

CS-GY 6513 Big Data

Assignment 1

Runze Li
rl5083@nyu.edu

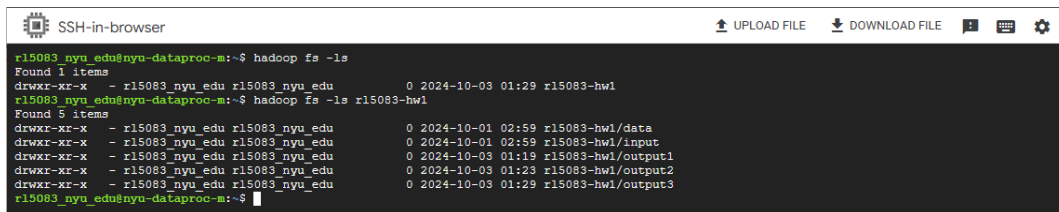
October 6, 2024

1 HDFS

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

a) Create a directory in HDFS with this format: netid-hw1 (e.g. mine will be 'jcr365-hw1'). Submit a screen grab of the output of a Hadoop file listing showing your home directory and your new directory in it.

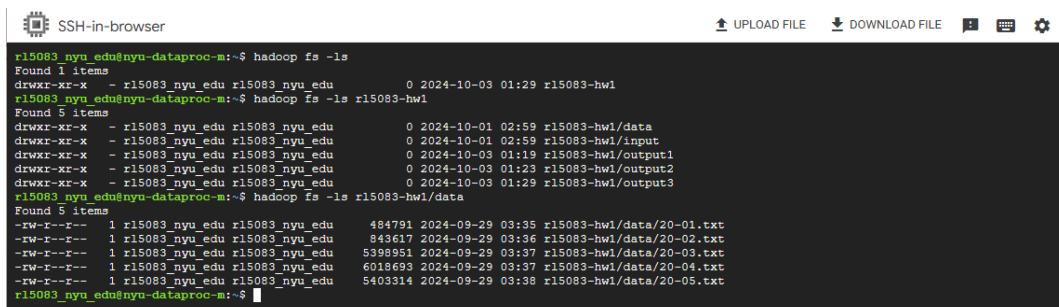
```
$ hadoop fs -mkdir rl5083-hw1
```



```
SSH-in-browser
Found 1 items
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-03 01:29 rl5083-hw1
rl5083_nyu_edu@nyu-dataproc-m:~$ hadoop fs -ls rl5083-hw1
Found 5 items
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-01 02:59 rl5083-hw1/data
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-01 02:59 rl5083-hw1/input
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-03 01:19 rl5083-hw1/output1
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-03 01:23 rl5083-hw1/output2
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-03 01:29 rl5083-hw1/output3
rl5083_nyu_edu@nyu-dataproc-m:~$
```

b) Create a subdirectory in HDFS, 'netid-hw1/data' and extract all input files into it. Submit a picture of directory listings or otherwise show the input files in it.

```
$ hadoop fs -mkdir rl5083-hw1/data
$ hadoop distcp gs://nyu-dataproc-hdfs-ingest/20-01.txt /user/rl5083_nyu_edu/rl5083-hw1/data
$ hadoop distcp gs://nyu-dataproc-hdfs-ingest/20-02.txt /user/rl5083_nyu_edu/rl5083-hw1/data
$ hadoop distcp gs://nyu-dataproc-hdfs-ingest/20-03.txt /user/rl5083_nyu_edu/rl5083-hw1/data
$ hadoop distcp gs://nyu-dataproc-hdfs-ingest/20-04.txt /user/rl5083_nyu_edu/rl5083-hw1/data
$ hadoop distcp gs://nyu-dataproc-hdfs-ingest/20-05.txt /user/rl5083_nyu_edu/rl5083-hw1/data
```



```
SSH-in-browser
Found 1 items
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-03 01:29 rl5083-hw1
rl5083_nyu_edu@nyu-dataproc-m:~$ hadoop fs -ls rl5083-hw1
Found 5 items
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-01 02:59 rl5083-hw1/data
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-01 02:59 rl5083-hw1/input
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-03 01:19 rl5083-hw1/output1
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-03 01:23 rl5083-hw1/output2
drwxr-xr-x - rl5083_nyu_edu rl5083_nyu_edu 0 2024-10-03 01:29 rl5083-hw1/output3
rl5083_nyu_edu@nyu-dataproc-m:~$ hadoop fs -ls rl5083-hw1/data
Found 5 items
-rw-r--r-- 1 rl5083_nyu_edu rl5083_nyu_edu 484791 2024-09-29 03:35 rl5083-hw1/data/20-01.txt
-rw-r--r-- 1 rl5083_nyu_edu rl5083_nyu_edu 843617 2024-09-29 03:36 rl5083-hw1/data/20-02.txt
-rw-r--r-- 1 rl5083_nyu_edu rl5083_nyu_edu 5398951 2024-09-29 03:37 rl5083-hw1/data/20-03.txt
-rw-r--r-- 1 rl5083_nyu_edu rl5083_nyu_edu 6018693 2024-09-29 03:37 rl5083-hw1/data/20-04.txt
-rw-r--r-- 1 rl5083_nyu_edu rl5083_nyu_edu 5403314 2024-09-29 03:38 rl5083-hw1/data/20-05.txt
rl5083_nyu_edu@nyu-dataproc-m:~$
```

2 Beginner's Language Models with MapReduce

2.1 10 Most likely words

For this problem, you do not need to compute the unigram (single word) probabilities. Recall from the earlier explanation that the denominator is constant across all words. So, you can just count words, then output the top 10. This is a 2 job problem. The output of job1 can feed into the job2.

You must NOT lowercase or normalize the input. You must tokenize punctuation as a single word/token. For example:

"The cat is purring, so do not wake her." should tokenize as:

"The", "cat", "is", "purring", "so", ",", "do", "not", "wake", "her", "."

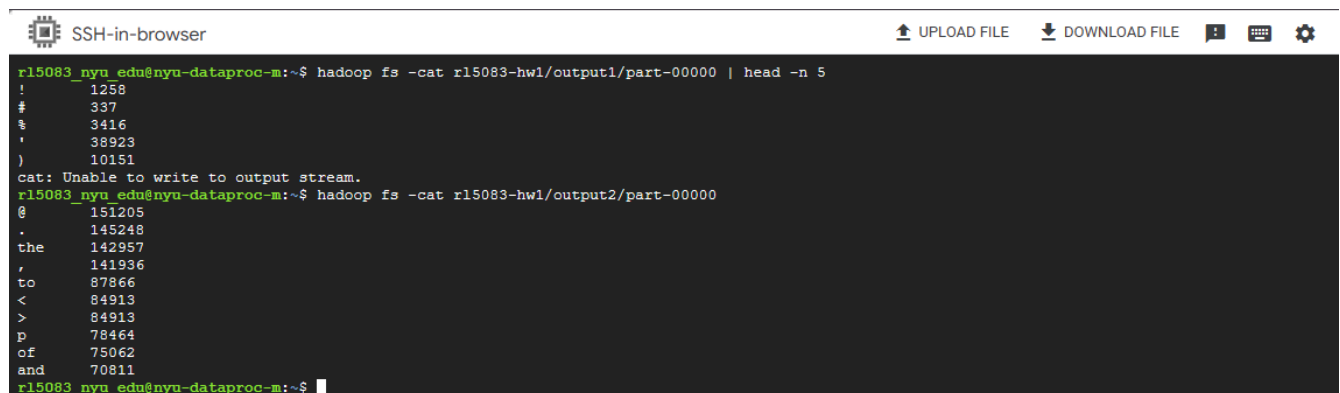
- The input is lines of text: TextInputFormat is your input data class.
- In Hadoop MapReduce, you can pass it individual files, whole directories (which are recursed) or simple wildcard patterns to match multiple files.
- Remember: Your mapper is guaranteed to exist only for a single input split. Your reducer is guaranteed to exist only for a single key.
- You have full control to define what is a key and what is a value.
- Replace all multiple spaces to a single space before.
- Ignore lines with less than 1 word.
- Mappers and Reducers cannot hold global state.
- You must use multiple jobs to solve this problem.
- Reducer parameters are one-way, one-shot iterators. You cannot double loop on the iterator.

Input for this problem: hw1text.zip (provided in class website)

After writing the code for WordCount (**mapper1.py**, **reducer1.py**) and Top-10 Words (**mapper2.py**, **reducer2.py**), we run the following command to get the result for WordCount and Top-10 Words:

```
$ hadoop jar $HADOOP_HOME/hadoop-streaming-3.3.6.jar -input /user/r15083_nyu_edu/r15083-hw1/data -output /user/r15083_nyu_edu/r15083-hw1/output1 -file mapper1.py -file reducer1.py -mapper "python mapper1.py" -reducer "python reducer1.py"
```

```
$ hadoop jar $HADOOP_HOME/hadoop-streaming-3.3.6.jar -D mapreduce.job.reduces=1 -input /user/r15083_nyu_edu/r15083-hw1/output1 -output /user/r15083_nyu_edu/r15083-hw1/output2 -file mapper2.py -file reducer2.py -mapper "python mapper2.py" -reducer "python reducer2.py"
```



```
SSH-in-browser
r15083_nyu_edu@nyu-dataproc-m:~$ hadoop fs -cat r15083-hw1/output1/part-00000 | head -n 5
!      1258
#      337
$      3416
%      38923
)      10151
cat: Unable to write to output stream.
r15083_nyu_edu@nyu-dataproc-m:~$ hadoop fs -cat r15083-hw1/output2/part-00000
@      151205
.      145248
the    142957
,      141936
to     87866
<      84913
>      84913
p      78464
of     75062
and    70811
r15083_nyu_edu@nyu-dataproc-m:~$
```

2.2 Extra credit – Simple ID Tokenizer

Use the word count given in class, or the output of Problem 2.1, and assign an increasing integer ID to each word (word) in order of decreasing count. ID's start at 1.

For example, if the word counts in the input are like this:

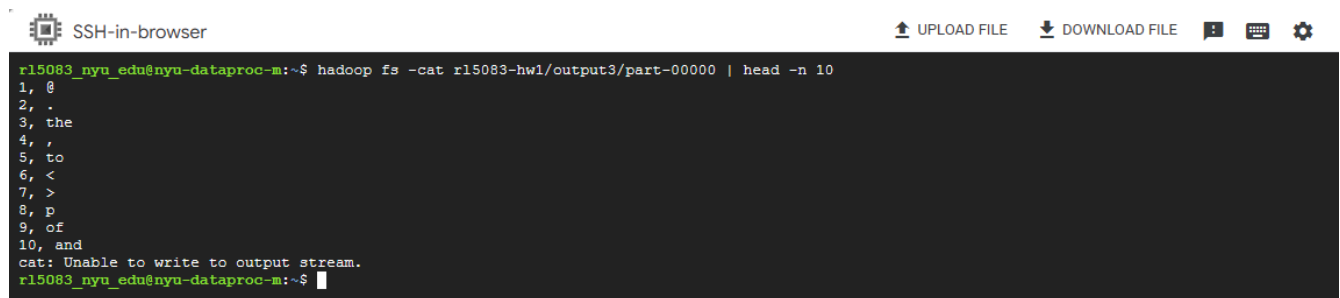
```
The, 123
house, 99
is, 88
cat, 76
...
```

Your output should be a table like this: 1, The
2, house
3, is
4, house
....

Input for this problem: hw1text.zip (provided in class website)

Here we use the result of WordCount as the input of ID-Tokenizer (**mapper3.py**, **reducer3.py**). Run the following command to get the result:

```
$ hadoop jar $HADOOP_HOME/hadoop-streaming-3.3.6.jar -D mapreduce.job.reduces=1
  -input /user/rl5083_nyu_edu/rl5083-hw1/output1 -output /user/rl5083_nyu_edu
  /rl5083-hw1/output3 -file mapper3.py -file reducer3.py -mapper "python
  mapper3.py" -reducer "python reducer3.py"
```



The screenshot shows a terminal window titled "SSH-in-browser". The terminal displays the execution of a Hadoop command to run a custom mapper and reducer. The output of the command is shown, listing the top 10 words by frequency from the input data. The words are: @, ., the, ,, to, <, >, p, of, and. The terminal also shows a message from 'cat' indicating it was unable to write to the output stream.

```
SSH-in-browser
r15083_nyu_edu@nyu-dataproc-m:~$ hadoop fs -cat r15083-hw1/output3/part-00000 | head -n 10
1, @
2, .
3, the
4, ,
5, to
6, <
7, >
8, p
9, of
10, and
cat: Unable to write to output stream.
r15083_nyu_edu@nyu-dataproc-m:~$
```