Design Decision:

Our mapper and reducer function is built in python.
First we have preprocessed our data. We have added the line number and a document id. We also have lowercased the words and removed the punctuation marks. Finally the common words are removed using stop words list.

We counted how many times a particular word appeared in a document using wordcount MapReduce program.

If a word appears in a document more than half of the average of occurrence times , then we add it to our stop words list. Working with Shakepear made us realize average was a good enough cut-off point because there were enough words being repeated more than this threshold.But since average can vary if maximum and minimum value has huge difference . Considering this, insead of taking average as threshold, we chose the half value of it.

Now for each file in input directory, we prepend the document id and line number for each line. After that, we block the stop words and remove all the punctuation marks.

Our local query file takes the final output from invertedIndexed file as input. Then it looks up the word from standard input . If the word is present in the file, then it prints out the document ID and line number of the word.

Manual:

Step1-5: Preprocess the Input

1. Run process.sh in your local machine.

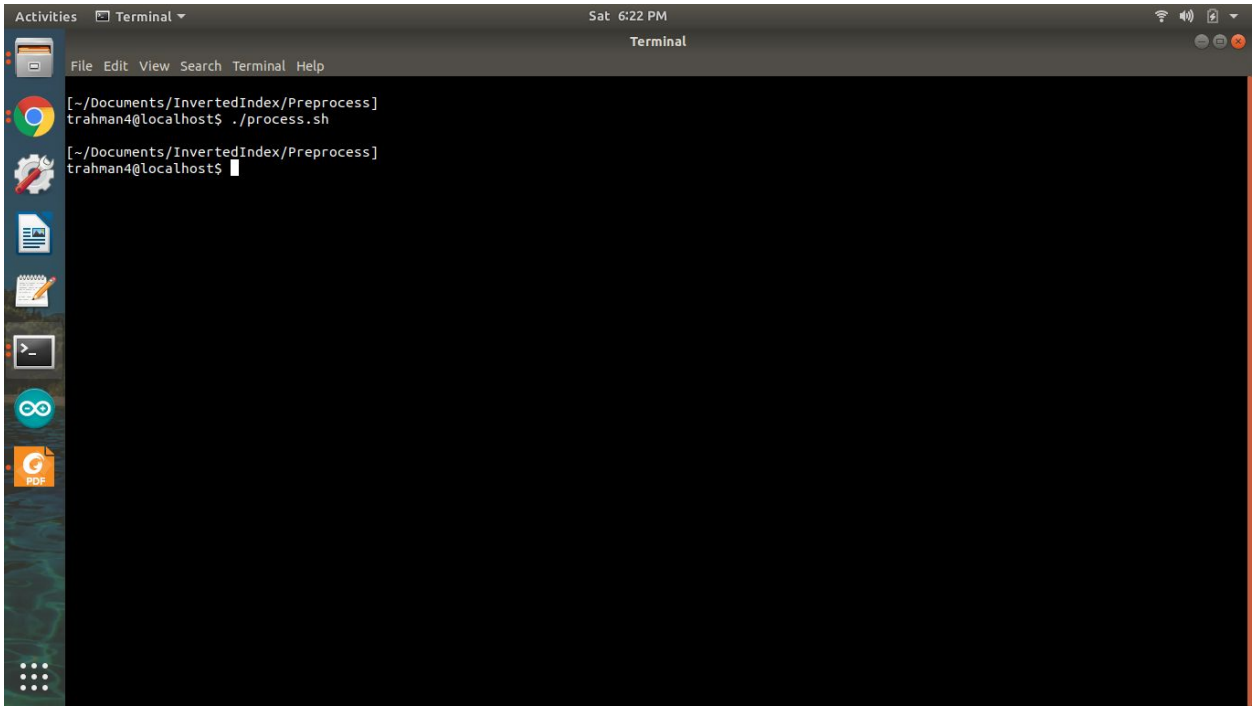


Figure 1: Run Process.sh

First this would remove the punctuation marks, add line number in the raw data and put the initially processed data into ‘clean’ directory under ‘raw’ directory.

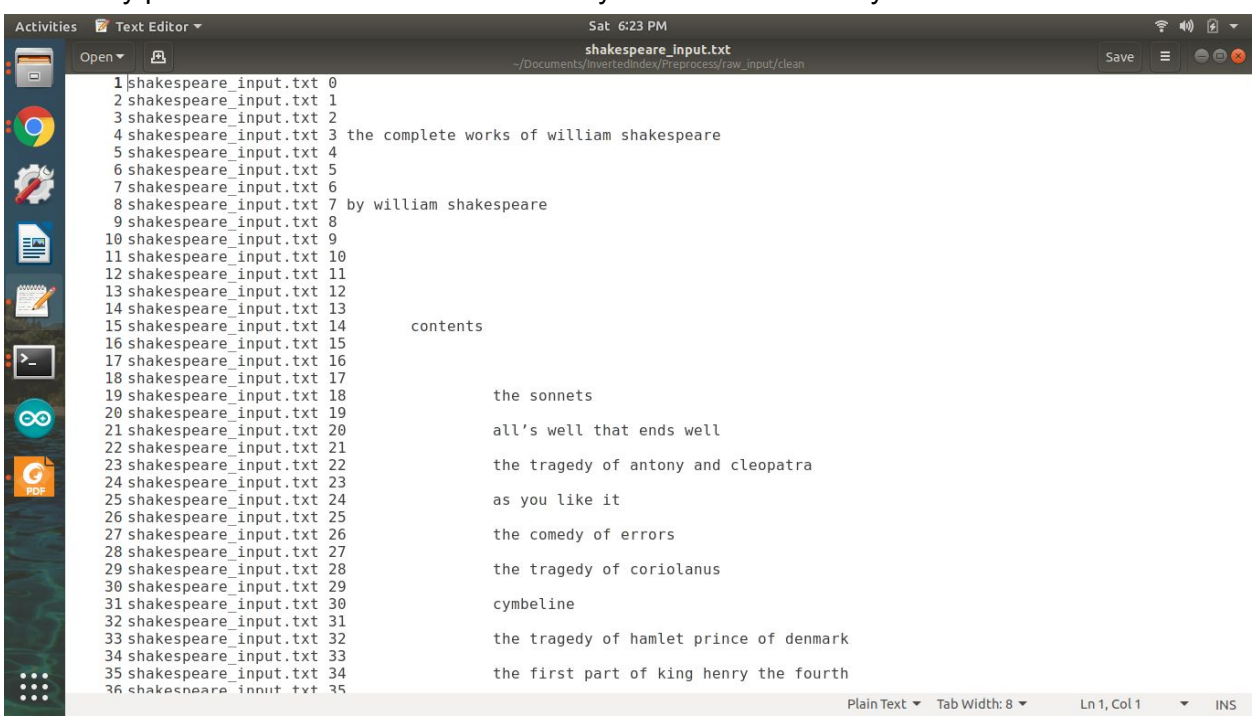


Figure 2:Creates the processed file in clean directory

This initial processed data would be saved under wordCount Mapreduce directory's processed input. Also, we would put this file under generate stop word's directory as a dirty data.

2. <u>Run WordCount on Cloud:</u>

After that, this initially processed data would have to be transferred to cloud as input to wordcount MapReduce program.

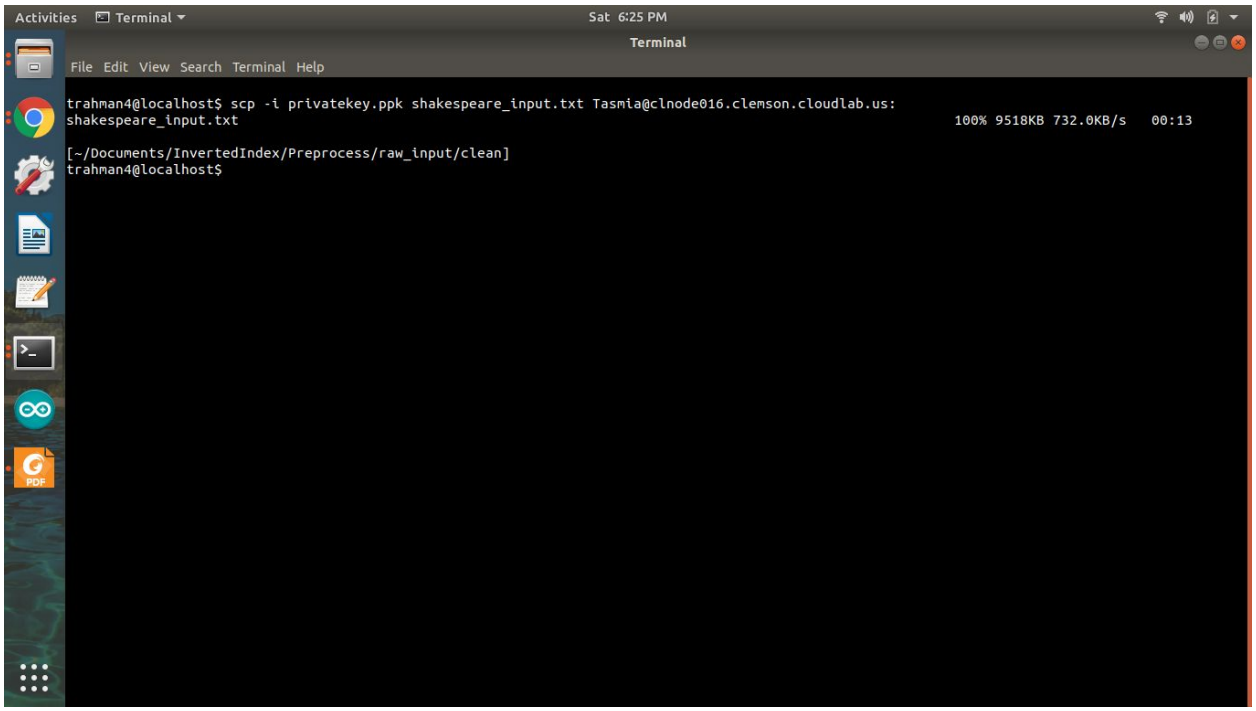
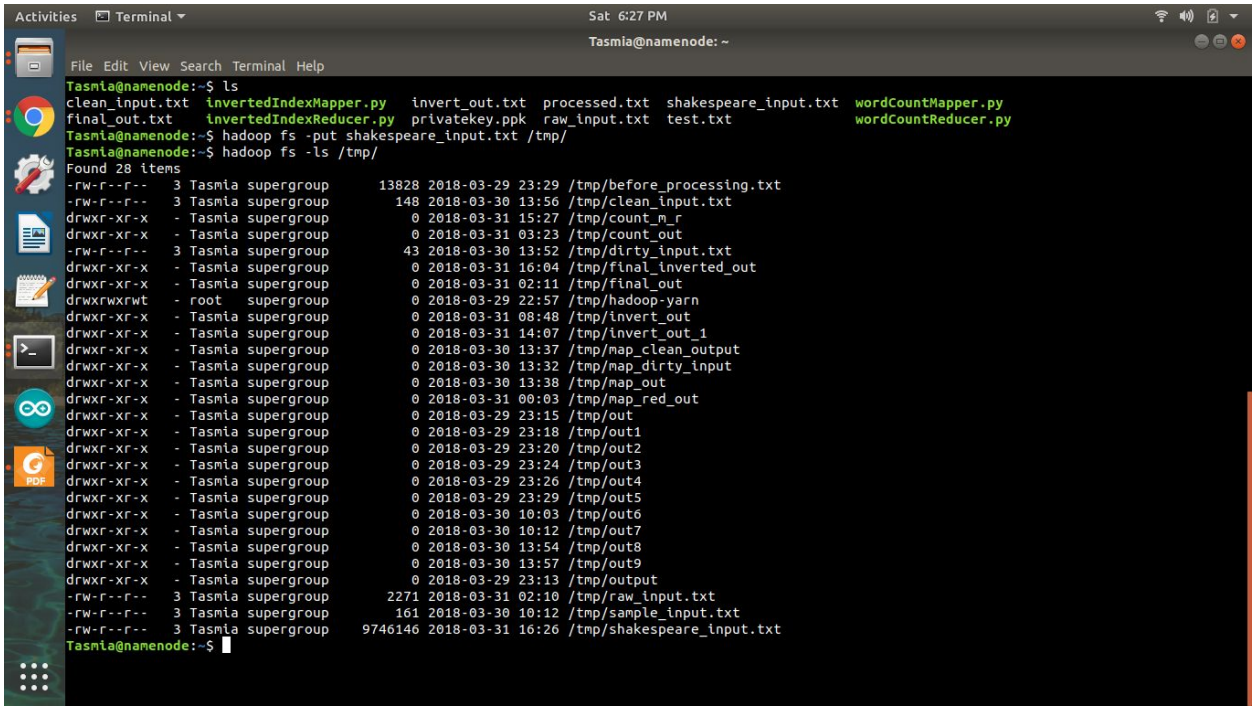
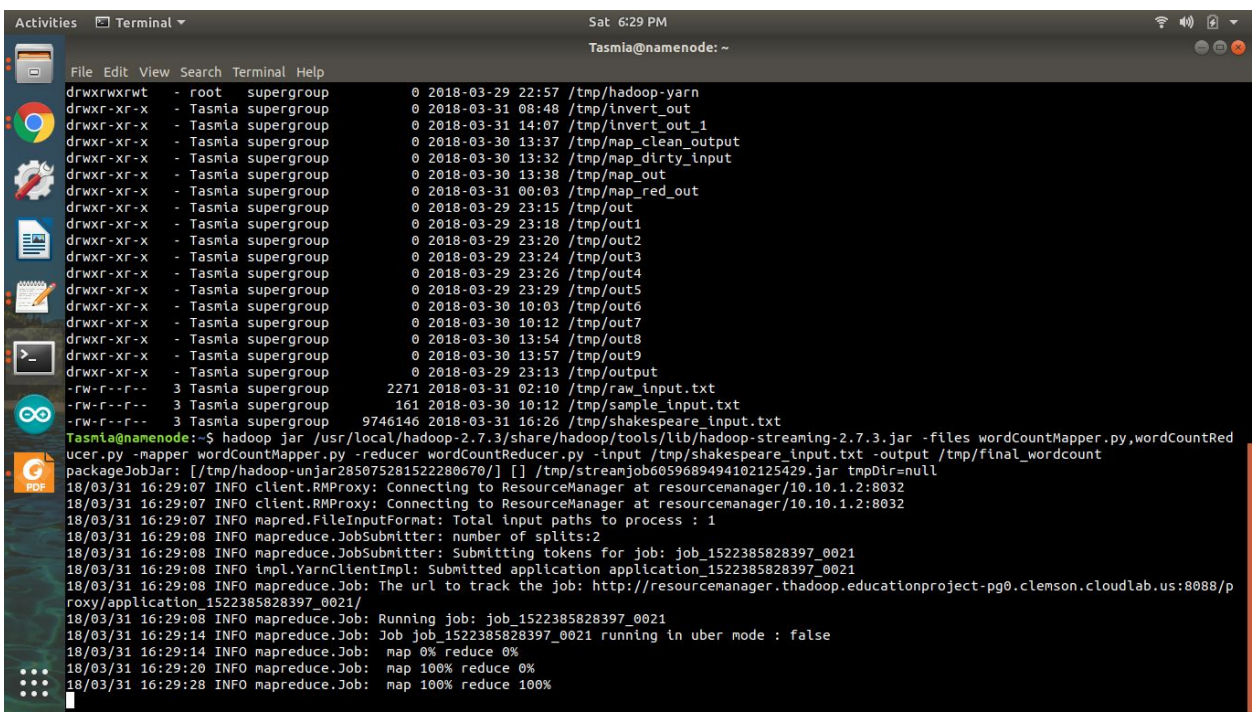


Figure 3: Copy the processes data to hadoop



```
Tasmia@namenode:~$ ls
clean_input.txt  invertedIndexMapper.py   invert_out.txt  processed.txt  shakespeare_input.txt  wordCountMapper.py
final_out.txt    invertedIndexReducer.py  privatekey.ppk  raw_input.txt  test.txt               wordCountReducer.py
Tasmia@namenode:~$ hadoop fs -put shakespeare_input.txt /tmp/
Tasmia@namenode:~$ hadoop fs -ls /tmp/
Found 28 items
-rw-r--r--   3 Tasmia supergroup      13828 2018-03-29 23:29 /tmp/before_processing.txt
-rw-r--r--   3 Tasmia supergroup        148 2018-03-30 13:56 /tmp/clean_input.txt
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 15:27 /tmp/count_m_r
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 03:23 /tmp/count_out
-rw-r--r--   3 Tasmia supergroup         43 2018-03-30 13:52 /tmp/dirty_input.txt
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 16:04 /tmp/final_inverted_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 02:11 /tmp/final_out
drwxrwxrwt   - root    supergroup         0 2018-03-29 22:57 /tmp/hadoop-yarn
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 08:48 /tmp/invert_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 14:07 /tmp/invert_out_1
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:37 /tmp/map_clean_output
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:32 /tmp/map_dirty_input
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:38 /tmp/map_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 00:03 /tmp/map_red_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:15 /tmp/out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:18 /tmp/out1
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:20 /tmp/out2
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:24 /tmp/out3
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:26 /tmp/out4
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:29 /tmp/out5
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 10:03 /tmp/out6
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 10:12 /tmp/out7
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:54 /tmp/out8
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:57 /tmp/out9
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:13 /tmp/output
-rw-r--r--   3 Tasmia supergroup       2271 2018-03-31 02:10 /tmp/raw_input.txt
-rw-r--r--   3 Tasmia supergroup        161 2018-03-30 10:12 /tmp/sample_input.txt
-rw-r--r--   3 Tasmia supergroup    9746146 2018-03-31 16:26 /tmp/shakespeare_input.txt
Tasmia@namenode:~$
```

Figure 4: Put the file into HDFS & ls to see the file



```
drwxrwxrwt   - root    supergroup          0 2018-03-29 22:57 /tmp/hadoop-yarn
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 08:48 /tmp/invert_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 14:07 /tmp/invert_out_1
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:37 /tmp/map_clean_output
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:32 /tmp/map_dirty_input
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:38 /tmp/map_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 00:03 /tmp/map_red_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:15 /tmp/out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:18 /tmp/out1
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:20 /tmp/out2
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:24 /tmp/out3
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:26 /tmp/out4
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:29 /tmp/out5
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 10:03 /tmp/out6
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 10:12 /tmp/out7
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:54 /tmp/out8
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:57 /tmp/out9
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:13 /tmp/output
-rw-r--r--   3 Tasmia supergroup       2271 2018-03-31 02:10 /tmp/raw_input.txt
-rw-r--r--   3 Tasmia supergroup        161 2018-03-30 10:12 /tmp/sample_input.txt
-rw-r--r--   3 Tasmia supergroup    9746146 2018-03-31 16:26 /tmp/shakespeare_input.txt
Tasmia@namenode:~$ hadoop jar /usr/local/hadoop-2.7.3/share/hadoop/tools/lib/hadoop-streaming-2.7.3.jar -files wordCountMapper.py,wordCountRed
ucer.py -mapper wordCountMapper.py -reducer wordCountReducer.py -input /tmp/shakespeare_input.txt -output /tmp/final_wordcount
packageJobJar: [/tmp/hadoop-unjar2850752815222806 70/] [] /tmp/streamjob6059689494102125429.jar tmpDir=null
18/03/31 16:29:07 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
18/03/31 16:29:07 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
18/03/31 16:29:07 INFO mapred.FileInputFormat: Total input paths to process : 1
18/03/31 16:29:08 INFO mapreduce.JobSubmitter: number of splits:2
18/03/31 16:29:08 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1522385828397_0021
18/03/31 16:29:08 INFO impl.YarnClientImpl: Submitted application application_1522385828397_0021
18/03/31 16:29:08 INFO mapreduce.Job: The url to track the job: http://resourcemanager.thadoop.educationproject-pg0.clemson.cloudlab.us:8088/p
roxy/application_1522385828397_0021/
18/03/31 16:29:08 INFO mapreduce.Job: Running job: job_1522385828397_0021
18/03/31 16:29:14 INFO mapreduce.Job: Job job_1522385828397_0021 running in uber mode : false
18/03/31 16:29:14 INFO mapreduce.Job:  map 0% reduce 0%
18/03/31 16:29:20 INFO mapreduce.Job:  map 100% reduce 0%
18/03/31 16:29:28 INFO mapreduce.Job:  map 100% reduce 100%
```

Figure 5: WordCount MapReduce 100% successful

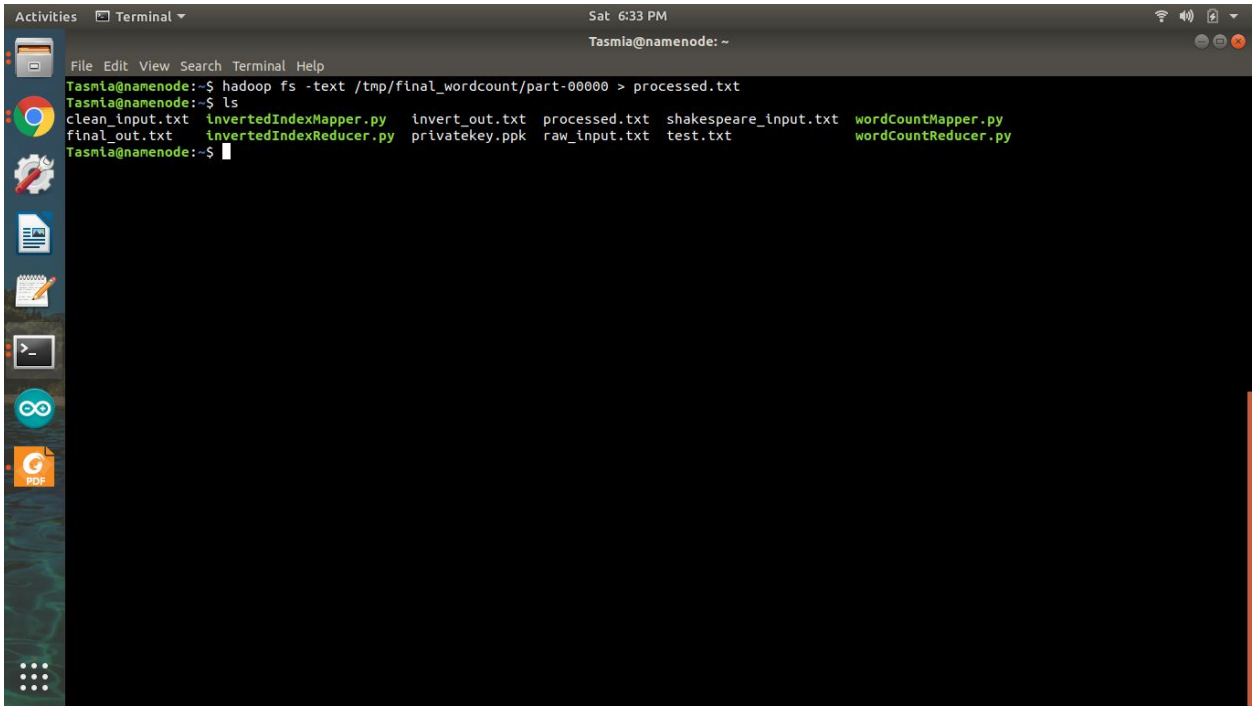


Figure 6: Copy the WordCount output to Cloud File System

This step counts how many times each word appears in a given input.

3.<u>Saving Output into Local Machine:</u>

Go to generateStopList directory. Save WordCount's output as processed.txt in this directory .

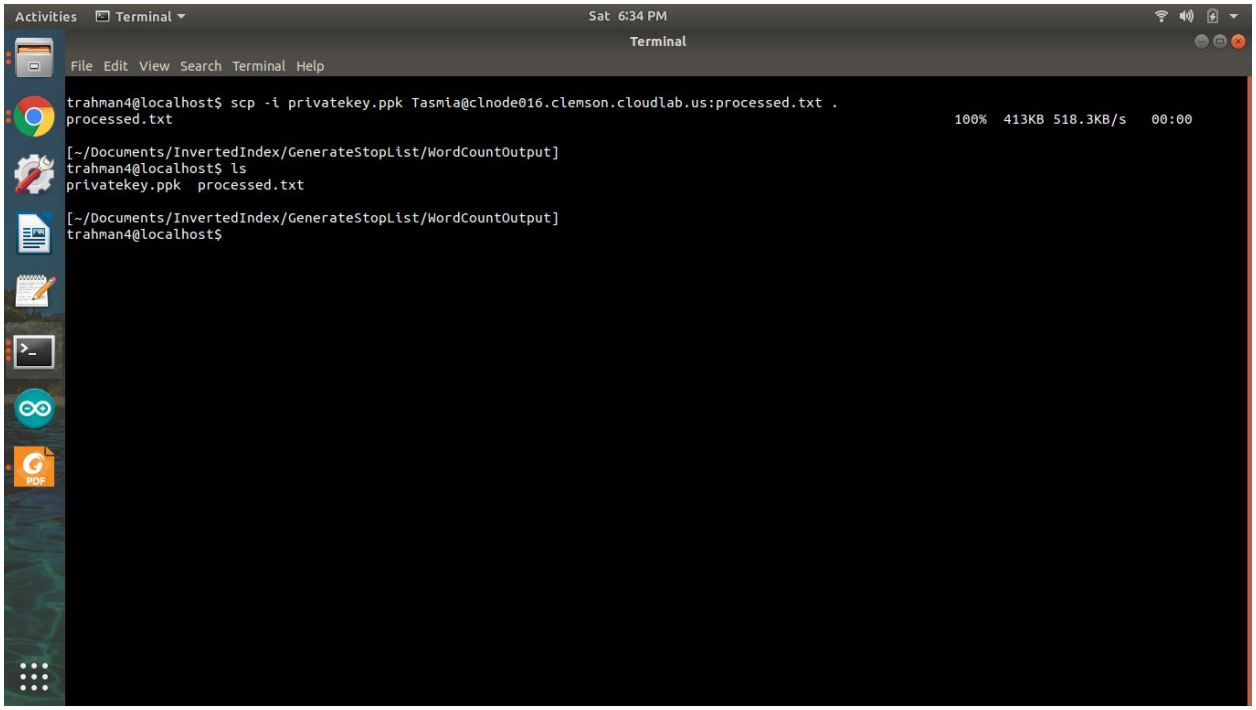


Figure 7: Copying the processed wordcount output from cloud local to WordCountOutput directoryTransfer the output from WordCount Mapreduce to Local File System

4. <u>Generate Stop Word List:</u>

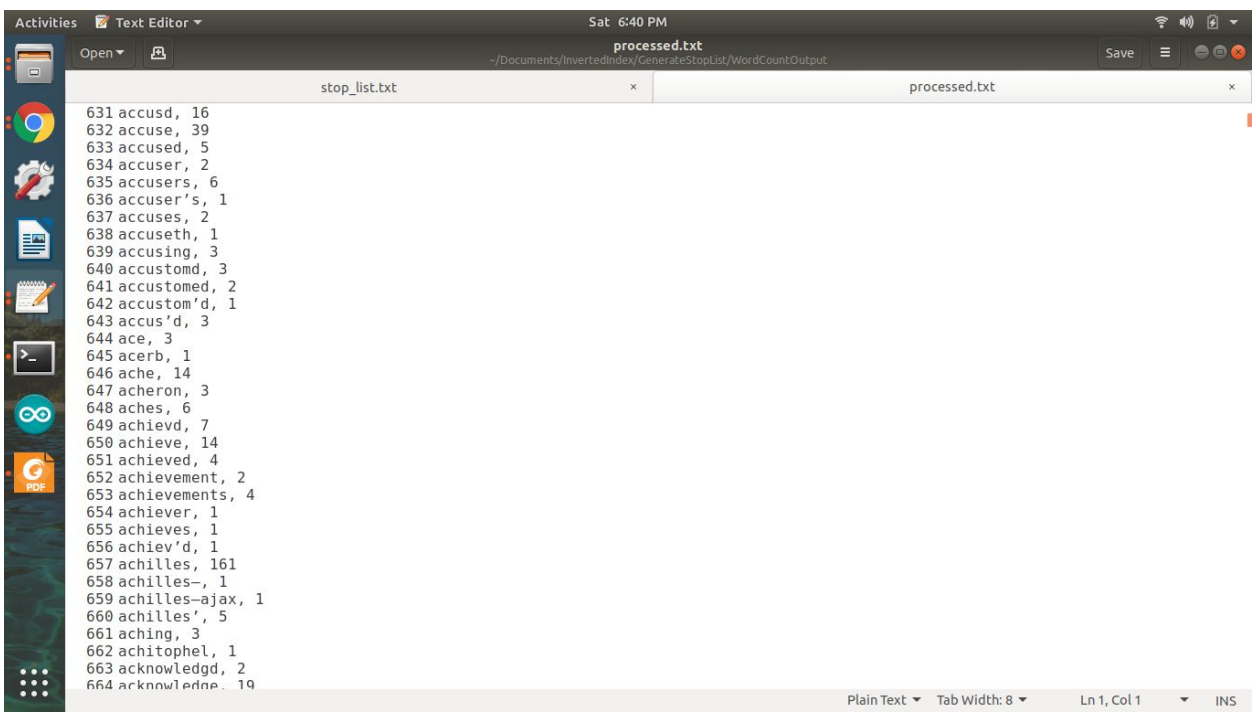


Figure 8: WordCount MapReduce Output

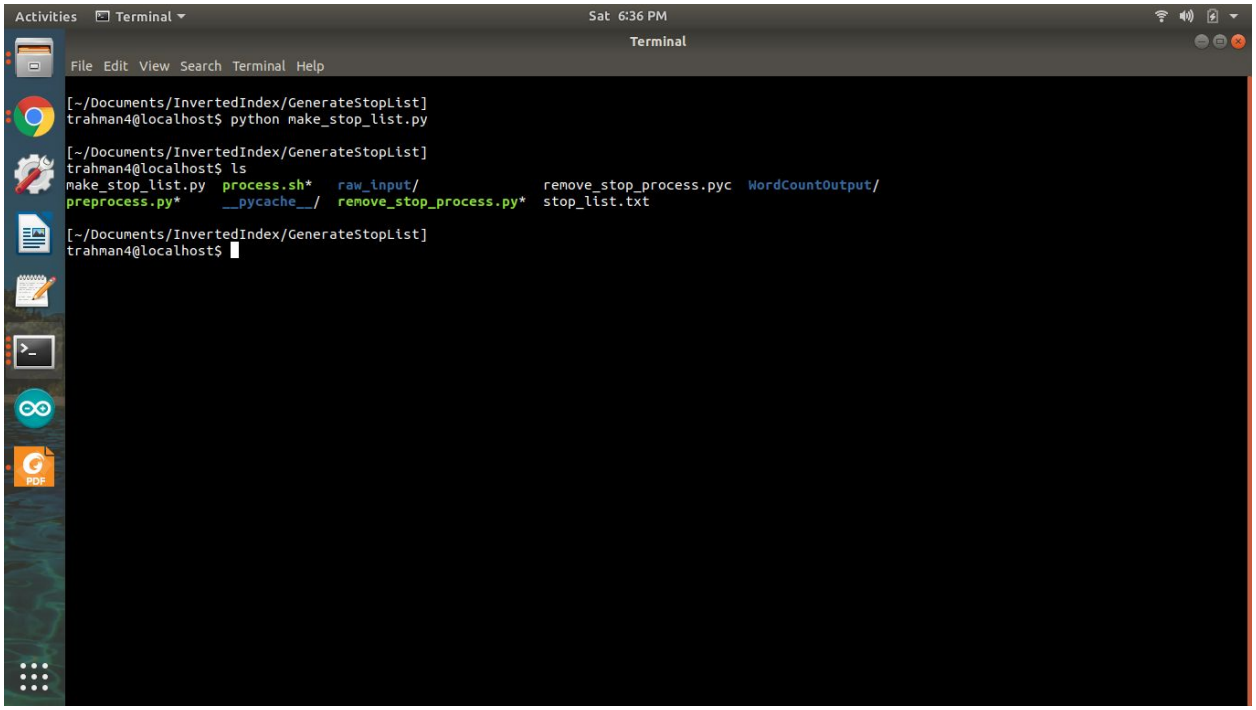


Figure 9: Run make_stop_list file to generate stop words

Run make_stop_ist.py. The output will be saved as stop_list.txt .

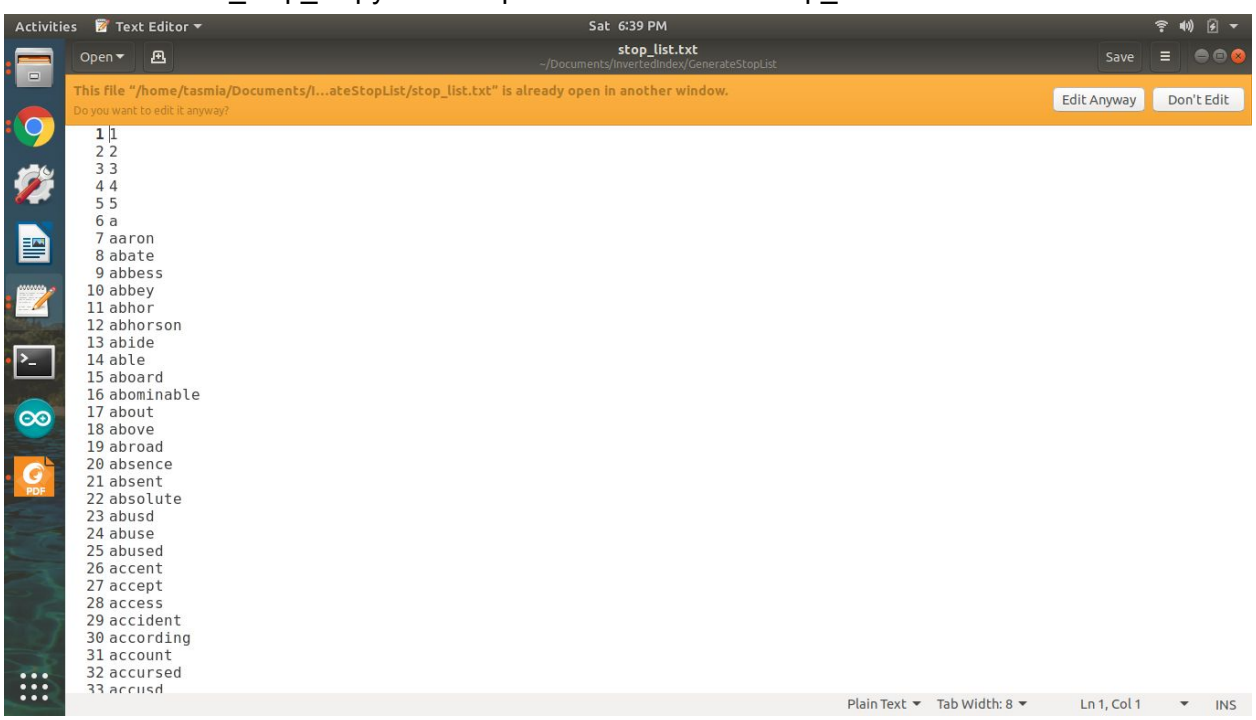


Figure 10: Generated Stop List

5.<u>Run Process.sh on WordCount :</u>

This step will remove all the stop words from raw input file. Now save the output as clean input under raw directory.

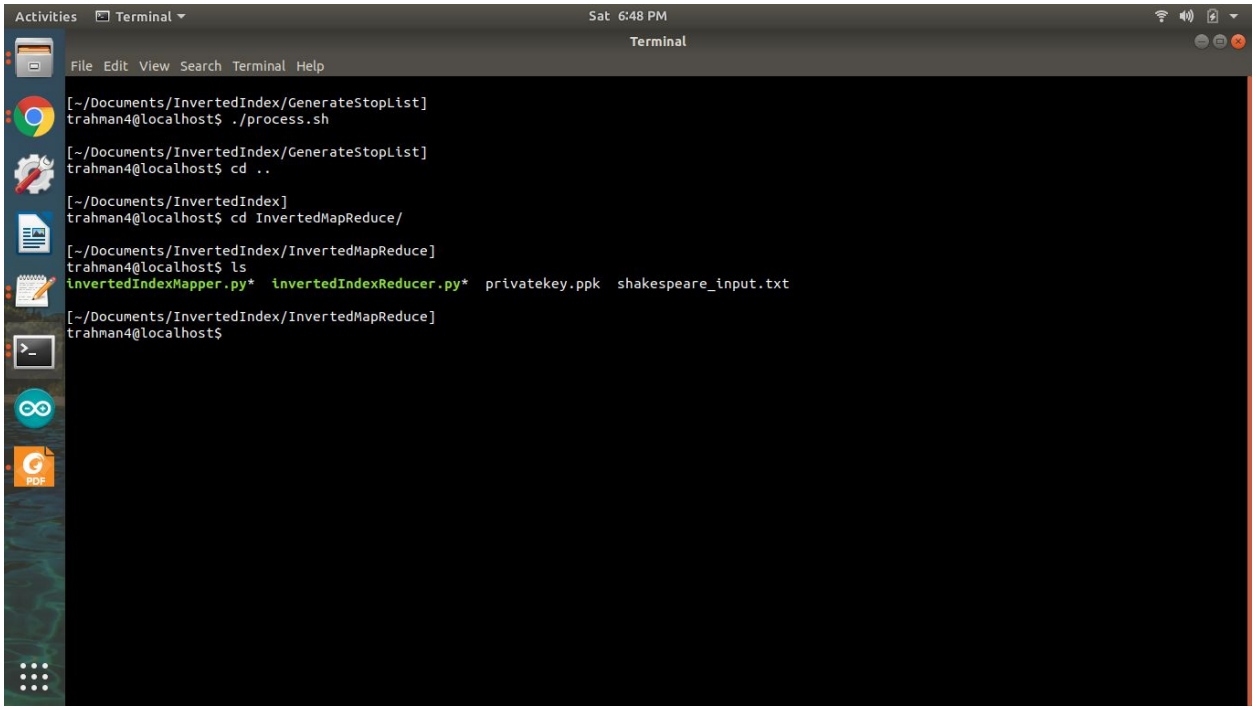


Figure 11: Run Process.sh again to remove the stop words

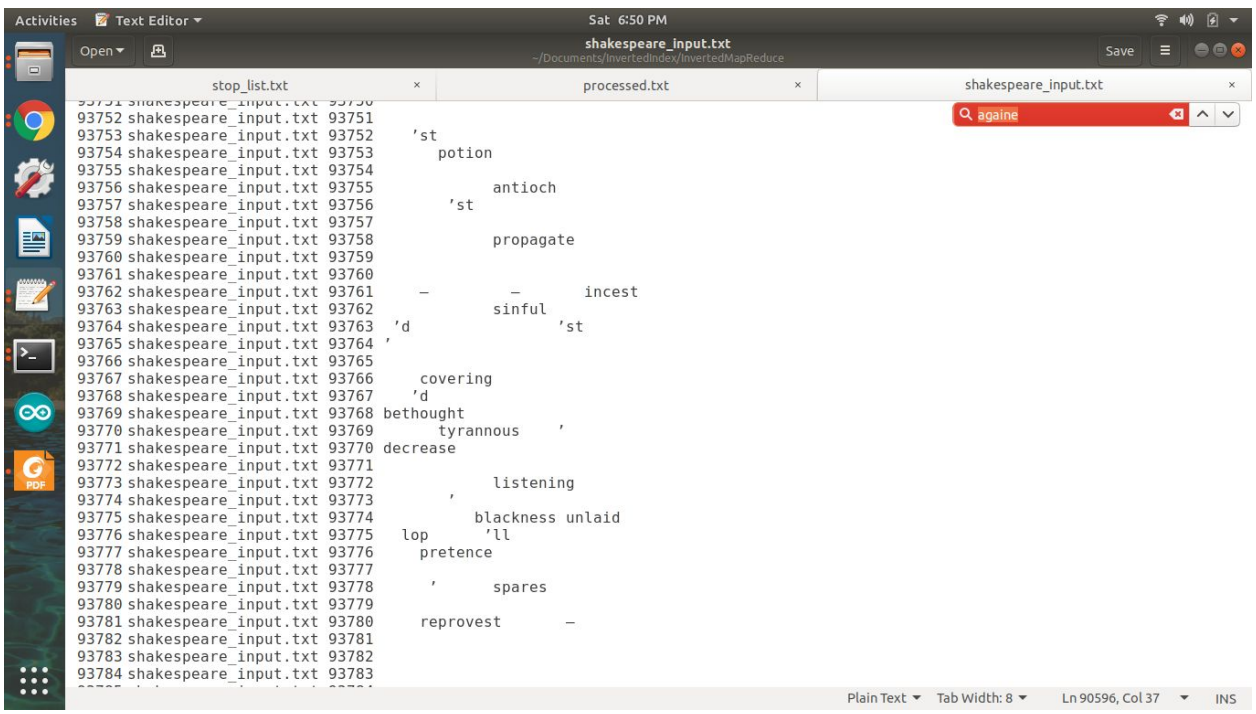


Figure 12: Proof that Stop Words are removed

Also, transfer the clean data to cloud.This clean file is our input for inverted Index MapReduce program.

```
Tasmia@namenode:~$ ls
clean_input.txt  invertedIndexMapper.py   invert_out.txt  processed.txt  shakespeare_input.txt  wordCountMapper.py
final_out.txt    invertedIndexReducer.py  privatekey.ppk  raw_input.txt  test.txt               wordCountReducer.py
Tasmia@namenode:~$ rm shakespeare_input.txt
Tasmia@namenode:~$ ls
clean_input.txt  invertedIndexMapper.py   invert_out.txt  processed.txt  test.txt            wordCountReducer.py
final_out.txt    invertedIndexReducer.py  privatekey.ppk  raw_input.txt  wordCountMapper.py
Tasmia@namenode:~$
```

Figure 13: Delete the previous shakespeare_input text file from cloud file system

```
Tasmia@namenode:~$ hadoop fs -rm /tmp/shakespeare_input.txt
18/03/31 16:54:03 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /tmp/shakespeare_input.txt
Tasmia@namenode:~$ ls
clean_input.txt  invertedIndexMapper.py   invert_out.txt  processed.txt  test.txt            wordCountReducer.py
final_out.txt    invertedIndexReducer.py  privatekey.ppk  raw_input.txt  wordCountMapper.py
Tasmia@namenode:~$ hadoop fs -ls /tmp/
Found 28 items
-rw-r--r--   3 Tasmia supergroup      13828 2018-03-29 23:29 /tmp/before_processing.txt
-rw-r--r--   3 Tasmia supergroup        148 2018-03-30 13:56 /tmp/clean_input.txt
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 15:27 /tmp/count_m_r
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 03:23 /tmp/count_out
-rw-r--r--   3 Tasmia supergroup         43 2018-03-30 13:52 /tmp/dirty_input.txt
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 16:04 /tmp/final_inverted_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 02:11 /tmp/final_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 16:29 /tmp/final_wordcount
drwxrwxrwt   - root    supergroup         0 2018-03-29 22:57 /tmp/hadoop-yarn
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 08:48 /tmp/invert_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 14:07 /tmp/invert_out_1
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:37 /tmp/map_clean_output
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:32 /tmp/map_dirty_input
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:38 /tmp/map_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-31 00:03 /tmp/map_red_out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:15 /tmp/out
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:18 /tmp/out1
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:20 /tmp/out2
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:24 /tmp/out3
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:26 /tmp/out4
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:29 /tmp/out5
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 10:03 /tmp/out6
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 10:12 /tmp/out7
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:54 /tmp/out8
drwxr-xr-x   - Tasmia supergroup          0 2018-03-30 13:57 /tmp/out9
drwxr-xr-x   - Tasmia supergroup          0 2018-03-29 23:13 /tmp/output
-rw-r--r--   3 Tasmia supergroup       2271 2018-03-31 02:10 /tmp/raw_input.txt
-rw-r--r--   3 Tasmia supergroup        161 2018-03-30 10:12 /tmp/sample_input.txt
Tasmia@namenode:~$
```

Figure 14: Delete the previous shakespeare_input text file from hdfs as well

Step 6-7: Building the Inverted Index:

6.Move invertedIndexMapReduce code to cloud

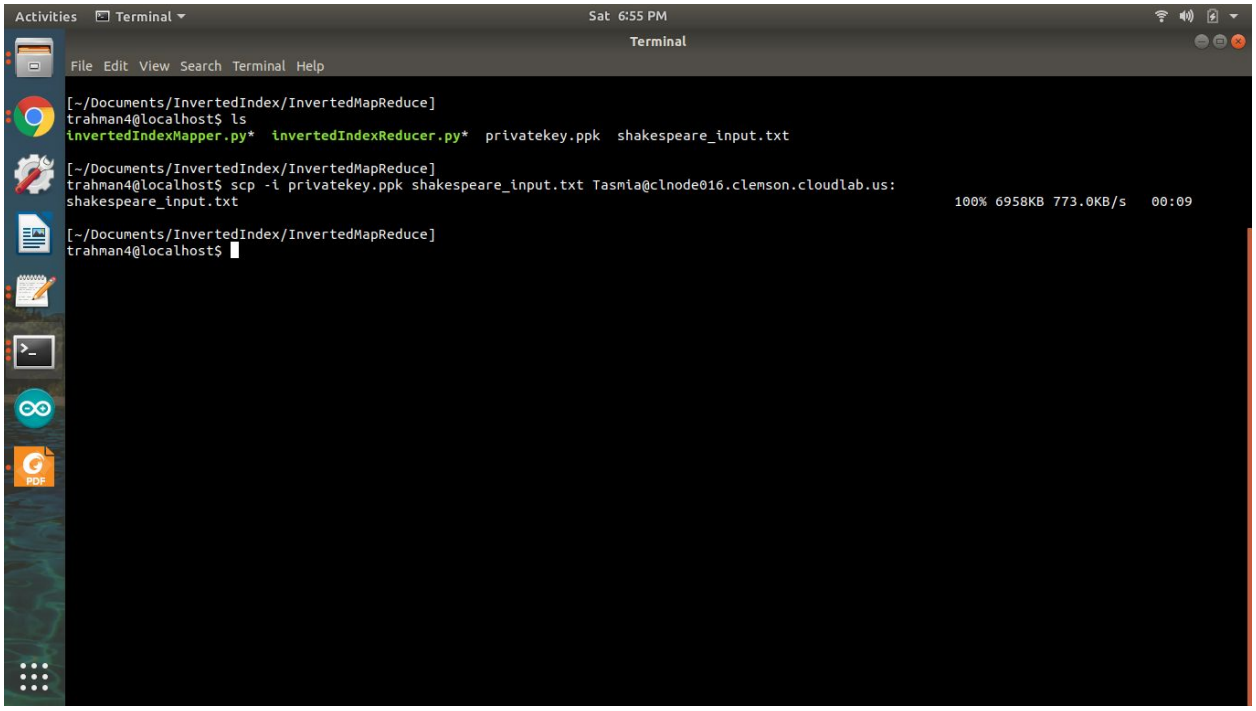


Figure 15: Copy the input file for InvertedIndexMapReduce to cloud file system

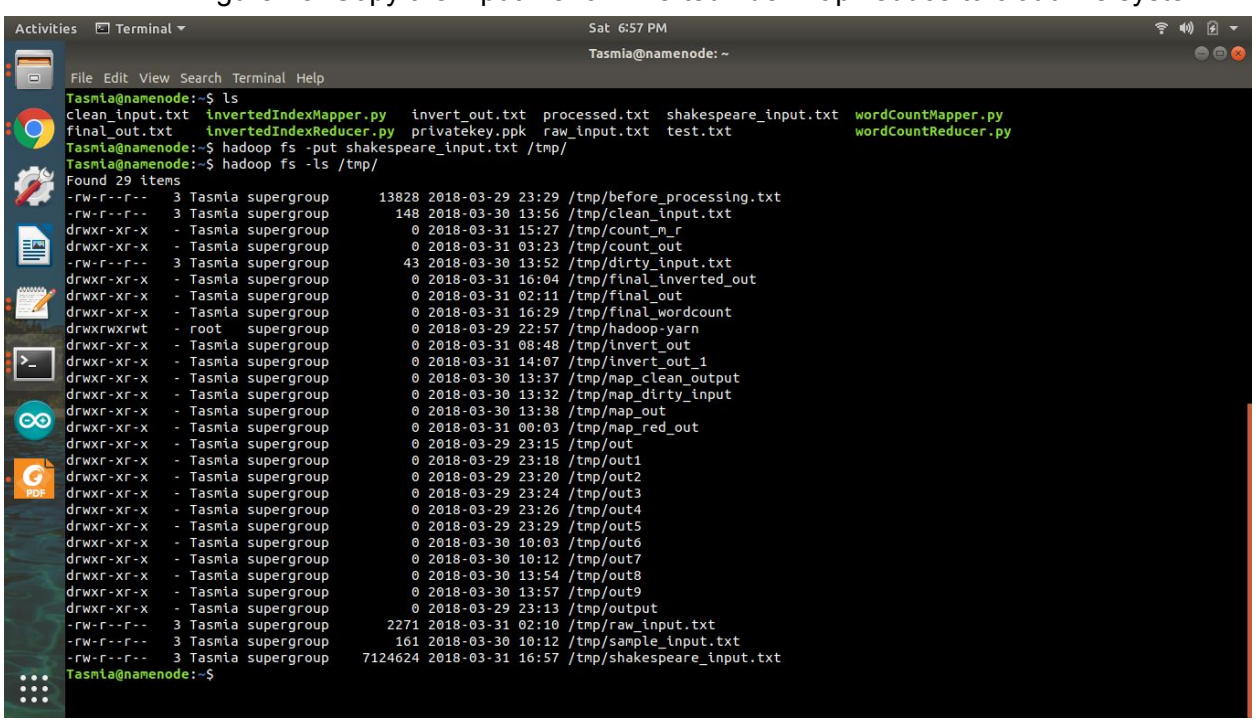
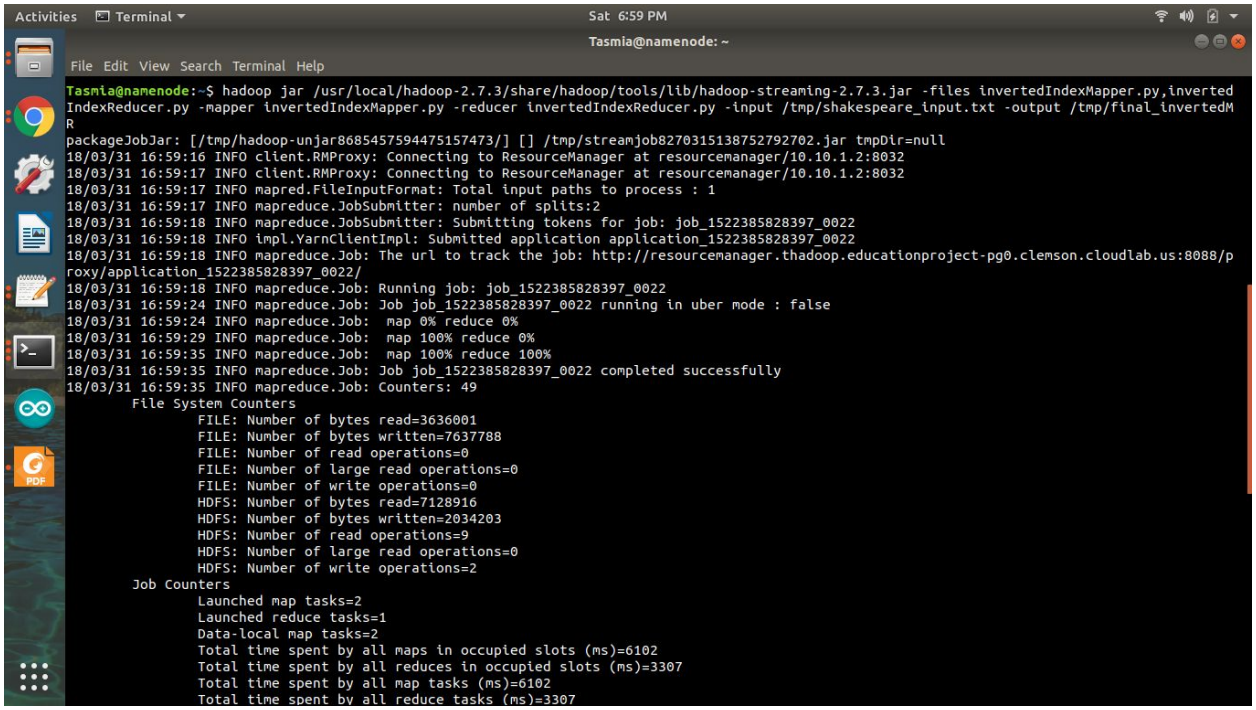


Figure 16. Copying the new shakespeare_input text file from cloud file system to hdfs

7 Run invertedIndexMapreduce



```
Tasmia@namenode:~$ hadoop jar /usr/local/hadoop-2.7.3/share/hadoop/tools/lib/hadoop-streaming-2.7.3.jar -files invertedIndexMapper.py,inverted
IndexReducer.py -mapper invertedIndexMapper.py -reducer invertedIndexReducer.py -input /tmp/shakespeare_input.txt -output /tmp/final_invertedM
R
packageJobJar: [/tmp/hadoop-unjar8685457594475157473/] [] /tmp/streamjob8270315138752792702.jar tmpDir=null
18/03/31 16:59:16 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
18/03/31 16:59:17 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
18/03/31 16:59:17 INFO mapred.FileInputFormat: Total input paths to process : 1
18/03/31 16:59:17 INFO mapreduce.JobSubmitter: number of splits:2
18/03/31 16:59:18 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1522385828397_0022
18/03/31 16:59:18 INFO impl.YarnClientImpl: Submitted application application_1522385828397_0022
18/03/31 16:59:18 INFO mapreduce.Job: The url to track the job: http://resourcemanager.thadoop.educationproject-pg0.clemson.cloudlab.us:8088/p
roxy/application_1522385828397_0022/
18/03/31 16:59:18 INFO mapreduce.Job: Running job: job_1522385828397_0022
18/03/31 16:59:24 INFO mapreduce.Job: Job job_1522385828397_0022 running in uber mode : false
18/03/31 16:59:24 INFO mapreduce.Job:  map 0% reduce 0%
18/03/31 16:59:29 INFO mapreduce.Job:  map 100% reduce 0%
18/03/31 16:59:35 INFO mapreduce.Job:  map 100% reduce 100%
18/03/31 16:59:35 INFO mapreduce.Job: Job job_1522385828397_0022 completed successfully
18/03/31 16:59:35 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=3636001
                FILE: Number of bytes written=7637788
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=7128916
                HDFS: Number of bytes written=2034203
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=6102
                Total time spent by all reduces in occupied slots (ms)=3307
                Total time spent by all map tasks (ms)=6102
                Total time spent by all reduce tasks (ms)=3307
```

Figure 17: InvertedIndex MapReduce 100%

Step 8-9:Query the Inverted Index:

8.Save the output of step 7 into local machine's query directory.

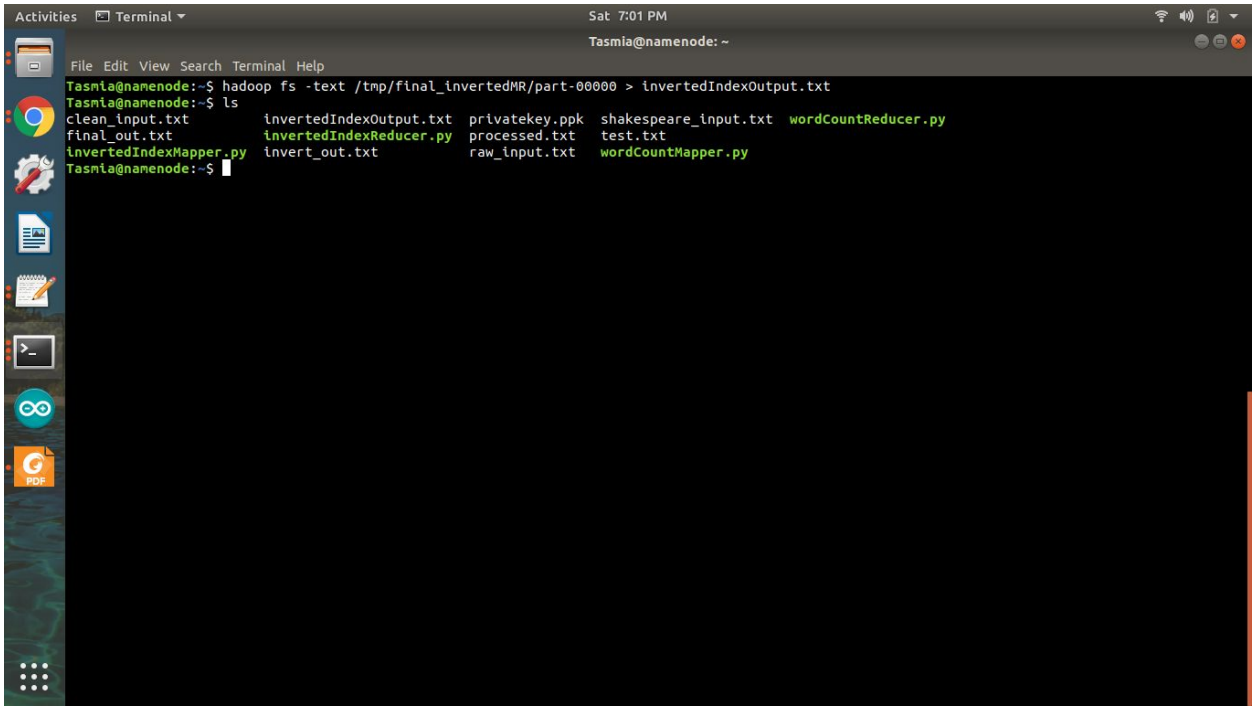


Figure 18: Copying the InvertedIndex MapReduce output to cloud file system

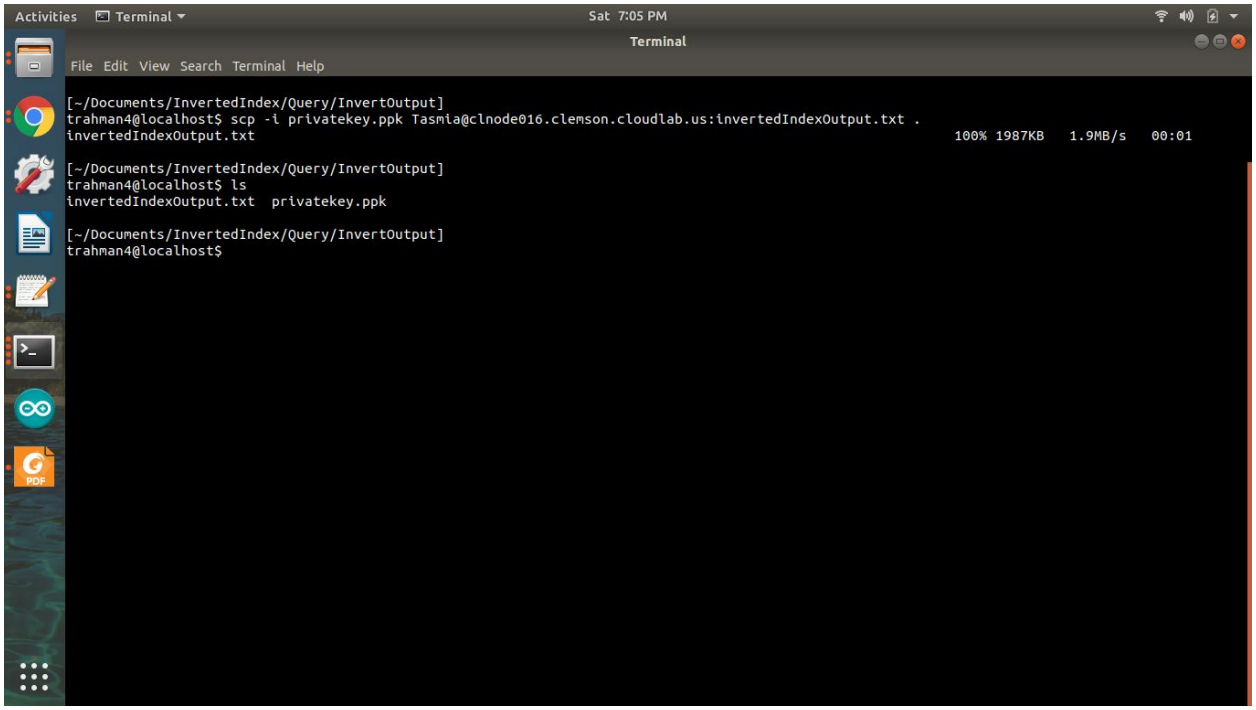


Figure 19:Copying the InvertedIndex output to query invertoutput directory so that we can make query

9.Run the query.py file to find out the location of a word .

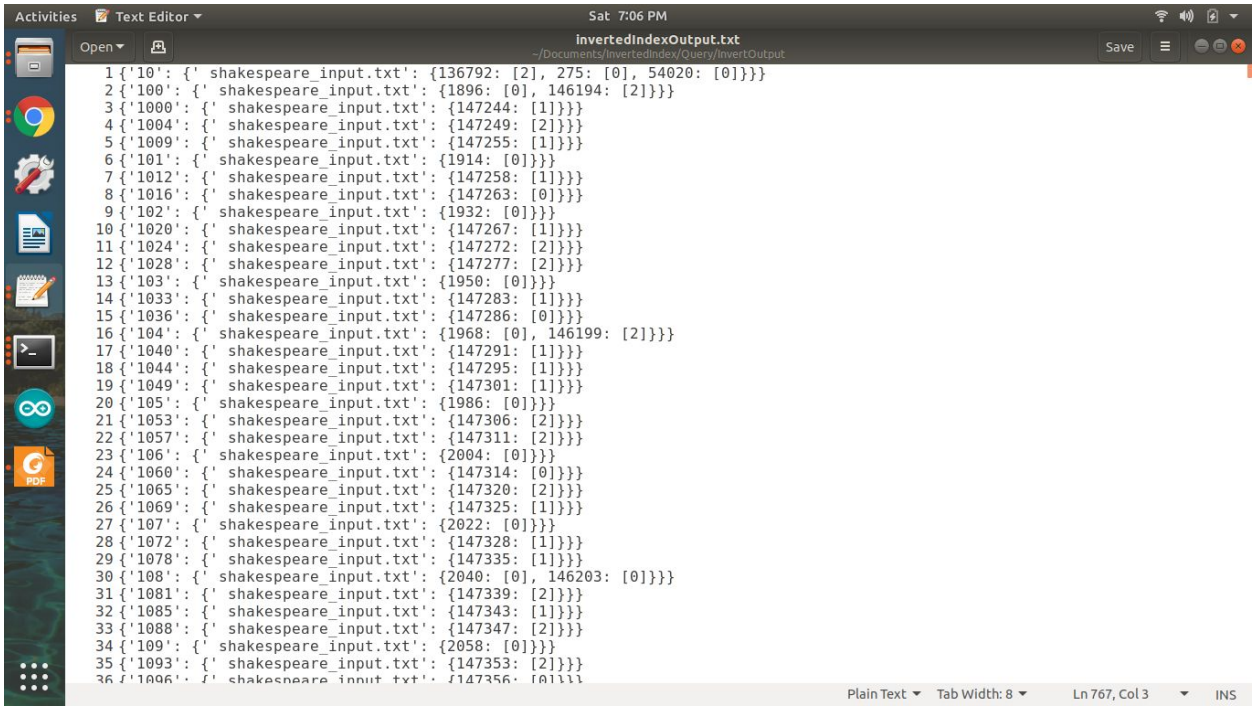


Figure 20. InvertedIndex MapReduce Output

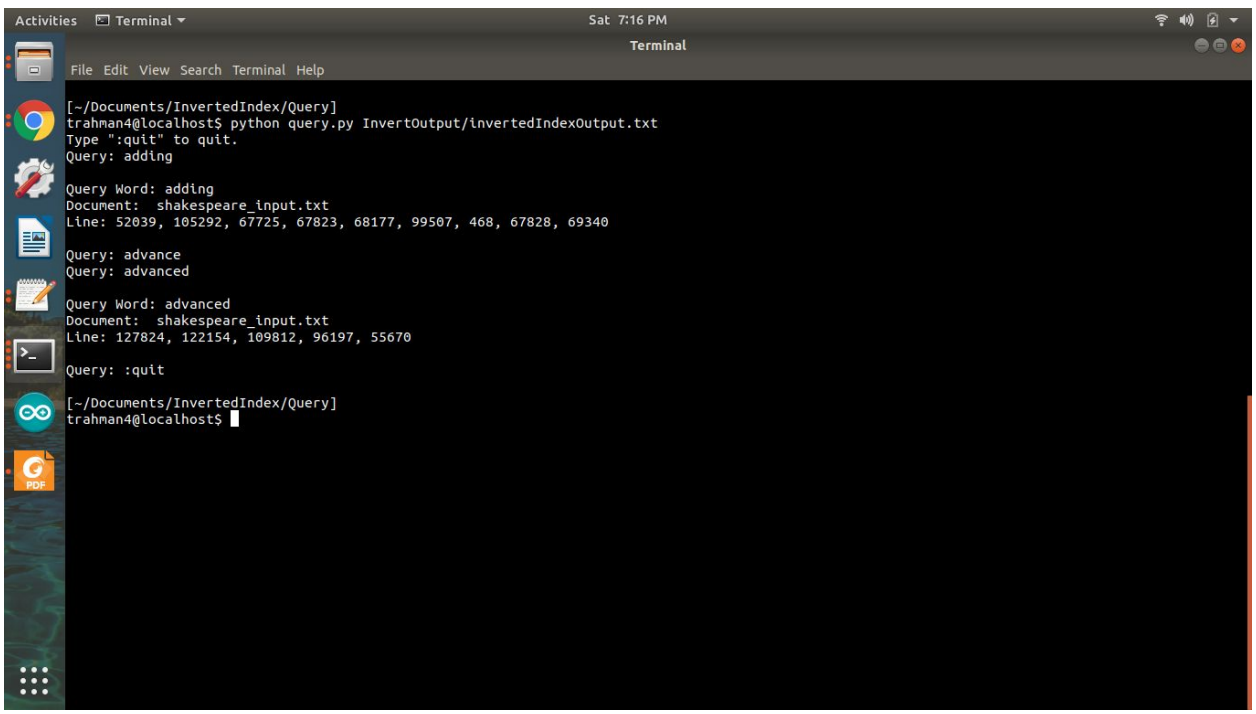


Figure 21: Different forms of Query

Directory Description:

- InvertedIndex
  - Preprocess
    - Raw_input
      - Clean
      - dirty
    - process.sh
  - GenerateStopList
    - Raw_input
      - Clean
      - dirty
    - WordCount_Output
      - processed.txt
    - process.sh
  - WordCountMapReduce_Stop
    - Processed_input
    - WordCountMapper.py
    - WordCountReducer.py
  - InvertedMapReduce
    - Shakespear_input.txt
    - invertedIndexMapper.py
    - invertedIndexReducer.py
  - Query
    - InvertedOutput
    - Query.py

Reference:

1. https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
2. https://stackoverflow.com/questions/28294352/mapreduce-inverted-index-program
3. https://github.com/kiran4399/inverted-index_hadoop
4. http://stdatalabs.blogspot.com/2017/03/mapreduce-vs-spark-inverted-index.html
5. https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm