Assignment 2
CS 4417

Due: March 19

Goal: The goal of this assignment is to gain familiarity and practical experience with the MapReduce programming model and give you experience with text processing.

Programming Language: Python

Evaluation: Part of our evaluation is through testing. Our test cases will NOT be made available to you before submission. It is your responsibility to test extensively. For the different parts of the assignments do not concern yourself with stop words or changing the case of letters

Part 0: VM Setup
The Department of Computer Science has a cluster where Cloudera has been installed. Cloudera is a company that provides Apache Hadoop. Each student will be provided with a virtual machine (VM) that is hosted on the cluster.

To use the VM you need to retrieve the *ssh* key that is in your OWL dropbox folder. An example key looks like this:

cs4417-lab-xxxxxx-pem

where xxxxxx represents your email identifier.

You need to change the permissions so that it is user read-only. The command for doing so on a Mac or Linux machine is the following:

chmod 600 cs4417-lab-xxxxxx.pem

You can now use the key to *ssh* into the VM:

ssh -i cs4417-lab-xxxxxx-pem xxxxxx@cs4417-lab-xxxxxx.pem

You should get the following prompt:[cs4417]>
You should enter ssh cloudera@xxxxxx

The software you need to complete the assignment is on the VM. You do not need to install anything.

.

Part 1: Calculate the frequency of a term in each document (20 points)
Given a set of documents, calculate the frequency of a term in each document. The output should be the term, document and number of occurrences of the term in the document. This is different from the example presented in the lectures in that the example focused on one document. In this example, the final output should consist of pairs in the following form:

((term, document identifier), count)

Submission: You should submit a zip file with the name *Part1.zip*. When unzipped there should be two files: *mapper.py* and *reducer.py*.

Part 2: Count Bigrams (15 points)
Take the word count example and extend it to count bigrams which refers to sequences of two consecutive words.

You should make use of Hadoop for this part.

Submission: You should submit a zip file with the name *Part2.zip*. When unzipped there should be two files: *mapper.py* and *reducer.py*.

Part 3: Count Unique Bigrams (15 points)
This is an extension of part 2 where you count the number of unique bigrams. One approach is to use two MapReduce passes. The first is what you did for Part 2 and the second is something you need to develop.

Submission: You should submit a zip file with the name Part3.zip. When unzipped there should be two files for each MapReduce pass, *i*: *mapperi.py* and *reduceri.py*. For example, if *i* is 1 then you should have *mapper1.py* and *reducer1.py* and if *i* is 2 then you should have *mapper2.py* and *reducer2.py*.

Part 4: Term-Frequency-Inverse Document Frequency in MapReduce (40 points)
The *tf-idf* metric is used to determine the importance of a word within a document. You are to write a program that uses the MapReduce paradigm. However, you do not have to use Hadoop for doing so. It is sufficient to use pipes to test the program.

The formula of tf-idf for document d and term t is the following:

$$\text{tf-id}_{t,d} = \text{tf}_{t,d} / N * \log_{10}(D / \text{df}_t)$$

where $\text{tf}_{t,d.}$ is the number of occurrences of the term t in document d, N is the total number of words in document d, D is the total number of documents and df_t is the number of documents that the term t occurs in. There are variations of the formula, but you should use the above formula since our test cases assume the above.

This requires multiple MapReduce jobs (more than 2). The first MapReduce (MR) job should calculate the term count for each term and document ($\text{tf}_{t,d}$).

The second MR job should calculate df_t for each term.

You should figure out the rest of the MR jobs needed.

For this assignment the number of documents is needed. You should have a file called *inputParameters*. This file should have one number which represents the number of documents. This will make it easier for the TAs to test.

Submission: You should submit a zip file with the name Part3.zip. When unzipped there should be two files for each MapReduce pass, *i*: *mapperi.py* and *reduceri.py*. For example, if *i* is 1 then you should have *mapper1.py* and *reducer1.py* and if *i* is 2 then you should have *mapper2.py* and *reducer2.py*.

Part 4: Writeup (10)

Please complete the following:

- For each part describe the input and output for each MR job.

- Please answer the following questions
  - What would you like to have done given more time?
  - How difficult was it to implement? How difficult would it be to implement another task, given this experience? What would be straightforward? What would take more time?

IMPORTANT: Keep a copy of the assignment outside of the VM.