SI 601 Winter 2016 Lab 5 (20 points)

Due at 5:30pm on Wednesday, Feb. 10, 2016

This lab is to familiarize you with the process of writing MapReduce code and PySpark code, and then running it on a Hadoop cluster.

In the class, I showed how to compute word counts for text files using MapReduce and PySpark. Now you need to compute the bigram counts. A bigram is every sequence of two adjacent words in a string of words. For the sake of simplicity, we will treat each line in the input text file as a string. For example, if the text is:

one two three four
three four five six

Then we should have these unique bigrams and counts:

"one two" 1
"two three" 1
"three four" 2
"four five" 1
"five six" 1

To get started, download the ' si601_w16_lab5.zip' file, unzip it, and rename 'si601_w16_lab5_mrjob.py' as 'si601_w16_lab5_mrjob_youruniquename.py', and rename 'si601_w16_lab5_spark.py' as 'si601_w16_lab5_spark_youruniquename.py'.

Part 0. Finding the input data (0 points)

The input data for lab 5 is in the HDFS directory “/user/yuhangw/si601w16lab5_ebooks” on the Fladoop cluster.

First ssh into flux-login.engin.umich.edu, then run this command:

hadoop fs -ls /user/yuhangw/si601w16lab5_ebooks

and you should see:

```
[yuhangw@flux-login2 ~]$ hadoop fs -ls /user/yuhangw/si601w16lab5_ebooks
Found 9 items
-rw-r--r--   3 yuhangw hadoop     167518 2016-01-31 19:40 /user/yuhangw/si601w16lab5_ebooks/ebooks11.txt
-rw-r--r--   3 yuhangw hadoop    1154501 2016-01-31 19:40 /user/yuhangw/si601w16lab5_ebooks/ebooks1268.txt
-rw-r--r--   3 yuhangw hadoop     717574 2016-01-31 19:40 /user/yuhangw/si601w16lab5_ebooks/ebooks1342.txt
-rw-r--r--   3 yuhangw hadoop     594933 2016-01-31 19:40 /user/yuhangw/si601w16lab5_ebooks/ebooks1661.txt
-rw-r--r--   3 yuhangw hadoop    3291648 2016-01-31 19:40 /user/yuhangw/si601w16lab5_ebooks/ebooks2600.txt
-rw-r--r--   3 yuhangw hadoop    1257296 2016-01-31 19:40 /user/yuhangw/si601w16lab5_ebooks/ebooks2701.txt
-rw-r--r--   3 yuhangw hadoop     359508 2016-01-31 19:40 /user/yuhangw/si601w16lab5_ebooks/ebooks27827.txt
-rw-r--r--   3 yuhangw hadoop     610155 2016-01-31 19:40 /user/yuhangw/si601w16lab5_ebooks/ebooks76.txt
-rw-r--r--   3 yuhangw hadoop     448689 2016-01-31 19:40 /user/yuhangw/si601w16lab5_ebooks/ebooks84.txt
```

There are 9 ebooks in this HDFS directory. These are the input files.

Part 1. MapReduce (10 points)

Add code to si601_w16_lab5_mrjob_youruniquename.py where specified. Then use FileZilla (or your SFTP client of choice) to copy your code to your home directory on flux-xfer.engin.umich.edu.

To run your on Fladoop cluster:

First follow instruction at http://caen.github.io/hadoop/user-hadoop.html#mrjob to create your .mrjob.conf file, use 'si601w16' as queuename.

Then run

```
$ module load python-hadoop/2.7
$ python2.7 si601_w16_lab5_mrjob_youruniquename.py -r hadoop --no-output
hdfs:///user/yuhangw/si601w16lab5_ebooks -o si601w16lab5_output_mrjob
```

If your code works, it will create the output directory si601w16lab5_output_mrjob under your home directory in HDFS.

Then run this command:

```
hadoop fs -ls si601w16lab5_output_mrjob
```

If you see something like

```
[yuhangw@flux-login2 ~]$ hadoop fs -ls si601w16lab5_output_mrjob
Found 2 items
-rw-r--r--   3 yuhangw hadoop          0 2016-01-31 19:51 si601w16lab5_output_mrjob/_SUCCESS
-rw-r--r--   3 yuhangw hadoop    6283309 2016-01-31 19:51 si601w16lab5_output_mrjob/part-00000
```

(Of course, you should see your uniquename instead of mine.)

You should then copy the files from HDFS to your home directory in the local file system by typing

```
hadoop fs -getmerge si601w16lab5_output_mrjob si601w16lab5_output_mrjob.txt
```

Your should see the file si601w16lab5_output_mrjob.txt in your current directory. Your goal is to make sure that your output file contains the same counts as si601w16lab5_output_mrjob_desired_output.txt

Note that the order of lines in your bigram output file may be different from mine, and that is OK.

Part 2. Spark (10 points)

In this part, you will compute the same bigram counts from the same input files using PySpark. The logic is the sill the same, except that the bigrams are to be sorted in decreasing order of frequency in the output. For bigrams with the same frequency, they should be sorted alphabetically.

Add code to si601_w16_lab5_spark_youruniquename.py where specified. Then use FileZilla (or your SFTP client of choice) to copy your code to your home directory on flux-xfer.engin.umich.edu.

To run your on Fladoop cluster:

spark-submit --master yarn-client --queue si601w16 --num-executors 2 --executor-memory 1g --executor-cores 2 si601_w16_lab5_spark_youruniquename.py hdfs:///user/yuhangw/si601w16lab5_ebooks si601w16lab5_output_spark

If your code works, it will create the output directory si601w16lab5_output_spark in HDFS.

Then run

hadoop fs -ls si601w16lab5_output_spark

you should see something like:

```
[yuhangw@flux-login2 ~]$ hadoop fs -ls si601w16lab5_output_spark
Found 2 items
-rw-r--r--   3 yuhangw hadoop          0 2016-01-31 20:03 si601w16lab5_output_spark/_SUCCESS
-rw-r--r--   3 yuhangw hadoop    6283309 2016-01-31 20:03 si601w16lab5_output_spark/part-00000
```

To see the content of the output file, run

hadoop fs -cat si601f15lab5_output_spark/part-00000 | head -n 10

and you should see the top 10 bigrams.

You should then copy the files from HDFS to your home directory in the local file system by typing

hadoop fs -getmerge si601w16lab5_output_spark si601w16lab5_output_spark.txt

Your should see the si601w16lab5_output_spark.txt in your current directory. Your goal is to make sure that your output file contains the same data as si601w16lab5_output_spark_desired_output.txt

What to submit:

Submit a zip file named si601_w16_lab5_youruniquename.zip containing your Python source code files and your output files.