Assignment 2
Posted Date: Nov 6, 2023
Submission Due: Nov 28, 2023 (11:59 pm)
Late assignments will not be accepted and will result in a 0 on the assignment

---

Objective: This assignment covers two learning objectives (lo).

- ***lo#1:*** 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.
- ***lo#2:*** Build a light-weight analytics engine, which will perform custom ETL operation, and one specific analysis (sentiment and semantic)

Plagiarism Policy:

- This assignment is an individual task. Collaboration of any type amounts to a violation of the academic integrity policy and will be reported to the AIO.
- Content cannot be copied verbatim from any source(s). Please understand the concept and write in your own words. In addition, cite the actual source. Failing to do so will be considered as plagiarism and/or cheating.
- Usage of AI tools are not allowed for this assignment (refer to the AI policy given in the syllabus).
- The Dalhousie Academic Integrity policy applies to all material submitted as part of this course. Please understand the policy, which is available at: https://www.dal.ca/dept/university_secretariat/academic-integrity.html

Problem 1A: Reuter News Data Reading & Transformation and storing in **MogoDb**. Objective is lo#1

1. From the two given news files (reut2-009.sgm, and reut2-014.sgm), create MongoDb Database – ReuterDb, where each Document contains a news article. The task must be done using a Java Program "ReutRead.java".
   a. To perform this operation, you need to write a Java code to scan the required texts between <TITLE></ TITLE > tags, and <BODY></BODY> tags which are inside each <REUTER></REUTER tag.
   b. In the ReuterDb, you may consider each news as a document. You can also include nested or sub-document. {
      "title": "ADVANCED MAGNETICS &lt;ADMG> IN AGREEMENT",
      "body: "Advanced Magnetics Inc said it
      reached a four mln dlrs research and development agreement with….."
      }
2. You need to include a flowchart and algorithm of your Reuters Data cleaning/transformation program on the PDF file.

Problem 1B: Reuter News Data Processing using **Spark**. Objective is lo#1

1. Using your GCP cloud account, configure and initialize Apache Spark cluster. (Follow the tutorials provided in Lab session).
2. Create a flowchart or write ½ page explanation on how you completed the task, include this part in your PDF file.
3. Using Apache Spark count (frequency count) the unique words found in "reut2-009.sgm".
4. You need to include a flowchart/algorithm of your Spark framework frequency count operation on the PDF file.

5. In your PDF file, also mention the words that have highest and lowest frequencies.
6. You can ignore stop words by removing them using stop words library before submitting the job to spark.

Grading Scheme: – Problem 1 (60% of 10 points)

- Flowchart of 1A: 5%
- Flowchart of 1B: 5%
- Completion of Tasks of 1A (including citation): 10%
- Completion of Tasks of 1B (including citation): 10%
- Code Quality and Formatting: 10%
- Functional Testing
    - Test Cases and Evidence of Testing using screenshots: 20% (1A: 10%, 1B: 10%)

Problem 2: Sentiment Analysis using BOW model on title of Reuters News Articles

Use **Core Java Program only** with no additional libraries. Use the parser (regex based parser)

1. Write a Java program to create bag-of-words for each News title. (code from online or other sources are not accepted)

e.g. news1 = " ADVANCED MAGNETICS ADMG IN AGREEMENT "
bow1 = {“advanced”:1, “magnetics”:1, “admg”:1, “in”:1, “agreement”:1}

You do not need any libraries. Just implement a simple counter using loop.

2. Compare each bag-of-words with a list of positive and negative words.

You can download list of positive and negative words from online source(s). You do not need any libraries. Just perform word by word comparison with a list of positive and negative words that you can get from any online platform. E.g. negative words can be found here https://gist.github.com/mkulakowski2/4289441

3. Tag each news title as “positive”, “negative”, or “neutral” based on overall score. You can add an additional column to present your finding.

E.g. frequencies of the matches “advanced” =+1, “agreement” = +1, Overall score = +2 (positive)
**You need to insert the titles and perform matching with score detection automatically using your program. The table structure should look like this. Hint: You can write this to a file or a database table, and then export to it in a tabular format and submit with your code.

News#	Title Content	match	score	Polarity
1	ADVANCED MAGNETICS ADMG IN AGREEMENT	Advanced, agreement	+2	Positive

Grading Scheme: – Problem 1 (40% of 10 points)

- Completion of task, clarity, and correct submission: 10%
- Code Quality and Formatting: 10%
- Functional Testing
    - Test Cases and Evidence of Testing using screenshots: 20%

Submission Guidelines:

1. All written reports, images, code etc. must be added in a folder, and compress it with **.ZIP** format only.
2. If not mentioned by TAs, then please rename the .zip file with your B00xxxxx_FnameLname_A2
3. Submit your Java code in gitlab. Your TA must have provided guidelines for that. If not, please ask the TA.
4. You must include Test Cases (at least 3 – manual testing of functionality or validation testing) for the developed application and provide necessary screenshots as evidence of testing. Note: This is not Junit test, this is functional test of each problem.
5. Check the next point “Suggestions” for quality improvement and time management.

Suggestions:

Better Quality: To obtain good grades, you should follow the points given below:

- Try to understand the assignment requirement and follow all the steps required.
- Do not miss adding citations. If you write a single sentence taking the idea from somewhere else, then give credit to the author. Therefore, provide citation for any report you write, or any code you implement
- When you add citation, make sure to add it in a standard format and uniform format. E.g. if I refer 3 sources for writing a report, then I must cite the 3 sources in same format. One source in MLA, two sources in APA citation format will be a mismatch. Therefore, follow any one standard citation format
- Make sure to provide inline citations within report, and programming code
- Any image/picture/flowchart/diagram you add, make sure to provide a caption and a number for that image. It should be placed at the bottom of the image. E.g. "**Fig 1: Weekly time management chart for CSCI 5408**"
- Any table you add, must have a number and caption. This should be added on top of the table. E.g. "**Tab1: Table highlights the requirements in a ordered format**"

Time Management: Follow proper time management to reduce stress, and last-minute preparations. I am suggesting you follow the pie chart, which will require you to spend 5 hours in a week outside the classroom time for this course.

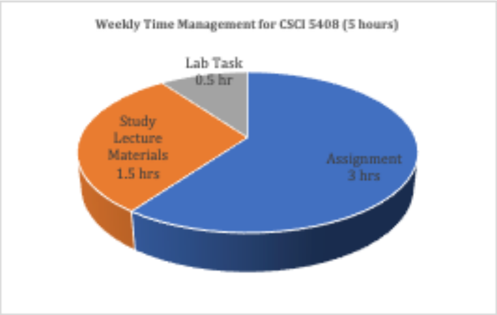

Fig 1: Weekly time management chart for CSCI 5408