NYU CS6513 Big Data **Spring 2023**

Assignment 1 - Hadoop HDSF & MapReduce – 100 points

Due Date: ***Friday, February 17, 11:55PM 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 Instructions

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 Dataproc, docker, AWS, or any), submit screen grabs (a picture in jpg or other suitable format) of the following:

a) create a directory **in HDFS** with this format: **netid-bd23** (e.g. mine will be 'jcr365-bd23'). 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 (trigram 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

Background and Definition: N-Grams

A language models LM describes the probability of words appearing in a sentence or corpus.

A unigram LM models the probability of a single word appearing in the corpus, but an ***n-gram*** LM models the probability of *the* ***n_th*** word appearing given the words n-1, n-2, … .

See: An Introduction to N-grams: What Are They and Why Do We Need Them? - XRDSXRDS (acm.org)

Let P(w) be the probability of w:

As an example, given the following corpus: “The Cat in the Hat is the best cat in the hat”, a **unigram** LM language model would be: (using fractions for clarity)

P(the) = 4/12
P(cat) = 2/12
P(in) = 2/12
P(hat) = 2/12
P(is) = 1/12
P(best) = 1/12

For unigrams, the probability of ‘cat’ appearing anywhere in the corpus is 2/12 using maximum likelihood estimation MLE (a.k.a. ***word count***) - note this is a very simplistic model – the closed universe model.

A bigram (n-gram, n=2) LM:

P(the cat) = 1/8
P(cat in) = 2/8
P(in the) = 2/8
P(the hat) = 2/8
P(hat is) = 1/8
P(is the) = 1/8
P(the best) = 1/8
P(best cat) = 1/8

However, most likely is that 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 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, so no smoothing for those statisticians in class):

P(cat|the) = P(the cat) / P(the) = (1/8) / (4/12) = 0.375

Homework Rules:

- punctuation does NOT count; so the words is ‘(1991)’ and ‘1991’are the same.
  You must parse your input: replace **all characters not in this set: [a-z, A-Z, 0-9]** with spaces.
- 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, but your code MUST be Hadoop MapReduce. For Python, use Hadoop streaming. Submit the result and code.

Solve: Compute the Bigram LM, of the form "word1, word2", P(word2 | word1)

P(w2 | w1) = P(w1 w2) / P(w1)

$$= \frac{\frac{count(w1\ w2)}{total(bigrams)}}{\frac{count(w1)}{total(unigrams)}}$$

- Your input is lines of text.
- Hadoop programs can individual files, directories (which are recursed) or simple wildcard patterns to match multiple files.
- You can/could use multiple jobs!
-

IMPORTANT: Solving this by ***a single map/reduce*** program is very difficult. 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*, and your reducer is only guaranteed to exist for one 1 (unless you use a single reducer).

Hints:

- Compute 4 things: unigrams counts, bigrams counts, total unigrams, total bigrams

- State does not save: you cannot count totals ….. unless you use Hadoop global counters…

- You define what the keys and values of the mapper…..and the values…..

- you can daisy chain jobs

3. EXTRA CREDIT: 25 points

Using the solution of problem 2, find the x below (print the word in this sequence with the highest probability)

united states ___x___

For P(x | united states) = p , find the x with the highest p