ENSF 612
Assignment 1
Marks: 25
This assignment is worth 5% of your course grades
Due 11:59 PM October 10, 2023

1) **Producing a random sub-sample of a big dataset using Hadoop MapReduce (Marks: 3)**

   Text analytics projects often start by sampling a small fraction of data from a bigger dataset to conduct some preliminary analysis, e.g., estimating the expected skew of words in the dataset. For this problem, you will develop a job that **randomly** selects *approximately* **10%** of the lines in a large input dataset and writes that result to the distributed file system for further analysis.

2) **Building n-grams using Hadoop MapReduce. (Marks: 5)**

   An n-gram is a contiguous sequence of n items in a sequence of text or document. Computing n-grams has many applications including natural language processing, modeling languages, and speech recognition. For example, calculating the counts for various n-grams can help with text prediction.

   For this problem, you will use Python and Hadoop streaming to build counts for a word-level 2- gram or a di-gram. Your program should be able to scan a set of text documents and then record all unique 2 word sequences in each line of the document. As a final output, it should list all the unique di- grams along with their counts. Make sure you consider additional requirements for implementing this solution (e.g., removing punctuation) as part of your analysis.

3) **Building an inverted index of a text corpus using Hadoop MapReduce. (Marks: 7)**

   An inverted index is often used in information retrieval and search applications. An inverted index maps content, e.g., words, to locations where the content is found, e.g., file names of documents. Such an index allows search engines to quickly locate content relevant to search queries.

   For this problem, you will build an inverted index for contents in an input directory. Your program should scan all the documents in the input directory. For each file, it should map every unique word in thefile to the name of that file. The final output of your program will consist of several records. The key of each record is a unique word occurring in the documents contained in the input directory. The value of each record is a list of file names of documents that contain that word. Make sure you consider additional requirements for implementing this solution (e.g., removing punctuation) as part of your analysis. Your program should be scalable to large datasets and hence should use MapReduce.

4) **Sorting using Hadoop MapReduce. (Marks: 10)**

For this problem, you are going to sort words found in a large text document. Specifically, your program should scan the document and then output the words in the document in ascending order. Make sure you consider additional preprocessing of the documents as part of your analysis.

Your program should be scalable to large datasets and hence should use MapReduce and **multiple reducers**. Single reducer solutions will only get partial credit since they won't be scalable. Part of the challenge involved is to deduce a partitioning algorithm and the correct number of reducers that will result in a total order sort, i.e., results appearing sorted across all reducers. The output of each reducer should only contain contiguous words – for example words starting with "s" followed by words starting with "t" – or words starting with a particular alphabet, e.g., words starting with "h".

Hint: You might need to read up the ***KeyFieldPartitioner*** option and examples given in the Hadoop streaming tutorial. You will need to define a multi-part key to explore solutions.

Deliverables:

- **The input, intermediate, and output key-value pairs for each question**
- **The pseudo code for each question's mapper and reducer**
- **The Hadoop command to run each MapReduce job (you will have to look up how to send a MapReduce command)**
- **Brief description justifying why your solution would work and why you've chosen a particular command option**

Note: Submit all of the above information in a single PDF file. You do not need to implement your solution on Hadoop and run python files. However, it is important that you study how to submit a Hadoop MapReduce job and the commands needed to do so.