NYU CS6513 Big Data Spring 2022

Assignment 1 - Hadoop HDSF & MapReduce – 100 points

Due Date: ***Wednesday, February 23, 11PM Eastern***

Abstract:

Show proficiency using HDFS and writing a MapReduce program, including submitting to Hadoop and getting results out of HDFS.

**** Give attribution to any code you use that is not your original code ****

Submission

Submit all code, pictures, and output files and pictures **as a ZIP**, using your **id and hw1:** *for example, jcr365-hw1.zip.* If we cannot run your program(s), you will not get full credit.

1. HDFS 15 points

Running any version of Hadoop (HPC, docker or otherwise), submit screen grabs (a picture in jpg or other suitable format) of the following:

a) create a directory **in HDFS** with this format: **netid-bd22** (e.g. mine will be 'jcr365-bd22'). Submit a screen grab of the output of a *Hadoop file listing* showing your home directory and your new directory in it.

b) Create a directory for the homework problem 1.2 (bigram count), and extract all input files into it. Call this directory as follows: hw1.2, e.g. mine will be hw1.2. Submit a picture of directory listings or otherwise show the input files in it.

2. Language Models with MapReduce 85 Points

Definition: N-Grams

In the simplest form, a language models describes the probability of words appearing in a sentence.

A unigram models the probability of a single word appearing in the corpus, while a bigram models the probability of two words appearing in an exact sequence. An n-gram, then, models the probability of a sequence of n words appearing consecutively.

See https://www.techtarget.com/searchenterpriseai/definition/language-modeling#:~:text=Language%20modeling%20(LM)%20is%20the,basis%20for%20their%20word%20predictions.

As an example, given the following **text (**as *the entire universe of words)*: "The Cat in the Hat is the best cat in the hat", a **unigram** language model would be: (using fractions for clarity)

the, 4/6
cat, 2/6
in, 2/6
hat, 2/6
is, 1/6
best, 1/6

A bigram (n-gram, n=2) count would be:
the cat, 1/8
cat in, 2/8
in the, 2/8
the hat, 2/8
hat is, 1/8
is the, 1/8
the best, 1/8
best cat, 1/8

For unigrams, no words precede the n-gram, so the probability of ‘cat’ appearing anywhere in the corpus is 2/6 using maximum likelihood estimation (MLE, note this is a very simplistic model – the closed universe model).

For bigrams, we could say that the probability of the ***phrase*** ‘the cat’ is 1/8. However, most likely we are not interested in the probability of the phrase, but in the conditional probability of ‘cat’ given that the word ‘the’ has been seen. In our corpus example, the probability of ‘cat’ given ‘the’, P(cat|the), is given by Bayes theorem:

P(B given A) = P(A and B) / P(A)

In this homework, let’s approximate this using the closed-corpus assumption (no unseen words exist):

P(cat|the) = P(the cat) / P(the) = (1/8) / (4/6) = 0.083

P(hat | in the) = P(in the hat) / P(in the)

Solve: You need to compute the unigrams, bigrams and trigram probabilities in the input.

Your input is lines of text.
Unigram: a single word
Bigram: two consecutive words in the input sequence
Trigram: three consecutive words

Note: with Hadoop you can pass it individual files, directories (which are recursed) or simple wildcard patterns to match multiple files.

IMPORTANT: You **cannot** solve this by ***a single map/reduce*** program. Why? Recall the discussion in class: mapper and reducers cannot hold/keep state across object instances. Your mapper is not guaranteed to exist past a single *input split*. Sop you cannot assume you could count words **and** the denominator correctly in a program

Hints:

- You will need to run multiple map/reduce 'jobs'. In Java, you could still use 1 driver.
- You can define the output keys of the mapper…..

Homework Rules:

- punctuation does NOT count; so the words is '(1991)' and '1991'are the same.
  You must parse your input: **remove all characters not in this set: [a-z, A-Z, 0-9]** ;

- all text should be normalized to lowercase

- Ignore lines with less than 3 words.

- Input should be lines of text (separated by new line and/or carriage return)

Input for this problem: **hw1dir1.zip** (provided in class website)

Write your own code in your language of choice. Your code **MUST** run in Hadoop MapReduce. For Python, use Hadoop streaming. Submit the result and code.

3. EXTRA CREDIT: 25 points

Using the solution of problem 2, print the word in this sequence with the highest probability:

united states ________

that is, for P(x | united states) = p , find the x for the highest p