Data Management, Warehousing, And Analytics (Summer 2023)

Assignment 3

Submitted by: Arihant Dugar (B00917961)

GitLab repo:
https://git.cs.dal.ca/dugar/csci5408_s23_b00917961_arihant_dugar/tree/main/Assignment3

Perform research on NoSQL and data processing – To achieve this task, you need to read and understand the usage of spark framework, MongoDB and then implement a programming framework for big data processing, and store.

Problem 1A:

Created a database and collection to store the news articles inside the cluster5408.

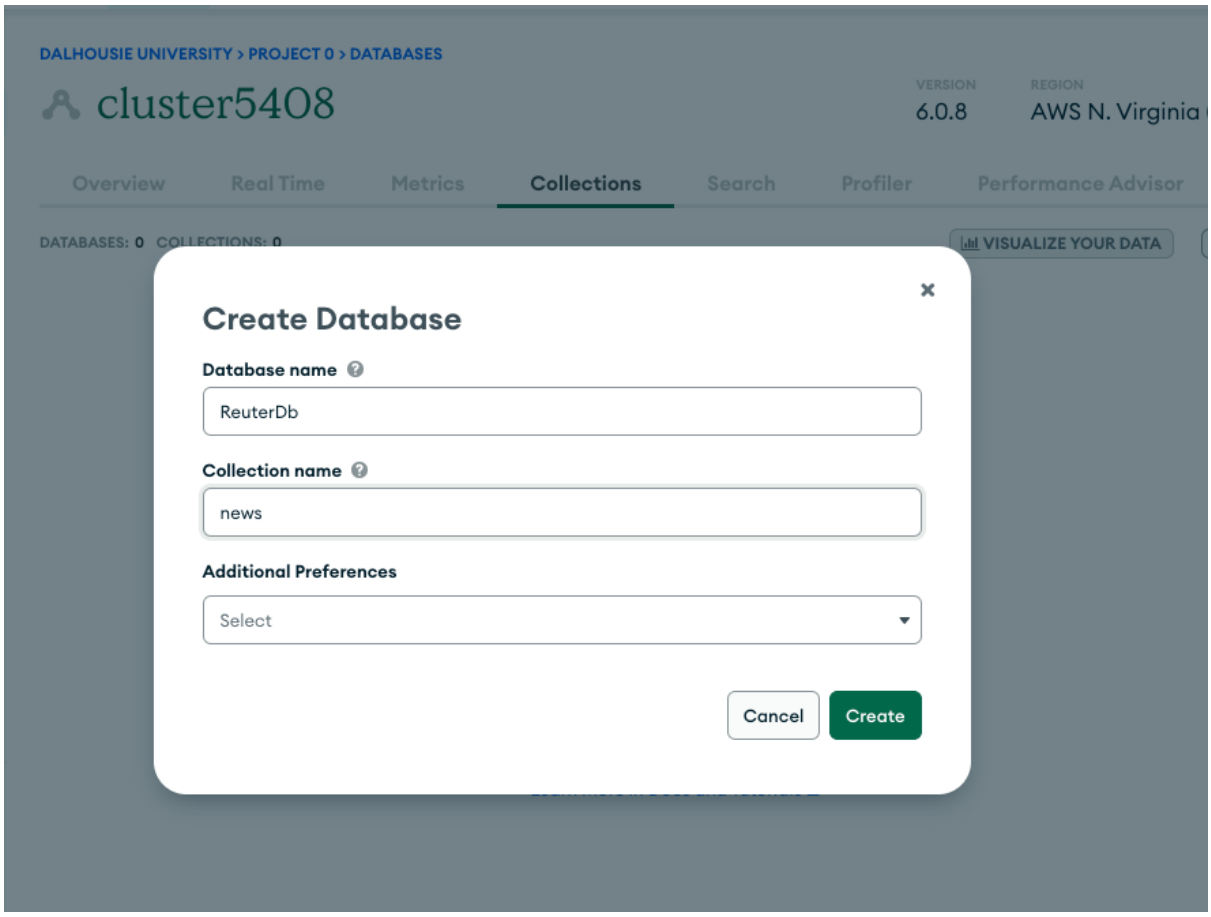

Fig 1.1: Creating database in Mongo DB cluster

After setting up the MongoDB instance on Atlas[2]. Wrote the Java program to parse the data (title, text, dateline and body) and store it in Mongo DB.

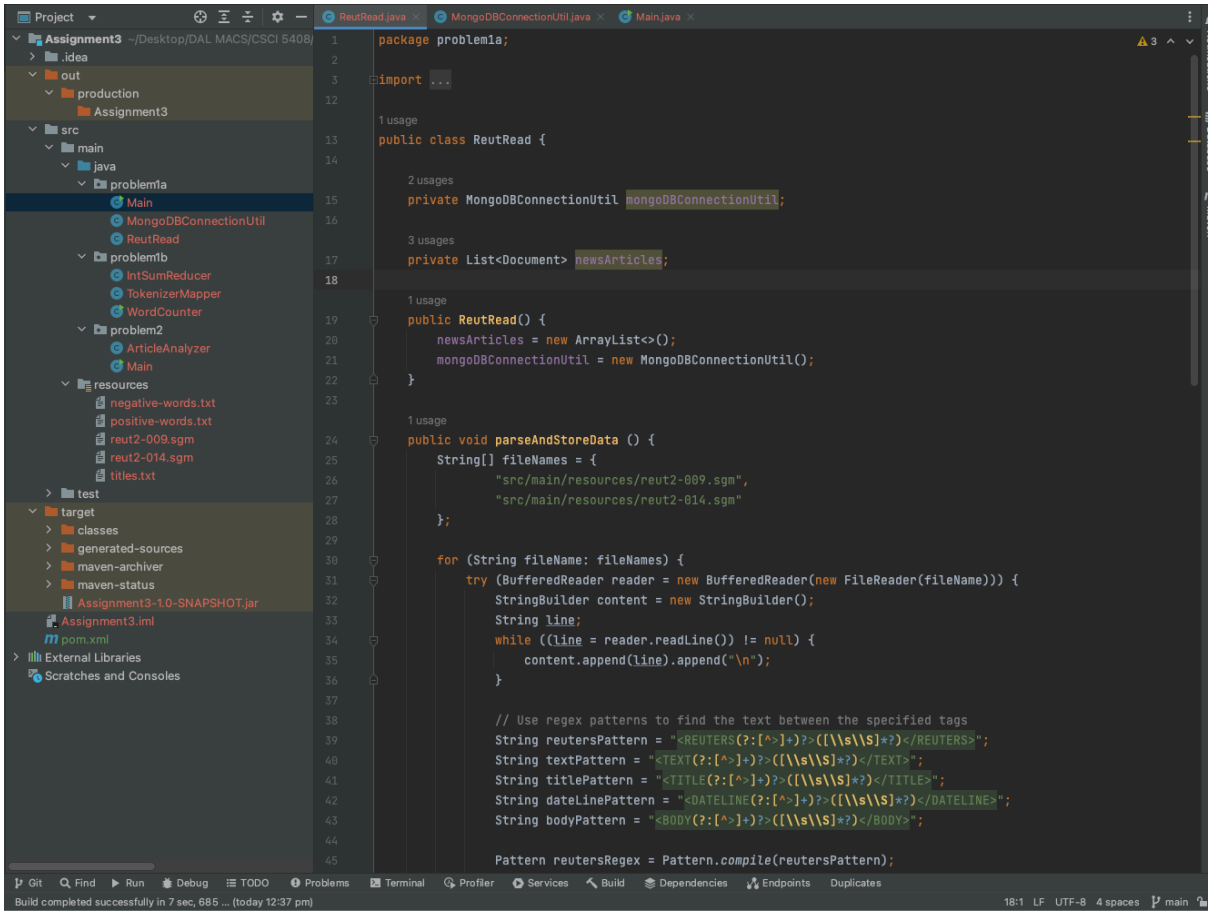

Fig 1.2: ReuterRead program to read news articles and parse data

I have created a separate MongoDBConnectionUtil class that takes care of connecting to database and writing data to the mongo DB collection.

```java
import java.util.List;

public class MongoDBConnectionUtil {

    private static final String CONNECTION_STRING = "mongodb+srv://arihant0996:Cm2dkFHYgpQCXxkS@cluster5408.8owfjo
    private static final String DATABASE_NAME = "ReuterDb";

    private static MongoDatabase mongoDatabase;

    public boolean addNewsArticles(List<Document> document) {
        ServerApi serverApi = ServerApi.builder()
                .version(ServerApiVersion.V1)
                .build();
        MongoClientSettings settings = MongoClientSettings.builder()
                .applyConnectionString(new ConnectionString(CONNECTION_STRING))
                .serverApi(serverApi)
                .build();

        try (MongoClient mongoClient = MongoClients.create(settings)) {
            mongoDatabase = mongoClient.getDatabase(DATABASE_NAME);
            MongoCollection articles = mongoDatabase.getCollection( s: "news");
            articles.insertMany(document);
            return true;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

Fig 1.3: Util class for database connection and operations

The code is configured in such a way that in some cases there are no title, so news without title is not valid. In that case we are omitting those articles and the total count is 1694 after the insertion. If we just insert without that check, then the total articles count is 2000. (It is just a configuration in my program to format and filter data)

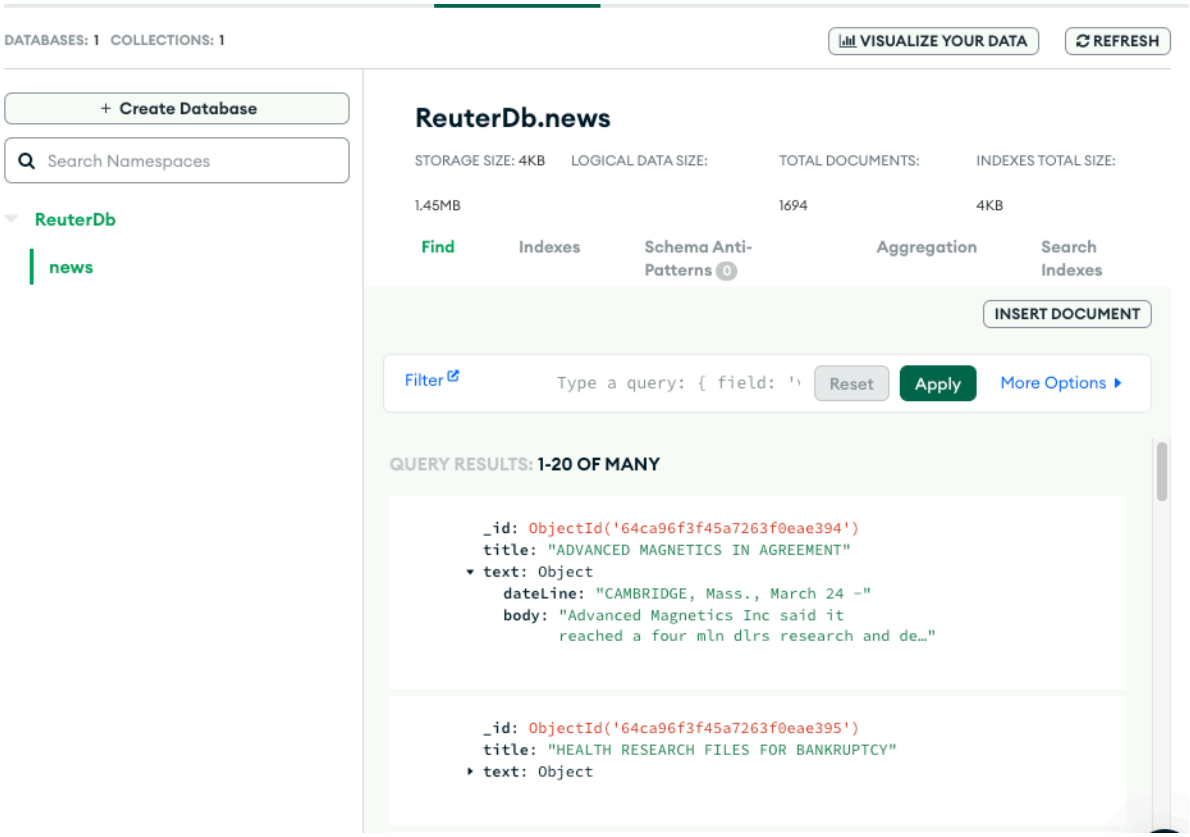

Fig 1.4: Mongo DB news collection data

I have formatted the title, text (dateline & body) data in such a way that it does not contain any tags or any special HTML encodings. The text is a document as it was mentioned that we can have nested documents. It also helps in readability and accessibility of data.

Below is the flowchart using lucid.app[1] for the data clean-up and transformation process:

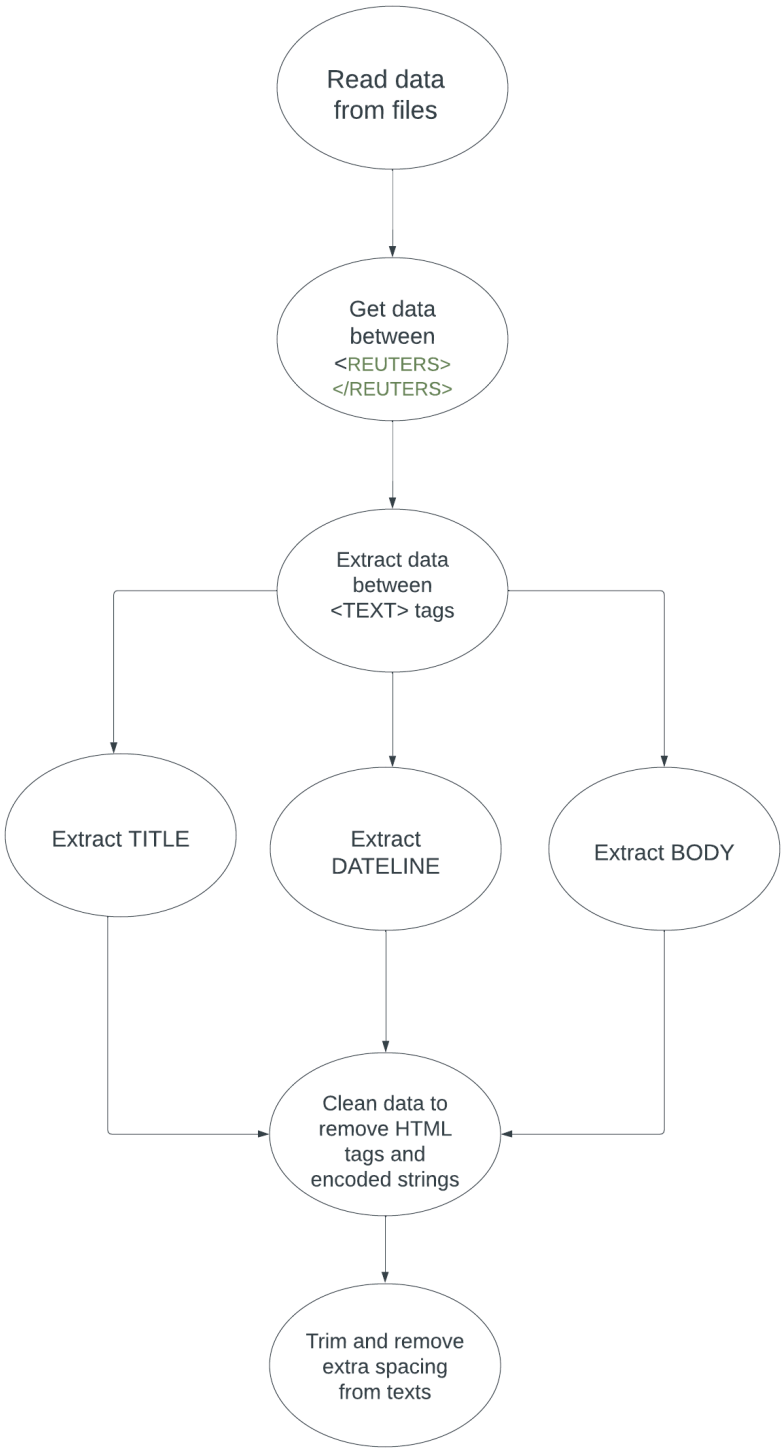

Fig 1.5: Flow chart for data clean up and transformation

Problem 1B:

Created and configured the Apache Spark cluster on GCP cloud. The cluster name is assignment-3.

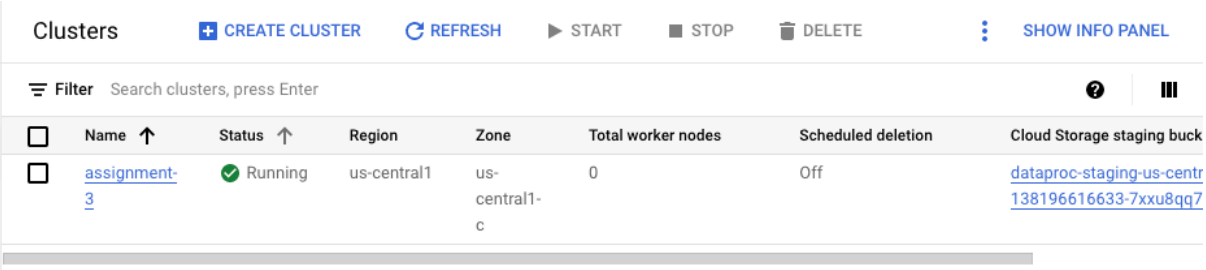

Fig 1.6: Apache spark cluster on GCP

To create an Apache Spark cluster on Google Cloud Platform (GCP) using dataproc, navigate to the dataproc section in the GCP Console and click on "Create Cluster." Provide a name for the cluster and configure the number of worker nodes, machine type, region, and other options based on your needs. Optionally, you can enable features like preemptible nodes for cost savings. Once created, the cluster will be ready to run distributed Spark jobs, enabling you to process large-scale data efficiently in a cloud-based environment.

Written the map reduce code to count the frequency of unique words found in reut2-009.sgm

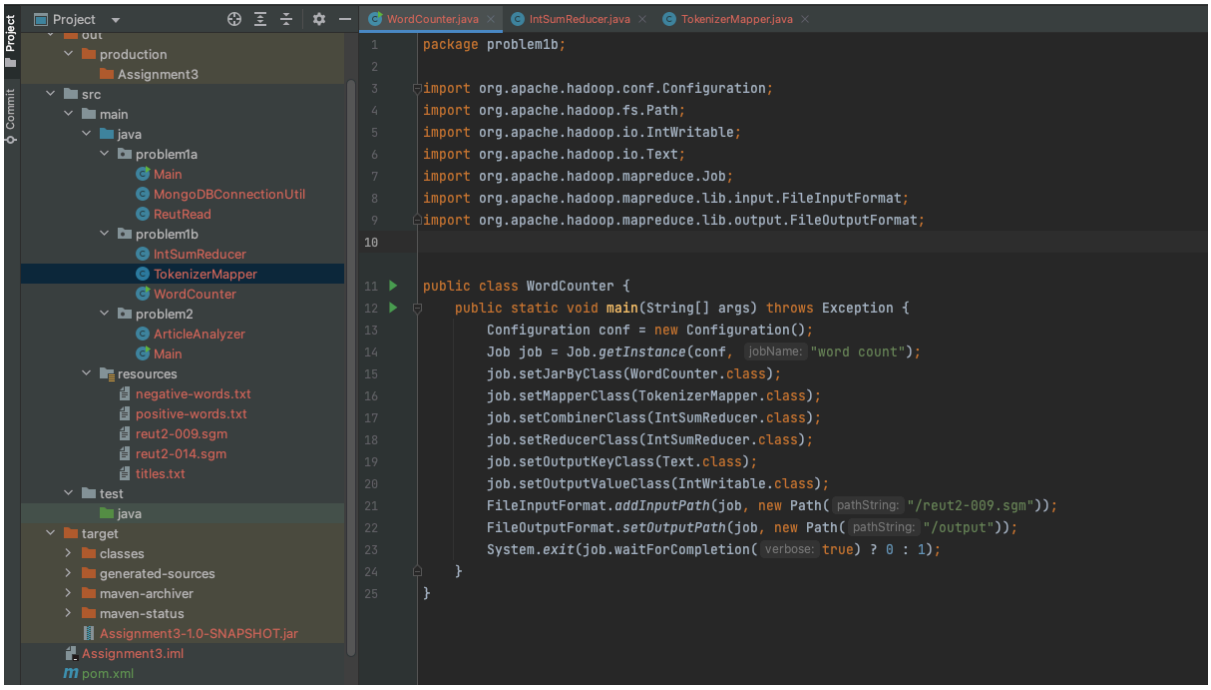

Fig 1.7: Word Counter Java program

When we create a Jar using mvn package command. We SSH into the cluster and upload it along with the source reut2-009.sgm file.

```
arihant0996@assignment-3-m:~$ ls
Assignment3-1.0-SNAPSHOT.jar  reut2-009.sgm
```

Fig 1.8: Uploaded jar and source files on cluster

The below command put copies single src file or multiple src files from cluster file system to the Hadoop Distributed File System.

```
arihant0996@assignment-3-m:~$ hdfs dfs -put reut2-009.sgm /
```

Fig 1.9: copy single src file from cluster file system to the Hadoop Distributed File System

We verify the file is present in the Hadoop file system using the ls command.

```
arihant0996@assignment-3-m:~$ hdfs dfs -ls /
Found 4 items
-rw-r--r--   1 arihant0996 hadoop    1338903 2023-08-01 22:20 /reut2-009.sgm
drwxrwxrwt   - hdfs         hadoop          0 2023-08-01 22:12 /tmp
drwxrwxrwt   - hdfs         hadoop          0 2023-08-01 22:12 /user
drwxrwxrwt   - hdfs         hadoop          0 2023-08-01 22:12 /var
```

Fig 1.10: View contents on Hadoop Distributed File System

Then we run our map reduce program using spark-submit. The screenshot includes the full details of the job along with the % completion for map and reduce.

```
arihant0996@assignment-3-m:~$ spark-submit --master local[*] --deploy-mode client  --class WordCounter Assignme
nt3-1.0-SNAPSHOT.jar hdfs://assignment-3-m/reut2-009.sgm
23/08/01 23:02:12 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at assignment-3-m.us-c
entral1-c.c.expensenest.internal./10.128.0.6:8032
23/08/01 23:02:12 INFO AHSProxy: Connecting to Application History server at assignment-3-m.us-central1-c.c.exp
ensenest.internal./10.128.0.6:10200
23/08/01 23:02:12 WARN JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Too
l interface and execute your application with ToolRunner to remedy this.
23/08/01 23:02:12 INFO JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/arihant
0996/.staging/job_1690927911342_0003
23/08/01 23:02:12 INFO FileInputFormat: Total input files to process : 1
23/08/01 23:02:12 INFO JobSubmitter: number of splits:1
23/08/01 23:02:13 INFO JobSubmitter: Submitting tokens for job: job_1690927911342_0003
23/08/01 23:02:13 INFO JobSubmitter: Executing with tokens: []
23/08/01 23:02:13 INFO Configuration: resource-types.xml not found
23/08/01 23:02:13 INFO ResourceUtils: Unable to find 'resource-types.xml'.
23/08/01 23:02:13 INFO YarnClientImpl: Submitted application application_1690927911342_0003
23/08/01 23:02:13 INFO Job: The url to track the job: http://assignment-3-m.us-central1-c.c.expensenest.interna
l.:8088/proxy/application_1690927911342_0003/
23/08/01 23:02:13 INFO Job: Running job: job_1690927911342_0003
23/08/01 23:02:22 INFO Job: Job job_1690927911342_0003 running in uber mode : false
23/08/01 23:02:22 INFO Job:  map 0% reduce 0%
23/08/01 23:02:29 INFO Job:  map 100% reduce 0%
23/08/01 23:02:40 INFO Job:  map 100% reduce 33%
23/08/01 23:02:41 INFO Job:  map 100% reduce 67%
23/08/01 23:02:42 INFO Job:  map 100% reduce 100%
23/08/01 23:02:43 INFO Job: Job job_1690927911342_0003 completed successfully
23/08/01 23:02:43 INFO Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=458412
                FILE: Number of bytes written=2013831
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1339003
                HDFS: Number of bytes written=355687
```

Fig 1.11: Output of map reduce program using spark-submit

After successful execution of the program, we see the files are generated inside the output folder in the Hadoop file system.

```
arihant0996@assignment-3-m:~$ hdfs dfs -ls /output
Found 4 items
-rw-r--r--   1 arihant0996 hadoop          0 2023-08-01 23:02 /output/_SUCCESS
-rw-r--r--   1 arihant0996 hadoop     118447 2023-08-01 23:02 /output/part-r-00000
-rw-r--r--   1 arihant0996 hadoop     118302 2023-08-01 23:02 /output/part-r-00001
-rw-r--r--   1 arihant0996 hadoop     118938 2023-08-01 23:02 /output/part-r-00002
```

Fig 1.12: Files generated inside the output folder in the Hadoop file system.

When we cat to see the file contents part by part, we see the unique words count that was generated using map reduce command.

```
arihant0996@assignment-3-m:~$ hadoop fs -cat /output/part-r-00000
"AS     1
"America's      2
"An     1
"At     2
"B"     12
"Big    1
"Brazil's       1
"Citibank       2
"Day-to-day     1
"Disaster       1
"Every  1
"Financial      1
"First  1
"For    3
"His    1
"I      47
"If     11
"It's   9
"Marcos 1
"No-one 1
"None   1
"Our    4
"Over   1
"People 1
"Pizza  1
"Quite  1
"Stockholders   1
"That's 1
"The    64
"These  2
"Today,"        1
"We're  6
```

Fig 1.13: The contents of part-r-00000 after running the map reduce program

The word with highest frequency is **"the"** and there are many words that have low frequency such as **"acm"**.

Sentiment Analysis using BOW model on title of Reuters News Articles

Problem 2:

Created a class ArticleAnalyser that takes care of processing articles titles from files. Then the data is stored in a file **titles.txt** inside resources folder. The **negative-words.txt**[5] and **positive-words.txt**[6] is also stored inside resources folder for the data analysis.

The analyseData function has the main logic which calculated the overall score and polarity of the articles.

```java
public void analyseData () {
    String fileName = "src/main/resources/titles.txt";

    try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {
        String line;
        int index = 0;
        while ((line = reader.readLine()) != null) {
            index++;
            String[] words = line.split( regex: " ");
            int tileLen = line.length();
            int largestWordLen = Arrays.stream(words).max(Comparator.comparingInt(String::length)).orElse( oth
            System.out.println(String.format("| %-5s | %-"+ tileLen +"s | %-"+ largestWordLen +"s | %-15s |",
                    "News", "Title Content", "Match", "Polarity"));
            int polarity = 0;
            Map<String, Integer> frequency = getWordsFrequency(words);
            for (Map.Entry<String, Integer> entry : frequency.entrySet()) {
                String word = entry.getKey();
                int count = entry.getValue();
                if(isNegativeWord(word.toLowerCase())) {
                    polarity += (-1*count);
                    System.out.println(String.format("| %-5s | %-"+ tileLen +"s | %-"+ largestWordLen +"s | %
                            index, line, word, "Negative"));
                } else if (isPositiveWord(word.toLowerCase())) {
                    polarity += count;
                    System.out.println(String.format("| %-5s | %-"+ tileLen +"s | %-"+ largestWordLen +"s | %
                            index, line, word, "Positive"));
                } else {
                    System.out.println(String.format("| %-5s | %-"+ tileLen +"s | %-"+ largestWordLen +"s | %
                            index, line, word, "Neutral"));
                }
            }
            System.out.println("Overall Score for the article: " + ((polarity > 0) ? "+" + polarity : polarit
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Fig 2.1: ArticleAnalyzer class that performs the main analysis tasks

The articles are tagged as “Positive”, “Negative” or “Neutral” depending on the overall score. Few examples of each below :

The article 295 has an overall score of +2 as savings has a frequency of 2 and its overall polarity is Positive.

```
| News  | Title Content                          | Match     | Polarity        |
| 295   | CROSSLAND SAVINGS ACQUIRES WESTERN SAVINGS | CROSSLAND | Neutral         |
| 295   | CROSSLAND SAVINGS ACQUIRES WESTERN SAVINGS | SAVINGS   | Positive        |
| 295   | CROSSLAND SAVINGS ACQUIRES WESTERN SAVINGS | WESTERN   | Neutral         |
| 295   | CROSSLAND SAVINGS ACQUIRES WESTERN SAVINGS | ACQUIRES  | Neutral         |
Overall Score for the article: +2
Title polarity: Positive
```

Fig 2.2: Details of Article 295 with positive polarity

The article 309 has an overall score of -2 and its overall polarity is Negative.

```
| News  | Title Content                                | Match    | Polarity        |
| 309   | SOUTH AFRICA PROBLEMS REMAIN DESPITE DEBT PACT | AFRICA   | Neutral         |
| 309   | SOUTH AFRICA PROBLEMS REMAIN DESPITE DEBT PACT | REMAIN   | Neutral         |
| 309   | SOUTH AFRICA PROBLEMS REMAIN DESPITE DEBT PACT | DESPITE  | Neutral         |
| 309   | SOUTH AFRICA PROBLEMS REMAIN DESPITE DEBT PACT | PROBLEMS | Negative        |
| 309   | SOUTH AFRICA PROBLEMS REMAIN DESPITE DEBT PACT | PACT     | Neutral         |
| 309   | SOUTH AFRICA PROBLEMS REMAIN DESPITE DEBT PACT | SOUTH    | Neutral         |
| 309   | SOUTH AFRICA PROBLEMS REMAIN DESPITE DEBT PACT | DEBT     | Negative        |
Overall Score for the article: -2
Title polarity: Negative
```

Fig 2.3: Details of Article 309 with negative polarity

The article 423 has an overall score of 0 and its overall polarity is Neutral.

```
| News  | Title Content                    | Match    | Polarity        |
| 423   | GROLIER DEBT UPGRADED BY MOODY'S | MOODY'S  | Neutral         |
| 423   | GROLIER DEBT UPGRADED BY MOODY'S | BY       | Neutral         |
| 423   | GROLIER DEBT UPGRADED BY MOODY'S | UPGRADED | Positive        |
| 423   | GROLIER DEBT UPGRADED BY MOODY'S | GROLIER  | Neutral         |
| 423   | GROLIER DEBT UPGRADED BY MOODY'S | DEBT     | Negative        |
Overall Score for the article: 0
Title polarity: Neutral
```

Fig 2.4: Details of Article 423 with Neutral polarity

References:

[1] "Lucidchart," Lucid.app, 2015. https://lucid.app/lucidchart (accessed Aug. 02, 2023).

[2] "MongoDB Atlas," MongoDB, 2023. https://www.mongodb.com/cloud/atlas/register (accessed Aug. 02, 2023).

[3] Code With Arjun, "MapReduce Word Count Example using Hadoop and Java," YouTube. Oct. 11, 2022. Accessed: Aug. 02, 2023. [YouTube Video]. Available: https://www.youtube.com/watch?v=qgBu8Go1SyM

[4] "Maven Repository: org.apache.hadoop» hadoop-core," Mvnrepository.com, 2023. https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-core (accessed Aug. 02, 2023).

[5] "negative-words.txt," Gist, Dec. 14, 2012. https://gist.github.com/mkulakowski2/4289441 (accessed Aug. 02, 2023).

[6] "positive-words.txt," Gist, Dec. 14, 2012. https://gist.github.com/mkulakowski2/4289437 (accessed Aug. 02, 2023).