NYU CS-GY 6513 Big Data - Assignment 1 Report

~Rohan Gore (rmg9725)

Question 1

Part a:

`hdfs dfs -mkdir /user/rmg9725_nyu_edu/hw1-rmg9725`

```
rmg9725_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -ls /user/$(whoami)/hw1-rmg9725
Found 1 items
drwxr-xr-x   - rmg9725_nyu_edu rmg9725_nyu_edu          0 2025-02-21 14:01 /user/rmg9725_nyu_edu/hw1-rmg9725/data
rmg9725_nyu_edu@nyu-dataproc-m:~$
```

Part b:

`hdfs dfs -mkdir /user/rmg9725_nyu_edu/hw1-rmg9725/data`

```
rmg9725_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -mkdir /user/rmg9725_nyu_edu/hw1-rmg9725/data
rmg9725_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -ls hw1-rmg9725/data
rmg9725_nyu_edu@nyu-dataproc-m:~$
```

Uploaded file on hadoop and unzipped on hadoop itself using:
`unzip hw1text.zip`

```
rmg9725_nyu_edu@nyu-dataproc-m:~$ unzip hw1text.zip
Archive:  hw1text.zip
   creating: hw1text/
  inflating: hw1text/ep-04-02-09.txt
  inflating: hw1text/ep-04-02-10.txt
```

Move unzipped folder from hadoop to hdfs:
`hdfs dfs -put hw1text hw1-rmg9725/data/`

Check files using
`hdfs dfs -ls hw1-rmg9725/data/hw1text`

```
rmg9725_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -ls hw1-rmg9725/data/hw1text
Found 90 items
-rw-r--r--   1 rmg9725_nyu_edu rmg9725_nyu_edu     180001 2025-02-25 02:30 hw1-rmg9725/data/hw1text/ep-04-02-09.txt
-rw-r--r--   1 rmg9725_nyu_edu rmg9725_nyu_edu     544044 2025-02-25 02:30 hw1-rmg9725/data/hw1text/ep-04-02-10.txt
-rw-r--r--   1 rmg9725_nyu_edu rmg9725_nyu_edu     528415 2025-02-25 02:31 hw1-rmg9725/data/hw1text/ep-04-02-11.txt
-rw-r--r--   1 rmg9725_nyu_edu rmg9725_nyu_edu     247074 2025-02-25 02:31 hw1-rmg9725/data/hw1text/ep-04-02-12.txt
-rw-r--r--   1 rmg9725_nyu_edu rmg9725_nyu_edu     325585 2025-02-25 02:30 hw1-rmg9725/data/hw1text/ep-04-02-25.txt
-rw-r--r--   1 rmg9725_nyu_edu rmg9725_nyu_edu     191117 2025-02-25 02:30 hw1-rmg9725/data/hw1text/ep-04-02-26.txt
-rw-r--r--   1 rmg9725_nyu_edu rmg9725_nyu_edu     244443 2025-02-25 02:30 hw1-rmg9725/data/hw1text/ep-04-03-08.txt
-rw-r--r--   1 rmg9725_nyu_edu rmg9725_nyu_edu     568111 2025-02-25 02:30 hw1-rmg9725/data/hw1text/ep-04-03-09.txt
```

Question 2

1. Uploaded files and giving them executable rights

```
chmod +x mapper.py
chmod +x reducer.py
chmod +x sort_mapper.py
chmod +x sort_reducer.py
```

2. **Pass 1 → first map and reduce task → gives wordcount for every word**

   Execution Command:

```
mapred streaming \
    -D mapreduce.job.reduces=1 \
    -input hw1-rmg9725/data/hw1text \
    -output hw1-rmg9725/output_wordcount \
    -mapper mapper.py \
    -reducer reducer.py \
    -file mapper.py \
    -file reducer.py
```

   # using `-D mapreduce.job.reduces=1 \` to cap reducers to 1 so that i only get one output file of count frequencies

   Can check output files here:
   `hdfs dfs -ls hw1-rmg9725/output_wordcount`

   Print output file using:
   `hdfs dfs -cat /user/rmg9725_nyu_edu/hw1-rmg9725/output_wordcount/part-00000`

   Code Files attached in zip (`mapper.py, reducer.py`)

3. **Pass 2 → second map and reduce tasks → sorts wrt frequency count → and prints in indexed format**

   Execution Command:

```
mapred streaming \
    -D mapreduce.job.reduces=1 \
    -input hw1-rmg9725/output_wordcount \
    -output hw1-rmg9725/output_idtokenizer \
    -mapper sort_mapper.py \
    -reducer sort_reducer.py \
    -file sort_mapper.py \
    -file sort_reducer.py
```

   VERY IMPORTANT: # using `-D mapreduce.job.reduces=1 \` to use only 1 reducer so that sorted order is received in only a single file OR ELSE it might be split into multiple files and then simple concatenation of those files would result in wrong answer.

   Can check output files here:
   `hdfs dfs -ls hw1-rmg9725/output_idtokenizer`

   Print output file using:
   `hdfs dfs -cat /user/rmg9725_nyu_edu/hw1-rmg9725/output_idtokenizer/part-00000`

   Bringing final output file for question 2 to hadoop to download it easily
   `hdfs dfs -get /user/rmg9725_nyu_edu/hw1-rmg9725/output_idtokenizer/part-00000`

   Code Files attached in zip (`sort_mapper.py, sort_reducer.py`)

   Output File attached in zip (`Q2_pass2.txt`)

Question 3

1. **1st MAP-Reduce Task** → Using original mapper with modified reducer to calculate total_words along with the word and frequency pair. Also return the total_word number.

   Execution Command:

   ```
   mapred streaming \
       -input hw1-rmg9725/data/hw1text \
       -output hw1-rmg9725/output_wordcount_with_total \
       -mapper mapper.py \
       -reducer reducer_mod.py \
       -file mapper.py \
       -file reducer_mod.py \
       -numReduceTasks 1
   ```

   Check Output using:
   ```
   hdfs dfs -cat hw1-rmg9725/output_wordcount_with_total/part-00000
   ```

   Extract the _"_TOTAL__" count into another tet file using:
   ```
   hdfs dfs -cat hw1-rmg9725/output_wordcount_with_total/* | grep '__TOTAL__' | cut -f 2 > total.txt
   ```

   Push the total.text file to HDFS using:
   ```
   hdfs dfs -put total.txt hw1-rmg9725/
   ```

   Code Files attached in zip (`mapper.py, reducer_mod.py`)

2. **2nd MAP-Reduce Task → Input 1:** Output of Task 2 - Pass 1, **Input 2**: total.text
   Calculating probabilities for all the words and now returning/printing <word, prob.> pairs into text file.

   Execution Command:

```
mapred streaming \
    -input hw1-rmg9725/output_wordcount \
    -output hw1-rmg9725/output_with_probs \
    -mapper prob_mapper.py \
    -reducer prob_reducer.py \
    -file prob_mapper.py \
    -file prob_reducer.py \
    -file total.txt
```

   Check files at:
   `hdfs dfs -ls /user/rmg9725_nyu_edu/hw1-rmg9725/output_with_probs/`

   Check file content using:

   `hdfs dfs -cat hw1-rmg9725/output_with_probs/part-00000` → We can see all the words with their occurrence probabilities.

   Code Files attached in zip (`prob_mapper.py`, `prob_reducer.py`)

3. **3rd MAP-Reduce Task → Input 1:** Output of Task 2 - Pass 2 (for the sorted order), **Input 2**: Output of Task 3 - Part 2
   Calculating probabilities for all the words and now returning/printing <word, prob.> pairs into text file.

   Execution Command:

```
mapred streaming \
     -input hw1-rmg9725/output_idtokenizer,hw1-rmg9725/output_with_probs
\
     -output hw1-rmg9725/out_p3 \
     -mapper "python3 join_mapper.py $map_input_file" \
     -reducer join_reducer.py \
     -file join_mapper.py \
     -file join_reducer.py
```

   Store the 10th and 15th word in a text file using:

```
hdfs dfs -cat hw1-rmg9725/out_p3/* | awk '$1 == 10 || $1 == 15 {print}'
> final_output.txt
```

Code Files attached in zip (`join_mapper.py, join_reducer.py`)

View the final result using
`Cat final_output.txt`

```
rmg9725_nyu_edu@nyu-dataproc-m:~$ hdfs dfs -cat hw1-rmg9725/out_p3/* | awk '$1 == 10 || $1 == 15 {print}' > final_output.txt
rmg9725_nyu_edu@nyu-dataproc-m:~$ ls
final_output.txt  hw1text.zip     join_reducer.py  part-00000      prob_reducer.py  reducer_mod.py  sort_reducer.py
hw1text           join_mapper.py  mapper.py        prob_mapper.py  reducer.py       sort_mapper.py  total.txt
rmg9725_nyu_edu@nyu-dataproc-m:~$ cat final_output.txt
10      for     0.010884213169050125
15      it      0.009174496190650921
```

Output File attached in zip (`final_output.txt`)

References & Collaborators

1. *Perplexity - DeepSeek r1, Sonar Huge,*
2. *ChatGPT - 4o, o3-mini*