CSCI 5408
DATA MANAGEMENT AND WAREHOUSING
ASSIGNMENT - 2

Banner ID: B00981016
GitLab Link: Assignment - 2

Table of Contents

Problem 1A:

Flowchart

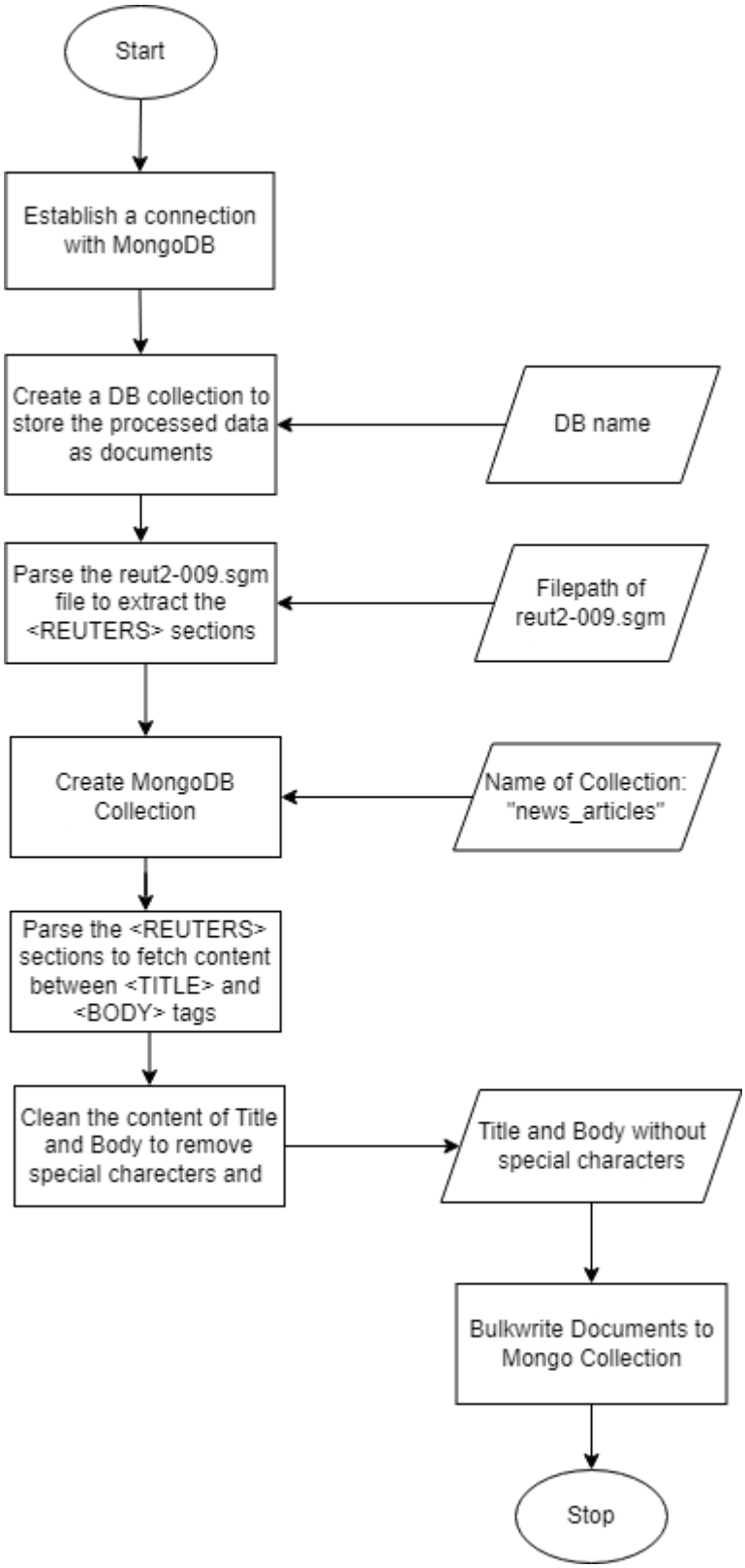

Figure 1: Flowchart for problem 1A (ReuterReader.java)

Algorithm

Step 1: Start

Step 2: Establish a connection to the MongoDB database.

Step 3: Create the ReuterDb database from the MongoDB connection.

Step 6: Extract Reuters text from a the “reut2-009.sgm” file.

Step 7: Create a collection named "news_articles" in the ReuterDb database.

Step 8: Retrieve the "news_articles" collection from the database.

Step 9: Extract titles and bodies from the list of Reuters text sections.

Step 10: For each Reuters section:

a. Extract the title using a regex pattern.

b. Extract the body using a regex pattern.

c. Clean the extracted title and body by removing special characters and HTML charecter entities.

d. Create a Document object containing the title and body.

e. Add the Document object to a list of documents to be inserted into the MongoDB collection.

Step 11: Perform a bulk write operation to insert the documents into the collection.

Step 12: Stop

Evidence of testing for Problem 1A

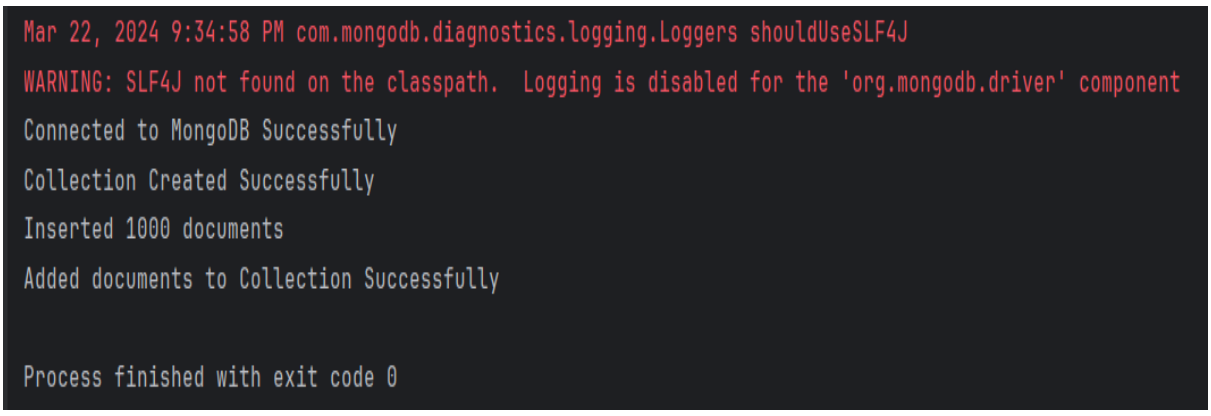

Figure 2: Program run successfully showing that all documents were added to MongoDB collection

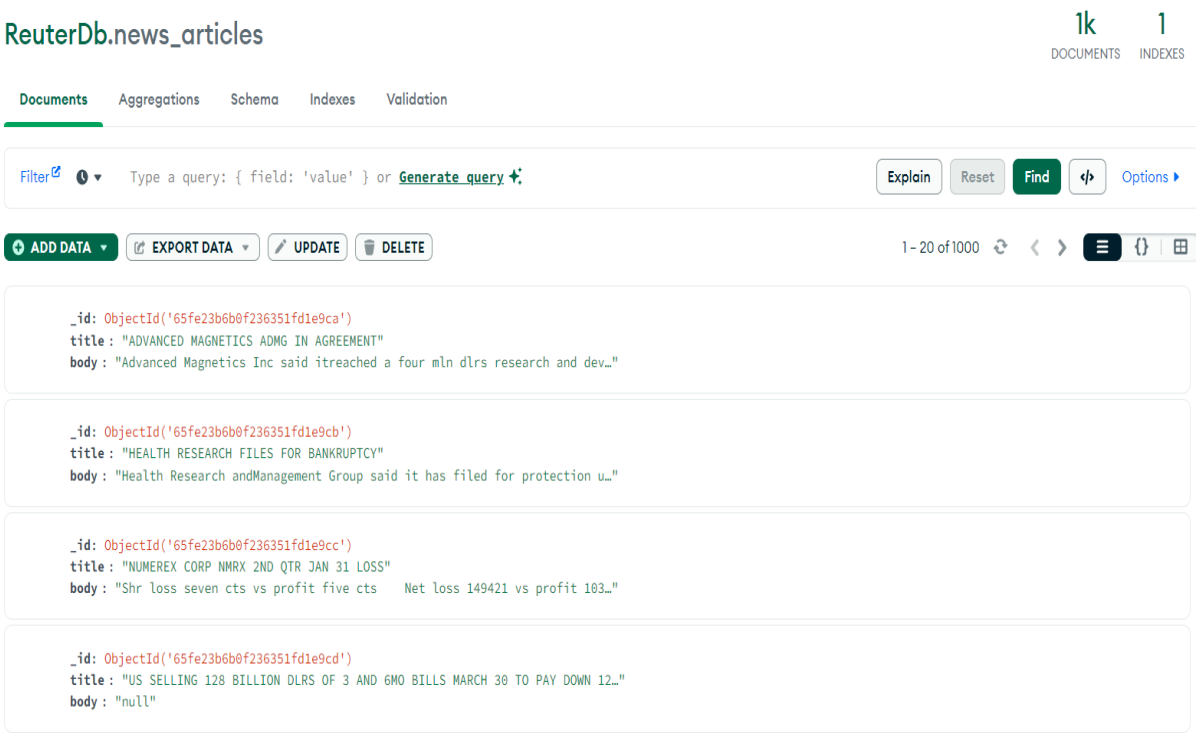

Figure 3: Created documents stored in MongoDB

Problem 1B

Setting up Apache Spark Cluster on GCP

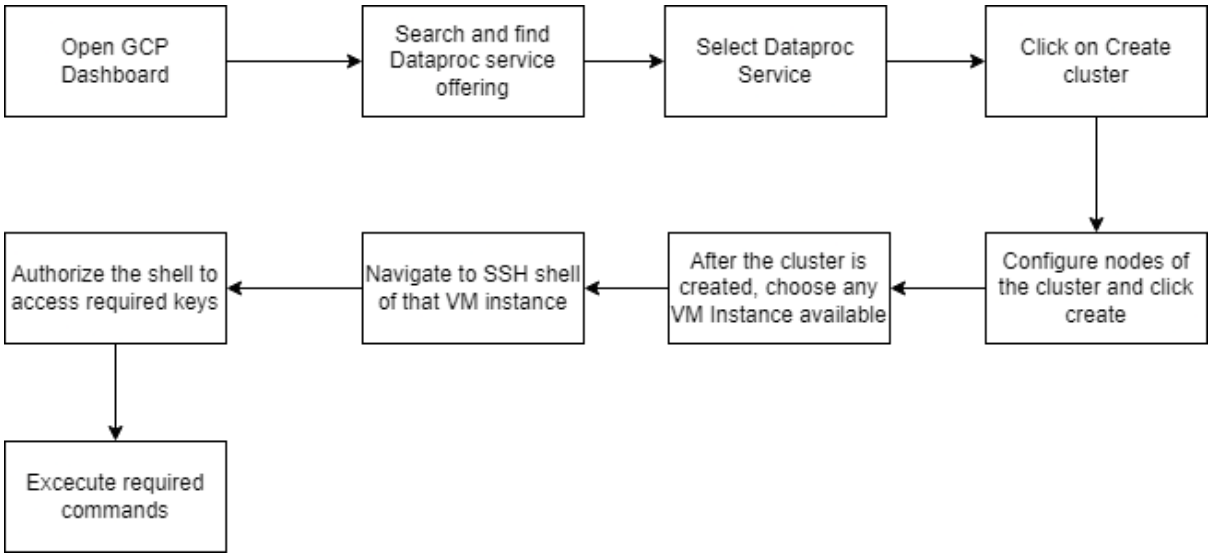

Figure 4: Steps of setting up Apache Spark Cluster on GCP

Apache Spark Frequency Count of Unique Words

Step 1: Start

Step 2: Create a SparkSession with the specified configurations.

Step 3: Specify the file path of the Reuters data file.

Step 4: Read the text file into a JavaRDD.

Step 5: Collect the lines of text into a single string, separated by spaces.

Step 6: Instantiate a DataCleaner object.

Step 7: Clean the content by removing XML tags and Html character entities.

Step 8: Remove single characters from the cleaned content.

Step 9: Remove stop words from the content.

Step 10: Break the cleaned content into words.

Step 11: Create a word frequency map to store each word along with its frequency.

Step 12: Go through the words:

a. Update the word frequency map with the frequency of each word.

Step 13: Identify words with the minimum and maximum frequencies:

a. Initialize variables to track the minimum and maximum frequencies.

b. Initialize lists to store words with minimum and maximum frequencies.

c. Iterate through the word frequency map:

i. If the frequency of the word is less than the current minimum frequency:

- Update the minimum frequency.

- Clear the list of words with minimum frequency and add the current word.

ii. If the frequency of the word is equal to the current minimum frequency:

- Add the word to the list of words with minimum frequency.

iii. If the frequency of the word is greater than the current maximum frequency:

- Update the maximum frequency.

- Clear the list of words with maximum frequency and add the current word.

iv. If the frequency of the word is equal to the current maximum frequency:

- Add the word to the list of words with maximum frequency.

Step 14: Print the first 20 words with minimum frequency.

Step 15: Print the words with maximum frequency.

Step 16: Print the number of words with minimum frequency.

Step 17: Print the number of words with maximum frequency.

Step 18: Stop the SparkSession.

Step 19: Stop

Evidence of Testing for Problem 1B

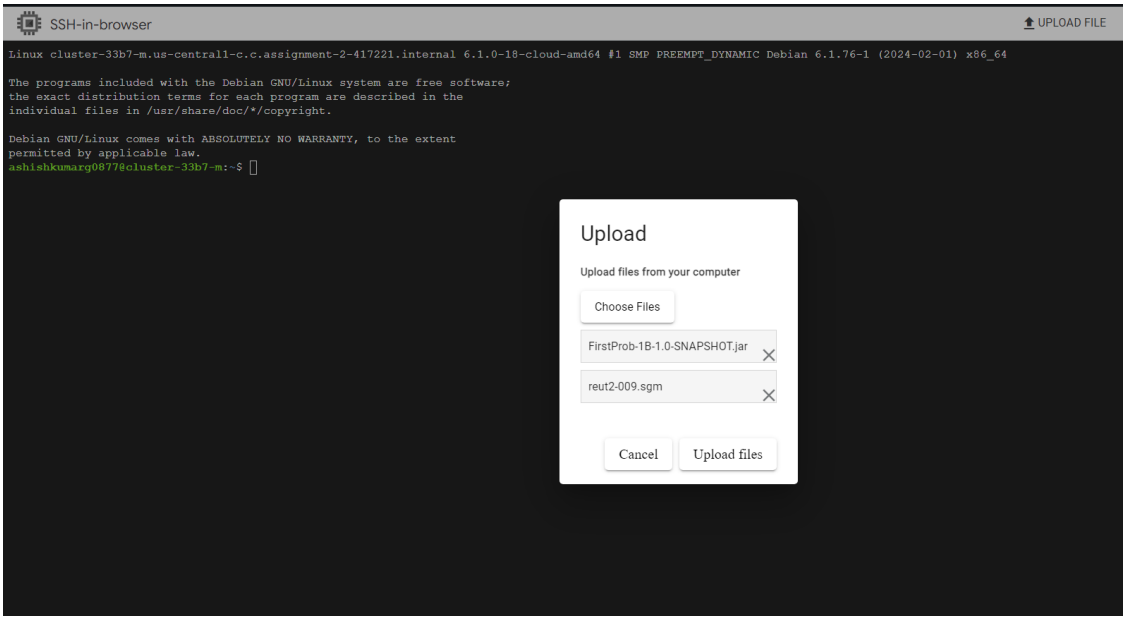

Figure 5: Upload the jar file and the file containing the articles to VM instance

To run the code in the cluster we need to create a .jar file from the code. Run the maven package command to generate the required jar file. Then open up the VM instance on GCP cluster previously created and upload the .jar and the reut2-009.sgm files. Then run the below command on the command line to get the output:

spark-submit --class org.example.Main FirstProb-1B-1.0-SNAPSHOT.jar

After the above command is run, we see the outputs such as words with minimum frequency, maximum frequency, and their respective counts.

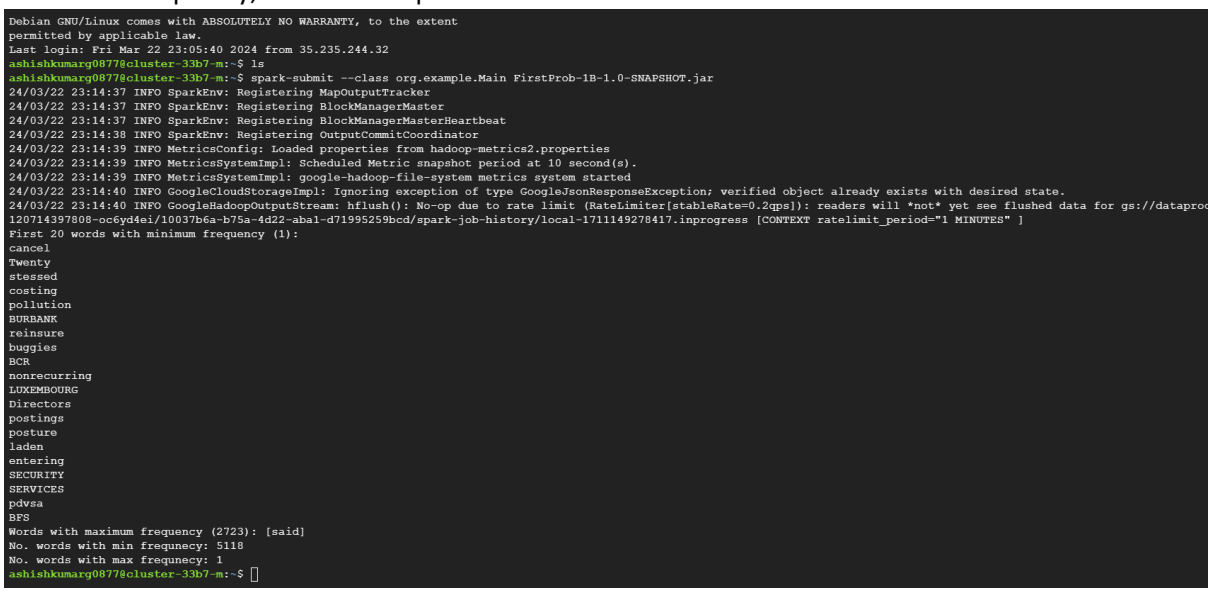

Figure 6: Output of the WordCount program run on the Reuters articles file

The word with maximum frequency is: ‘said’

The frequency count of the word ‘said’ is 2723.

There are 5118 words in the articles file with the same minimum frequency i.e., 1. The above image shows the first 20 of them. The first 20 words with minimum frequency are: cancel, Twenty, stessed, costing, pollution, BURBANK, reinsure, buggies, BCR, nonrecurring, LUXEMBOURG, Directors, postings, posture, laden, entering, SECURITY, SERVICES, pdvsa, BFS.

Problem 2:

Entities chosen from Assignment-1 are: City_University, Faculty/Degree, Program, Courses.

Neo4j is a graph database that stores data in form of nodes and edges, where nodes are the datapoints and the edges are the relationships between the datapoints. Neo4j uses Cypher query language [1] to perform CRUD operations. The syntax of it is a bit different from the SQL but some parallels can be drawn upon careful observation.

Creation of University Node:

To create the city_university node in neo4j below command is used

```
CREATE (:University {name: 'city_university'});
```

- University is the label assigned to the node.
- {name: 'city_university'} is a map representing the properties of the node. In this case, it specifies the name property of the node as "city_university".

Creation Of Faculty/Degree Node:

To create the Degree node in neo4j below command is used

```
CREATE (:Degree {name: 'Faculty of Computer Science'});
```

Creation Of Program Node:

To create the Program node in neo4j below command is used

```
CREATE (:Program {name: 'Applied Computer Science'});
```

Creation Of Course Nodes:

To create the different course nodes in neo4j below command is used

```
CREATE (:Course {name: 'Communication', courseCode: '5100'}),
       (:Course {name: 'Software development', courseCode: '5308'}),
       (:Course {name: 'Databases', courseCode: '5408'});
```

Creation Of Relations Between Nodes

Below are the queries used to create all the relations between the nodes

```
MATCH (u:city_university {name: 'city_university'})
MATCH (d:degree {name: 'Faculty of Computer Science'})
CREATE (u)-[:OFFERS]->(d);
```

```
MATCH (d:Degree {name: 'Faculty of Computer Science'})
MATCH (p:Program {name: 'Applied Computer Science'})
CREATE (d)-[:HAS]->(p);
```

```
MATCH (p:Program {name: 'Computer Science'})
MATCH (c:Course)
WHERE c.name IN ['Communication', 'Software development', 'Databases']
CREATE (p)-[:PROVIDES]->(c);
```

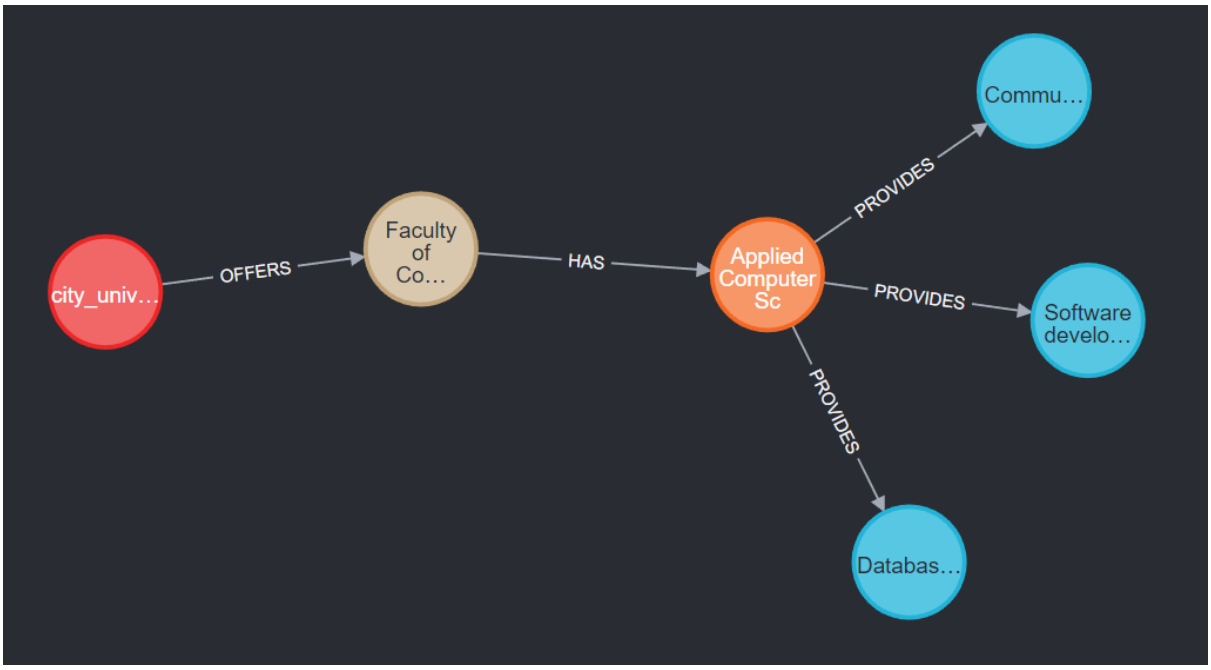

Figure 7: Graph created from the chosen entities from Assignment-1

Problem 3

Flow of Excecution

Below is the execution flow of the program

1. Create a MongoDB client connection to the specified MongoDB Atlas cluster.
2. Retrieve the "ReuterDb" database from the MongoDB connection.
3. Retrieve the "news_articles" collection from the "ReuterDb" database.
4. Instantiate a BOWSentiment object.
5. Read negative and positive words from their respective files.
6. Create an empty list to store titles.
7. Retrieve titles from documents in the MongoDB collection and add them to the created list.
8. Compute the bag of words for the titles
9. Count number of matches in bag of words with list of positive and negative word lists
10. Calculate sentiment polarity and score
11. Write sentiment analysis results to a CSV file.
12. Read and display data from the CSV file on the output console

Evidence of Testing for Problem 3

```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
Mar 22, 2024 9:58:08 PM com.mongodb.diagnostics.logging.Loggers shouldUseSLF4J
WARNING: SLF4J not found on the classpath.  Logging is disabled for the 'org.mongodb.driver' component
Connected Successfully
Collection Retrieved Successfully
Data written successfully to sentiments.csv
+-------+-----------------------------------------------------------------------------------------+-------------------------+-------+----------+
| News# | Title Content                                                                           | Matched Words           | Score | Polarity |
+-------+-----------------------------------------------------------------------------------------+-------------------------+-------+----------+
| 1     | ADVANCED MAGNETICS ADMG IN AGREEMENT                                                    | advanced                | 1     | Positive |
| 2     | HEALTH RESEARCH FILES FOR BANKRUPTCY                                                    |                         | 0     | Neutral  |
| 3     | NUMEREX CORP NMRX 2ND QTR JAN 31 LOSS                                                   | loss                    | -1    | Negative |
| 4     | US SELLING 128 BILLION DLRS OF 3 AND 6MO BILLS MARCH 30 TO PAY DOWN 12 BILLION DLRS     |                         | 0     | Neutral  |
| 5     | US 2YEAR NOTE AVERAGE YIELD 643 PCT STOP 644 PCT AWARDED AT HIGH YIELD 85 PCT           | awarded                 | 1     | Positive |
| 6     | COMMODORE CBU ATARI IN SETTLEMENT                                                       |                         | 0     | Neutral  |
| 7     | BALDRIGE SUPPORTS NIC TALKS ON CURRENCIES                                               | supports                | 1     | Positive |
| 8     | TRIANGLE TRI BEGINS EXCHANGE OFFER                                                      |                         | 0     | Neutral  |
| 9     | SOUTHMARK SM UNIT IN PUBLIC OFFERING OF STOCK                                           |                         | 0     | Neutral  |
| 10    | EASTMAN KODAK CO TO SELL HOLDINGS IN ICN PHARMACEUTICALS AND VIRATEK INC                |                         | 0     | Neutral  |
| 11    | FEUD PERSISTS AT US HOUSE BUDGET COMMITTTEE                                             |                         | 0     | Neutral  |
| 12    | TREASURY BALANCES AT FED ROSE ON MARCH 23                                               |                         | 0     | Neutral  |
| 13    | FARM CREDIT SYSTEM SEEN NEEDING 800 MLN DLRS AID                                        |                         | 0     | Neutral  |
```

Figure 8: Program successfully calculated sentiments for all the titles and displayed on console

	A	B	C	D	E
1	News#	Title Content	Matched Words	Score	Polarity
2	1	ADVANCED MAGNETICS ADMG IN AGREEMENT	advanced	1	Positive
3	2	HEALTH RESEARCH FILES FOR BANKRUPTCY		0	Neutral
4	3	NUMEREX CORP NMRX 2ND QTR JAN 31 LOSS	loss	-1	Negative
5	4	US SELLING 128 BILLION DLRS OF 3 AND 6MO BILLS MARCH 30 TO PAY DOWN 12 BILLION DLRS		0	Neutral
6	5	US 2YEAR NOTE AVERAGE YIELD 643 PCT STOP 644 PCT AWARDED A	awarded	1	Positive
7	6	COMMODORE CBU ATARI IN SETTLEMENT		0	Neutral
8	7	BALDRIGE SUPPORTS NIC TALKS ON CURRENCIES	supports	1	Positive
9	8	TRIANGLE TRI BEGINS EXCHANGE OFFER		0	Neutral
10	9	SOUTHMARK SM UNIT IN PUBLIC OFFERING OF STOCK		0	Neutral
11	10	EASTMAN KODAK CO TO SELL HOLDINGS IN ICN PHARMACEUTICALS AND VIRATEK INC		0	Neutral
12	11	FEUD PERSISTS AT US HOUSE BUDGET COMMITTTEE		0	Neutral
13	12	TREASURY BALANCES AT FED ROSE ON MARCH 23		0	Neutral
14	13	FARM CREDIT SYSTEM SEEN NEEDING 800 MLN DLRS AID		0	Neutral
15	14	USX X USS UNIT RAISES PRICES		0	Neutral
16	15	UNIONIST URGES RETALIATION AGAINST JAPAN		0	Neutral
17	16	EXXON XON GETS 992 MLN DLR CONTRACT		0	Neutral
18	17	EATON ETN GETS 530 MLN DLR CONTRACT		0	Neutral
19	18	ZAIRE AUTHORIZED TO BUY PL 480 RICE USDA		0	Neutral
20	19	MCDONNELL DOUGLAS GETS 306 MLN DLR CONTRACT		0	Neutral
21	20	MIDIVEST ACQUIRES ASSETS OF BUSINESS AVIATION		0	Neutral
22	21	US WHEAT CREDITS FOR JORDAN SWITCHED		0	Neutral
23	22	DOLLAR EXPECTED TO FALL DESPITE INTERVENTION	fall	-1	Negative
24	23	US TO SELL 128 BILLION DLRS IN BILLS		0	Neutral
25	24	INLAND STEEL IAD TO BUILD NEW PLANT IN INDIANA		0	Neutral
26	25	EASTMAN KODAK EK TO SELL HOLDINGS		0	Neutral
27	26	GUINNESS SUES BOESKY IN FEDERAL COURT	sues	-1	Negative

Figure 9: Sentiments.csv with columns TitleContent, Matched Words, Score and Polarity

There were:

- 151 Negative sentiment titles
- 133 Positive sentiment titles
- 716 Neutral sentiment titles

References

[1] “Introduction”, *Neo4j* [Online]. Available: https://neo4j.com/docs/cypher-manual/current/introduction/