CSE 482: Big Data Analysis (Spring 2022) Homework 5

Due date: Monday, April 10, 2022

Total: 40 points

1. **[5 points]** For this question, you need to write the corresponding Linux and HDFS commands to perform the tasks listed below. For some of the tasks listed below, you may need to refer to the documentation available at `https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html`. Create a script file (similar to Exercise 8) that will record your keystroke activities on the Hadoop virtual machine. The script file should contain all the commands you had use to perform the steps listed below. The script can be launched by typing the following on the terminal of the Hadoop virtual machine:

   ```
   hadoop@cse482~VirtualBox:~$ script q1_yourMSUNetID.txt
   ```

   The tasks you need to perform are as follows. Make sure you perform the steps sequentially and that each step is recorded in the script file. If you miss a step, you'll need to redo the script again from scratch (after removing the old script).

   (a) Use wget to download the file hw5.tar from `http://www.cse.msu.edu/~cse482/hw5.tar`.

   (b) Unarchive the `hw5.tar` file. This should create a new directory named `hw5`.

   (c) Change to the newly created directory `hw5` and launch the Hadoop server (i.e., start the namenodes, datanodes, resource manager, and node managers).

   (d) Use `hadoop fs` command to upload the directory named `question1` under the `hw5` folder to HDFS.

   (e) Use `hadoop fs` command to list all the files under the directory named `question1` on HDFS.

   (f) Use `hadoop fs` command to display the last 1kB of the file `temp.txt` under the `question1` directory on HDFS.

   (g) Use `hadoop fs` command to remove the file `temp2.txt` under the directory named `question1` on HDFS.

   (h) Use `hadoop fs` command to rename the directory `question1` on HDFS to `data`. **Hint:** Renaming a file/directory is equivalent to "moving" the file from the original file/directory name to a new name.

   (i) Use `hadoop fs` command to merge all the files located under the `data` directory on HDFS into a single file named `result.txt` to be stored on the (local) Linux filesystem on the terminal.

   (j) Stop the Hadoop server (i.e., terminate the namenode, datanode, resource manager, and node manager processes) and exit the script.

You need to submit the script file named `q1_yourMSUNetID.txt` as well as the `result.txt` file.

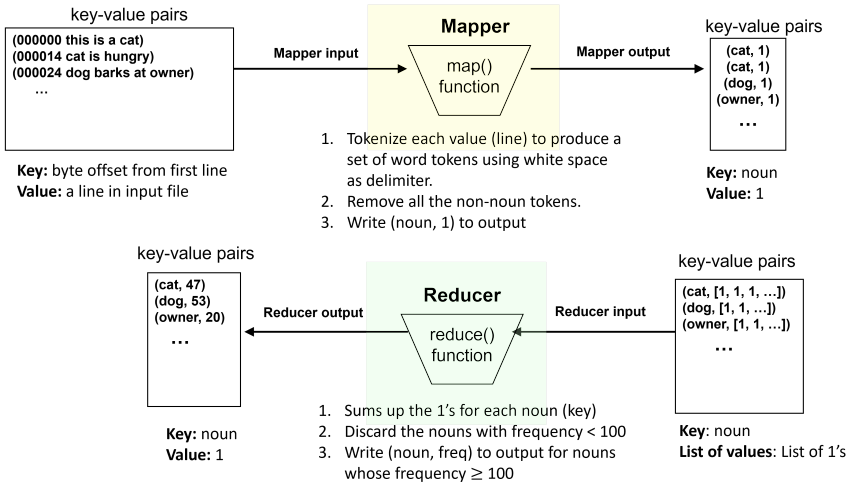
2. **[12 points]** For each problem and data set described below, state how would you setup the (key,value) pairs as inputs and outputs for the mappers and reducers. Also, explain clearly the operations that must be performed by the map() and reduce() functions given their input key-value pairs. Assume the Hadoop program will process the input data one line at a time. Draw the corresponding input-output key-value pairs of the mappers and reducers associated with the problem similar to the example shown below. You must use the ppt file provided to draw the diagram. You can add additional examples to the input data in the diagram if needed. For the reducer output, feel free to make up the value for each key (as long as it looks reasonable given the data provided). State whether a combiner can be used for the given problem.

   Example Question:

   Data set: Collections of text documents.

   Problem: Count frequency of nouns that appear at least 100 times in the documents.

   Example Answer:



   Can use a combiner? No due to the filtering requirement that the frequency must be ≥ 100.

   (a) **Data set:** Employee database. Each line in the input file consists the following information: Name, Department, Occupation, Start Year, and Salary, where Start Year is the year in which the employee joined the organization. For example:

```
Mary Doe,HR,Manager,2005,200000
John Doe,Product,Engineer,2015,145000
```

Problem: For each department, compute the average salary of its employees who joined before 2010.

(b) **Data set:** Online music streaming data. Each line in the data file has 4 columns (userID, artistID, songID, timestamp), where timestamp is the date and time for which the user plays the song performed by the given artist.

Problem: For each user, list his/her favorite song by each artist the user has streamed. For example, among the songs performed by artist #2, if user #1 plays the song #5 most often compared to other songs by the artist, the reducer's output must indicate that the favorite song by artist #2 for user #1 is song #5. **Note:** you need to figure out how the output should be represented as key-value pairs.

(c) **Data set:** Twitter follower graph. Each line in the input file contains a 2-tuple (follower,followee). For example, the record

john123,mary456

means john123 is a follower of the tweets posted by mary456.

Problem: Find all pairs of users who have reciprocal relation, i.e., are mutual followers of each other. For example, if john123 and mary456 has a reciprocal relation, then john123 is a follower of mary456, and mary456 is a follower of john123.

3. [**13 points**] Download the data file `congestion.csv` from D2L. Each line in the data file has the following 4 comma-separated values:

```
outlook,temperature,construction,class
```

For this question, you need to write a Hadoop program that computes the entropy of each attribute—outlook, temperature, and construction—with respect to the class variable. The class variable here corresponds to the last column in each line (which indicates whether there is congestion).

Note that a serial version of the Java program to compute entropy is already provided with the `hw5.tar` file downloaded in question 1. Your task is to write the corresponding Hadoop version of the algorithm. You may use the entropy function from this code for your Hadoop implementation. Try to execute and understand the serial program first to see how it works if you want to use the given function.

Download the result generated by your Hadoop program into a file called `result_q3.txt`. Your reducer output should contain the following key-value pair, where the key is the attribute name (outlook, temperature, or construction) and the value is its entropy. You may hard-code the attribute name into your Hadoop program.

Create an archived zip/tar file for question 3. The file must contain the source code of your Hadoop program as well as the `result_q3.txt` file. Submit the zip/tar file to D2L.

4. **[10 points]** Repeat question 3 by writing a Hadoop python streaming program to compute the entropy of each attribute in the `congestion.csv` file. The mapper and reducer python programs should be written using only standard python library (including the math and sys library). You should not use numpy, pandas, or scikit-learn to compute entropy. Create an archived zip/tar file that contains the source code, `mapper.py` and `reducer.py` files along with the results of your Hadoop program downloaded from HDFS. You can name the result file as `result_q4.txt`. Submit the zip/tar file to D2L.