CSE427 – Homework 3

M. Neumann

Due THU 02/11/2016 10am

Getting Started

Update your SVN repository. Find instructions on how to checkout, update, and commit to your SVN repository here: `http://sites.wustl.edu/neumann/resources/cse427s_resources/`

When needed, you will find additional materials for *homework x* in the folder hwx. So, for the current assignment the folder is hw3.

Hint: You can **check your submission** by viewing your SVN repository `https://svn.seas.wustl.edu/repositories/<yourwustlkey>/cse427s_sp16` in a web browser.

Indicating Group Work

Use the file `partners.txt` to indicate group work. Follow these instructions exactly, to ensure to get credit!

- `partners.txt` needs to include up to 2 wustlkeys in the first two lines (one line per wustlkey)
- **first line/wustlkey is the repository, where the solution is located**. We will **only** consider the submission in this repository!
- Every student in a group needs to have **the same** `partners.txt` in the hwx folder in their repository (indicating that the partnership are **mutually accepted**)!
- If you do not have a partner, try to find one. If you want to submit on your own, indicate your wusltkey in the first line of `partners.txt` and leave the second line blank.

Problem 1: Average Word Length (40%)

In this problem you will write a MapReduce program that reads any text input and computes the average word length of all words that start with each character. The solution should be **case-sensitive**.

Use the following **test input** to test your implementation:

```
No now is definitely not the best time
```

The output for this test input (using one Reducer) would be:

```
N    2.0
b    4.0
d    10.0
i    2.0
n    3.0
t    3.5
```

(a) Write down the Mapper output and the Reducer input for the **test input** provided above.

(b) Implement the Driver, Mapper, and Reducer performing the average word length computation. Use the stubs provided in:

```
~/workspace/avgwordlength/src/stubs
```

Test your implementation on the test input. Make sure that there are no errors when executing the MapReduce job. You can not get (partial) credit for this question if we cannot successfully run your job.

(c) Run your implementation on the `shakespeare` folder in HDFS (make sure the folder only contains the following files: poems, histories, comedies, and tragedies). What are the average word lengths for the following letters:

- A
- W
- a
- t
- z

Submit your answers to (a) and (c) by editing the `hw3.txt` file in your SVN repository. Submit your answer to (b) by storing all required classes to a jar file named **`AvgWordLength.jar`** and add this to the hw3 folder in your SVN repository.

Problem 2: Unit Testing MapReduce jobs (20%)

In this Problem, you will write unit tests for the average word count MapReduce program you implemented in problem 1. (Testing MapReduce programs will be covered in Lab 2 on TUE 9th of Feb. So, you might want to do this problem after the Lab.)

(a) Implement three tests for testing the average word length implementation, one each for the Mapper, Reducer, and the entire MapReduce flow. Print out the result set on the console for each test.

(b) Run the tests. Provide a screen-shot showing the results for the tests (including your command if you use the command line).

Submit your answer to (a) by adding `TestAvgWordLength.java` to the hw3 folder in your SVN repository. Submit your answer to (b) by renaming the screen-shot in `problem2b.png` (you can also use .jpg) and `svn add problem2b.png` to the `hw3` folder in your SVN repository.

Problem 3: Passing a Parameter via the Command Line (40%)

You will write an Average Word Length program that uses a Boolean parameter called *caseSensitive* to determine whether the Mapper class should treat upper and lower case letters as different or whether all letters should be converted to lower case.
Preparation: Copy the package including the Driver, Mapper, and Reducer code you have written earlier from `~/workspace/avgwordlength/src/` to the following directory:

```
~/workspace/toolrunner/src/
```

Note on Copying Source Files
You can use Eclipse to copy a Java source file from one project or package to another by right-clicking on the file and selecting Copy, then right-clicking the new package and selecting Paste. If the packages have different names (e.g. if you copy from averagewordlength.solution to toolrunner.stubs), Eclipse will automatically change the package directive at the top of the file. If you copy the file using a file browser or the shell, you will have to do that manually.

(a) Modify the `AvgWordLength` driver to use ToolRunner. As before, test your implementation on the **test input** provided in Problem 1. Are there any differences in the job execution?

(b) Modify the `LetterMapper` to use the configuration parameter `caseSensitive` to either perform case sensitive processing (leave the letters as they are) or insensitive processing (convert all letters to lower-case) by writing a `setup()` method to get the value of the parameter. Case sensitive processing should be your **default**. As before, test your implementation on the **test input** provided in Problem 1. Are there any differences in the job execution?

(c) Run your implementation providing `caseSenstive=false` as a runtime parameter on the `shakespeare` folder in HDFS (make sure the folder only contains the following files: poems, histories, comedies, and tragedies). Provide the command for this the `hw3.txt` file. Does the order of command line inputs matter (what happens if you specify the parameter after the output directory)? What are the average word lengths for the following letters:

- a
- w
- z

(d) What other way is there to pass a parameter to the Mapper (not using the command line)? Which class needs to be updated for this? Provide the lines of code needed to pass a parameter programmatically to the Mapper in the `hw3.txt` file. Which way do you prefer and why?

Submit your answers to all parts (a-d) by editing the `hw3.txt` file in your SVN repository. Additionally, add a jar file named **`AvgWordLength_ToolRunner.jar`** file containing all required classes to run Average Word Length case sensitive or case insensitive based on the respective command line input to the hw3 folder in your SVN repository (this implementation will give (partial) credit for (a) and (b)).

Bonus Problem (5% up to a max. of 100%) - no group work!

Write a review for this homework and store it in the file `hw3_review.txt` provided in your SVN repository. This file should only include the review, **no other information** such as name, wustlkey, etc. Remember that you are not graded for the content of your review, solely it's completion.

You can only earn bonus points if you write **at least 50 words**. Bonus points are given to the **owner of the repository only** (no group work!).