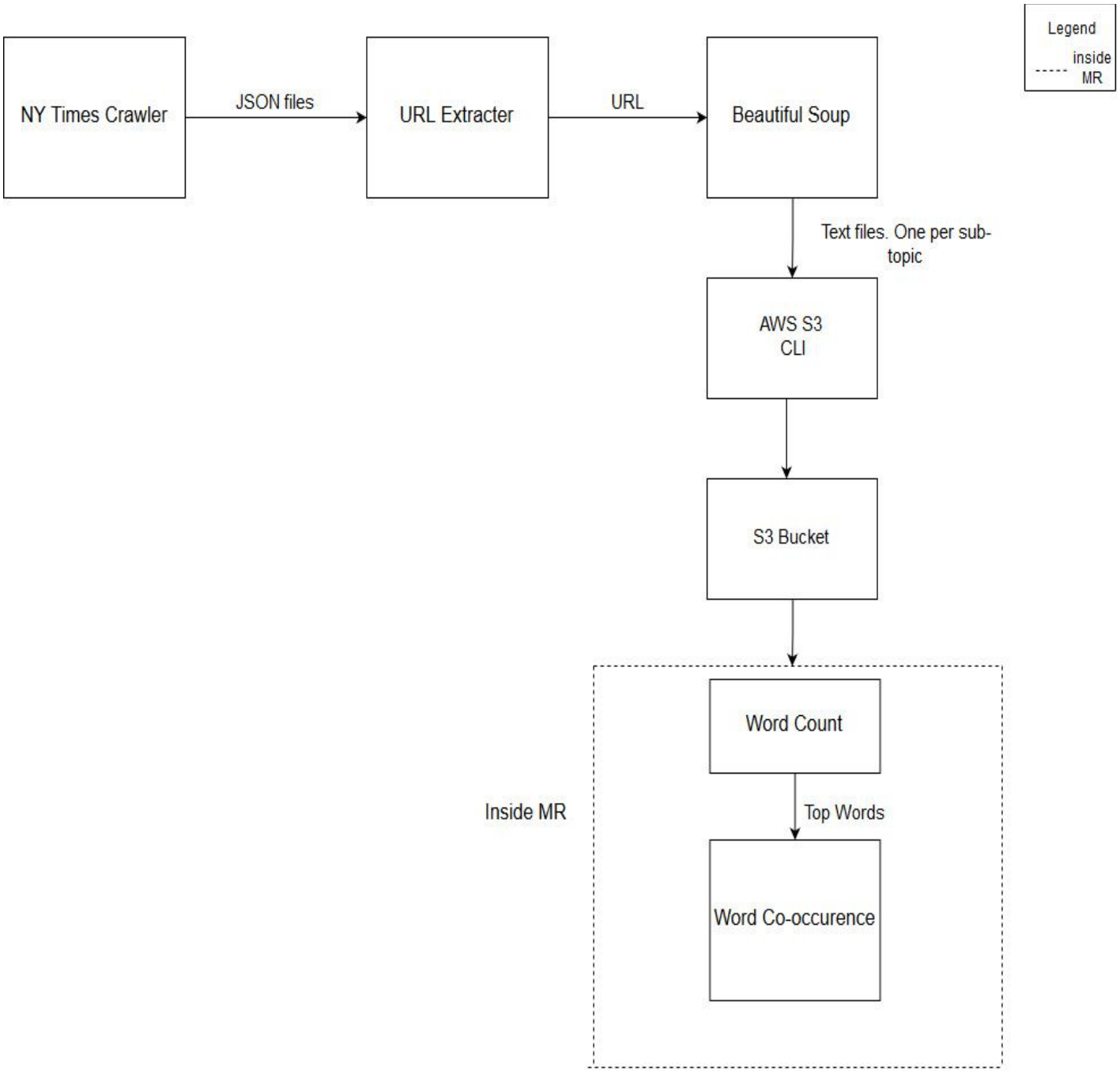
The scrapped articles are stored in text file with one article per file. The purpose of having multiple files is to learn and understand Amazon S3 CLI tool, using which multiple files of huge size can be uploaded to S3, by avoiding the slow upload rate using the web browser.

The setup of S3 CLI requires some configuration of key and secret before using it. The command for this is as below.

aws s3 cp "C:\DIC\Labs\Lab2\BeautifulSoup\rawdata\Apple" s3://diclab2kprabhakshishirs/nyt/rawData --include ".txt" --recursive.*

Once all the multiple files are uploaded to nyt folder in S3, the folder is given as input the Word Count and Co-occurrence MR.

The high level block diagram of above process is shown as below.


Legend
inside MR
NY Times Crawler
JSON files
URL Extracter
URL
Beautiful Soup
Text files. One per sub-topic
AWS S3 CLI
S3 Bucket
Inside MR
Word Count
Top Words
Word Co-occurence

Word Count MR:

The Word count MR is generic for all types of data. It takes the input form Common Crawl, Twitter and NY times folder.

The input to the mapper is the text files location and the output from mapper is **<Word,1>,** the key value pair is emitted which is received by the reducer.

The input to the reducer is **<Word, Iterable<IntWritable>>,** for each item in the value, the reduce function increments the count with the value returned by the iterator.

The reducer emits the aggregated count for each word, hence its output is **<Word, Count>.**

Word Co-Occurrence MR:

Same as word count, the Co-occurrence MR can take data of any data source.

The Key feature of Co-Occurrence MR is to read the top ten words from the word count output and use it for finding the co-occurrence in the article.

A word pair is selected for a top word by considering all the words in the article. Sot the boundary is the entire article for determining the co-occurred word in the article.

The algorithm for the Co-occurrence is as below

```
top_words = readTopWordsFromS3()

Tokens = all tokens in article

For all words in  top_words

Do:

        For all tokens in article

                Do:

                        emit(<<word, token>, 1>)

        }
```

Visualizations:

Word Count

Word Co-occurrence:

Webpages:

- https://public.tableau.com/profile/kiran.prabhakar#!/vizhome/WordCount_15558819001340/Sheet1?publish=yes
- https://public.tableau.com/profile/kiran.prabhakar#!/vizhome/WordCount_subtopic/Sheet1?publish=yes
- https://public.tableau.com/profile/kiran.prabhakar#!/vizhome/CoOccurrance/Sheet1?publish=yes

Collected Data information:

	Apple	Microsoft	Amazon	Google	Facebook	Word Count	Co Occurrence
NYT	286	101	293	299	1518	24949	78620
CC	501	452	180	375	89	51952	42465
Twitter	8509	6630	8343	11103	7356	56203	47639

Conclusion:

We have applied popular Big Data methods to collect and analyze data according to our chosen topic of interest.

Map Reduce operations using popular algorithms such as word count and word co-occurrence have been used to gain meaningful insight into data collected.

The final result has been visualized in the form of a word cloud that has been generated using the visualization tool Tableau.