LAB - 2

1. For experiment 1, I compiled and ran the source code WordCount.java provided to us. I followed the instructions specified in Lab 1.
    - I first copied the shakespeare data from the local file system to the hadoop system. Attaching screenshot below.

```
cpre419@cpre419-VirtualBox:~/hadoop/sbin$ hadoop fs -copyFromLocal /home/cpre419/Downloads/shakespeare /data
2021-02-27 23:04:54,220 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
cpre419@cpre419-VirtualBox:~/hadoop/sbin$ hadoop fs -ls /data
Found 1 items
-rw-r--r--   1 cpre419 supergroup    4538523 2021-02-27 23:04 /data/shakespeare
cpre419@cpre419-VirtualBox:~/hadoop/sbin$
```

    - I then created an empty directory for lab 2 and made sure to specify an empty path when running the jar file. Attaching screenshot of running the wordcount.jar

```
cpre419@cpre419-VirtualBox:~/hadoop/sbin$ hadoop  jar  ~/Downloads/wordcount.jar  WordCount  /data/shakespeare  /lab2/output
2021-02-27 23:15:57,637 INFO impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2021-02-27 23:15:57,915 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2021-02-27 23:15:57,915 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2021-02-27 23:15:58,571 INFO input.FileInputFormat: Total input files to process : 1
2021-02-27 23:15:58,756 INFO mapreduce.JobSubmitter: number of splits:1
2021-02-27 23:15:59,251 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local921485047_0001
2021-02-27 23:15:59,267 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-02-27 23:15:59,573 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2021-02-27 23:15:59,574 INFO mapreduce.Job: Running job: job_local921485047_0001
2021-02-27 23:15:59,582 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2021-02-27 23:15:59,616 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-02-27 23:15:59,622 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2021-02-27 23:15:59,623 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2021-02-27 23:15:59,820 INFO mapred.LocalJobRunner: Waiting for map tasks
2021-02-27 23:15:59,821 INFO mapred.LocalJobRunner: Starting task: attempt_local921485047_0001_m_000000_0
2021-02-27 23:15:59,921 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-02-27 23:15:59,924 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2021-02-27 23:16:00,004 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2021-02-27 23:16:00,022 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/data/shakespeare:0+4538523
2021-02-27 23:16:00,347 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2021-02-27 23:16:00,347 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2021-02-27 23:16:00,347 INFO mapred.MapTask: soft limit at 83886080
2021-02-27 23:16:00,347 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2021-02-27 23:16:00,347 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2021-02-27 23:16:00,374 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2021-02-27 23:16:00,497 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-02-27 23:16:00,600 INFO mapreduce.Job: Job job_local921485047_0001 running in uber mode : false
2021-02-27 23:16:00,600 INFO mapreduce.Job:  map 0% reduce 0%
```

    file below.

    - I have attached the workspace folder, java file and outputs for Experiment 1 in my submission. Attaching a screenshot of the output below:

part-r-00000 - Notepad

File Edit Format View Help

```
!        10526
!'By     1
!'twas   1
!,       1
!As      1
!Ay      1
!Come    1
!Give't 1
!Handkerchief    1
!Hear    1
!Hoo     1
!How     1
!Kill    1
!Nay     1
!Out     1
!Please 1
!Remain 1
```

2. Attaching the Driver.java code I wrote and the temp and final output files generated. Also attaching code with comments, workspace directory and screenshots below for reference.

   Screenshot of running the code.

```
cpre419@cpre419-VirtualBox:~/hadoop/sbin$ hadoop jar ~/Downloads/bigrams.jar Driver
2021-02-28 19:46:10,151 INFO impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2021-02-28 19:46:10,458 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2021-02-28 19:46:10,458 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2021-02-28 19:46:10,993 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-02-28 19:46:11,271 INFO input.FileInputFormat: Total input files to process : 1
2021-02-28 19:46:11,514 INFO mapreduce.JobSubmitter: number of splits:1
2021-02-28 19:46:11,937 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local628943490_0001
2021-02-28 19:46:11,939 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-02-28 19:46:12,272 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2021-02-28 19:46:12,277 INFO mapreduce.Job: Running job: job_local628943490_0001
2021-02-28 19:46:12,290 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2021-02-28 19:46:12,296 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-02-28 19:46:12,296 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2021-02-28 19:46:12,315 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2021-02-28 19:46:12,516 INFO mapred.LocalJobRunner: Waiting for map tasks
2021-02-28 19:46:12,517 INFO mapred.LocalJobRunner: Starting task: attempt_local628943490_0001_m_000000_0
2021-02-28 19:46:12,614 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2021-02-28 19:46:12,618 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2021-02-28 19:46:12,741 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2021-02-28 19:46:12,753 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/data/shakespeare:0+4538523
2021-02-28 19:46:13,120 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2021-02-28 19:46:13,122 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2021-02-28 19:46:13,122 INFO mapred.MapTask: soft limit at 83886080
2021-02-28 19:46:13,122 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2021-02-28 19:46:13,122 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2021-02-28 19:46:13,132 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
```

   Screenshots of the temp and output.

```
a b     2
a babbled       1
a babe  6
a baboons       1
a baby  4
a bachelor      15
a bade  1
a bag   2
a baggage       1
```

part-r-00001 - Notepad

File Edit Format View Help

```
i am    1855
my lord 1685
i have  1617
in the  1579
i will  1572
to the  1513
of the  1375
it is   1080
to be   968
that i  904
```

Q.
Think about how you might be able to get around the fact that bigrams might span lines of input. Briefly describe how you might deal with that situation?

From what I understood, the TextInputFormat.class is responsible for handling the text input, computing the input splits and deciding the logic based on which the splits are computed. By default, the logic is for this is to split the text based on the occurrence of new lines.
Since we want to find bigrams that span multiple lines of the input, one idea I have is to write a custom InputFormat class such that the logic based on how the splits are computed uses the "." as a delimiter.
Writing a custom InputFormat class is something I found on this link.
http://hadoop.apache.org/docs/stable/api/org/apache/hadoop/mapreduce/InputFormat.html

If this is not possible, another way I can think of to solve this problem, is to pass the previous word as one of the fields in the value list sent to the reducer. Some processing on this might help the reducer account for bigrams spanning two different lines.