

Report

Assignment No. – 4

Problem -2

To perform this task, I considered the processed News description (ignored other fields) that I obtained from NewsAPI and stored in MongoDB in Assignment No. – 3.

Step - 1)

In order to perform this task, I first fetched the News Article containing just the News “Description”. Refer Figure 1 to see the method to connect to MongoDB and to fetch just the description from the News Article dataset.

```
// fetch description from News Article uploaded on MongoDB
1 usage
2 public ArrayList<String> fetchNewsDataDescriptionFromMongoDB() {
3     ArrayList<String> descriptionList = new ArrayList<>();
4     ConnectionString connectionString = new ConnectionString("mongodb+srv://Faiza:jgxu96UQixANGk6V@cluster0.oelyvip.mongodb.net/?retryWrites=true&w=ma
5     MongoClientSettings settings = MongoClientSettings.builder()
6         .applyConnectionString(connectionString)
7         .serverApi(ServerApi.builder()
8             .version(ServerApiVersion.V1)
9             .build())
10        .build();
11     MongoClient mongoClient = MongoClient.create(settings);
12     MongoDBDatabase mongoDatabase = mongoClient.getDatabase("myMongoNews");
13     MongoCollection<Document> mongoCollection = mongoDatabase.getCollection("newsDataCollection");
14     MongoCursor<Document> cursor = mongoCollection.find().iterator();
15     while (cursor.hasNext()) {
16         String str = cursor.next().get("description").toString();
17         descriptionList.add(str);
18     }
19     return descriptionList;
20 }
```

Figure 1: Represents the method to fetch description from MongoDB

Step – 2)

After getting the description from MongoDB, I need to create bag-of-words for each News description. So that each bag-of-words can be compared to positive and negative words. I read the files of positive and negative word as shown in Figure 2, 3 and 4 so that I can compare the bag of words from these text files.

```
2 usages
1 public static List<String> readFile(String fileName) {
2     List<String> readLines = Collections.emptyList();
3     try {
4         readLines = Files.readAllLines(Paths.get(fileName), StandardCharsets.UTF_8);
5     } catch (IOException e) {
6         e.printStackTrace();
7     }
8     return readLines;
9 }
```

Figure 2: Representing the method to readFile()

```

1 usage
public List<String> readNegativeWordsFile() {
    List<String> list = SentimentAnalysis.readFile(fileName: "negative-words.txt");
    // System.out.println(list);
    return list;
}

```

Figure 3: Read negative.txt file using method readNegativeWordsFile()

```

1 usage
public List<String> readPositiveWordsFile() {
    List<String> list = SentimentAnalysis.readFile(fileName: "positive-words.txt");
    // System.out.println(list);
    return list;
}

```

Figure 4: Read positive.txt file using the method readPositiveWordsFile()

Step -3)

- I performed word by word comparison by creating a method **wordComaprisonOfDescription()**.
- In which I called list of two files i.e. **readPositiveWordsFile()** and **readNegativeWordsFile()**.
- Then I created array list of string to get the description and clean it to remove extra special characters or numbers using regex. I did filtering because I just need bag of words and not the characters for comaparison.
- I stored the clean description in array list of string.
- I used Iterator to fetch each cleaned description from the article.
- If the **bagOfWords** contains the word increment by 1, else keep count as 1 for that word.
- After matching the words present in the description, I need to calculate the score to get the polarity.
- I initialized the score with 0 and mapped each bagOfWords.
- If score < 0 → polarity is “negative”, if score > 0 → polarity is “positive” else polarity → “neutral”.
- After this calculation, I return the **totalListOfDescriptionWithPolarity** using **HashMap<Integer, List<String>>**.
- Refer figure 4, 5 and 6, to see the code for the method wordComaprisonOfDescription().

```

usage
public HashMap<Integer, List<String>> wordComparisonOfDescription() {
    List<String> positiveWords = readPositiveWordsFile();
    List<String> negativeWords = readNegativeWordsFile();

    HashMap<Integer, List<String>> totalListOfDescriptionWithPolarity = new HashMap<>();
    ArrayList<String> description = fetchNewsDataDescriptionFromMongoDB();
    ArrayList<String> cleanedDescription = new ArrayList<>();
    String removeSpecialCharacters = "[^a-zA-Z0-9~+~]";
    String removeNumbers = "[013456789]";

    for (String list_element : description) {
        list_element = list_element.replaceAll(removeSpecialCharacters, replacement: " ");
        list_element = list_element.replaceAll(removeNumbers, replacement: " ");
        cleanedDescription.add(list_element);
    }
    Iterator<String> iterator = cleanedDescription.iterator();
    int articleNumber = 1;
    while (iterator.hasNext()) {
        HashMap<String, Integer> bagOfWords = new HashMap<>();
        String eachDescription = iterator.next();
        String tempDescription[] = eachDescription.split(regex: " ");

        for (String word : tempDescription) {
            if (bagOfWords.containsKey(word)) {
                bagOfWords.put(word, bagOfWords.get(word) + 1);
            }
        }
    }
}

```

Figure 4: Represents method wordComparisonOfDescription()

```

        if (bagOfWords.containsKey(word)) {
            bagOfWords.put(word, bagOfWords.get(word) + 1);
        } else {
            bagOfWords.put(word, 1);
        }
    }

    int score = 0;
    String matchWord = " ";

    for (Map.Entry<String, Integer> entry : bagOfWords.entrySet()) {
        if (negativeWords.contains(entry.getKey())) {
            matchWord = matchWord + "," + entry.getKey();
            score = score - entry.getValue();
        } else if (positiveWords.contains(entry.getKey())) {
            matchWord = matchWord + "," + entry.getKey();
            score = score + entry.getValue();
        }
        else{
            score += 0;
        }
    }

    // check polarity for the bag of words
    String polarity = null;
    if (score < 0) {
        polarity = "negative";
    } else if (score > 0) {
        polarity = "positive";
    }
}

```

Figure 5: Represents method wordComparisonOfDescription() (continuation)

```

    }
    // check polarity for the bag of words
    String polarity = null;
    if (score < 0) {
        polarity = "negative";
    } else if (score > 0) {
        polarity = "positive";
    } else {
        polarity = "neutral";
    }
    List<String> values = new ArrayList<>();
    values.add(eachDescription);
    values.add(matchWord);
    values.add(polarity);
    totalListOfDescriptionWithPolarity.put(articleNumber, values);
    articleNumber++;
}
return totalListOfDescriptionWithPolarity;
}

```

Figure 6: Represents method `wordComparisonOfDescription()` (continuation)

Step – 4)

- I created a method to **displayTable()**.
- I used **AsciiTable** to render the table.
- The table consists of the columns “News Article”, “Description”, “Match”, “Polarity”.
- I called the method **wordComaprisonOfDescription()** inside **displayTable()** to render the data in the table. The data is nothing but the News Article Number, Description, matching words and polarity that we calculated in word `wordComaprisonOfDescription()`.
- Because the description was too long, the table was not able to render that amount of data. So I used substring to limit the display and used ellipses.
- I also used file writer to write the output in different text file as “sentiment_table.txt”. Refer figure 8.
- Refer figure 7, to see the code for method `displayTable()`.

```

1 usage
public void displayTable() {
    AsciiTable asciiTable = new AsciiTable();
    asciiTable.addRule();
    asciiTable.addRow(...columns: "News Article", "Description", "Match", "Polarity");
    asciiTable.addRule();
    HashMap<Integer, List<String>> result = wordComparisonOfDescription();
    for (Map.Entry<Integer, List<String>> mapEntry : result.entrySet())
    {
        Integer newsArticleNumber = mapEntry.getKey();
        List<String> valueList = mapEntry.getValue();
        String displayDescription = valueList.get(0);
        if (displayDescription.length() > 50) {
            displayDescription = displayDescription.substring(0, 50);
            displayDescription = displayDescription + "...";
        }
        String displayWord = valueList.get(1);
        String displayPolarity = valueList.get(2);
        asciiTable.addRow(newsArticleNumber, displayDescription, displayWord, displayPolarity);
        asciiTable.addRule();
    }
}

```

Figure 7: Represents method displayTable()

```

String renderTable = asciiTable.render();

//write table into file
try {
    FileWriter fileWriter = new FileWriter(fileName: "sentiment_table.txt");
    fileWriter.write(renderTable);
} catch (IOException e) {
    throw new RuntimeException(e);
}
System.out.println(renderTable);
}

```

Figure 8: Represents method displayTable() (continuation)

Output:

As there were 827 articles, I took screenshot of the start, middle and end part of the table. To see full table please check the sentiment_table.txt file.

```
"C:\Program Files\Java\jdk-11.0.15\bin\java.exe" ...
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
```

News Article	Description	Match	Polarity
1	GM is training more first responders to be able to...		neutral
2	The first in-person Canada Day celebrations in thr...	,freedom	positive
3	The move by the UK US Canada and Japan will str...	,strike	negative
4	Canada said on Tuesday it was imposing new sanctio...	,imposing	negative

364	Many have told Halifax they are closing their acco...	,boycott	negative
365	House price growth slows as rising costs bite		neutral
366	Sharp rises in property prices is making satelllite...	,pricey,desirable	neutral
367	Halifax said that although wages have risen by jus...		neutral
368	The pace of Canadian home price growth slowed in M...	,slowed	negative

447	New technologies and widespread internet and smart...	,led,smart	positive
448	Whether you re considering a big move or want to d...		neutral
449	As legal single-event sports betting expands across...	,susceptible,proble	negative
450	ICICI Bank Campus Power The new Campus Power pl...	,comprehensive	positive
451	PRNewswire -- Measure Learning the premier prov...	,premier	positive

823	The Federal Reserve s commitment to reining in -...	,commitment,uncondi tional,risk	positive
824	Ireland s cabinet is set to agree to significantly...	,available	positive
825	Saudi Arabia s non-oil private sector expanded in ...	,fastest	positive
826	Middle Eastern stock markets ended lower on Tuesda...		neutral
827	The White House expects the next round of economic...		neutral

Process finished with exit code 0

References:

[1] “Extract multiple fields using MongoDB 3.2.0 java driver,” *Stack Overflow*. [Online]. Available: <https://stackoverflow.com/questions/34695546/extract-multiple-fields-using-mongodb-3-2-0-java-driver>. [Accessed: 25-Jul-2022].

[2] [1] S. van der Meer, README.adoc at master vdmeer/asciitable. [Online]. <https://github.com/vdmeer/asciitable/blob/master/README.adoc>. [Accessed: 25-Jul-2022].

[3] “How to read a text file into ArrayList in Java? Examples,” *Java67.com*. [Online]. Available: <https://www.java67.com/2016/07/how-to-read-text-file-into-arraylist-in-java.html>. [Accessed: 26-Jul-2022].

Git Link:

https://git.cs.dal.ca/umatiya/csci5408_s22_a4_faiza_umatiya_b00899642