CSE 587 Assignment 1
Rajkumar Rathlavath(50496190)

1. **What are the 25 most common words and the number of occurrences of each when you do not remove stop words?**

   The following is the snippet of top 25 words with most occurrence where the stop words are the, to were, with, with, my, but, at etc.

```
(base) rajkumarrathlavath@Rajkumars-Air hadoop_files % hadoop fs -cat /output3/part-r-00000 | sort -k2,2nr | head -n 25
2024-03-25 17:41:06,583 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
the     79970
and     52078
to      43257
of      39773
a       29990
i       24113
in      23745
that    20551
he      19682
his     18356
you     15361
with    14955
is      13284
my      13243
was     12991
not     12900
for     11843
it      11791
as      11044
but     10208
be      9622
at      9218
have    9214
had     9186
this    8251
(base) rajkumarrathlavath@Rajkumars-Air hadoop_files %
```

   To achieve this, a mapper and a reducer are employed. The mapper processes the input data, extracting text from files, removing unwanted characters, and assigning a value of 1 to each word, creating intermediate key/value pairs. The reducer then aggregates these intermediate values, incrementing the value for each key by 1, thereby counting the frequency of each word. The output is then sorted using the command line sort to extract the 25 most frequent words.

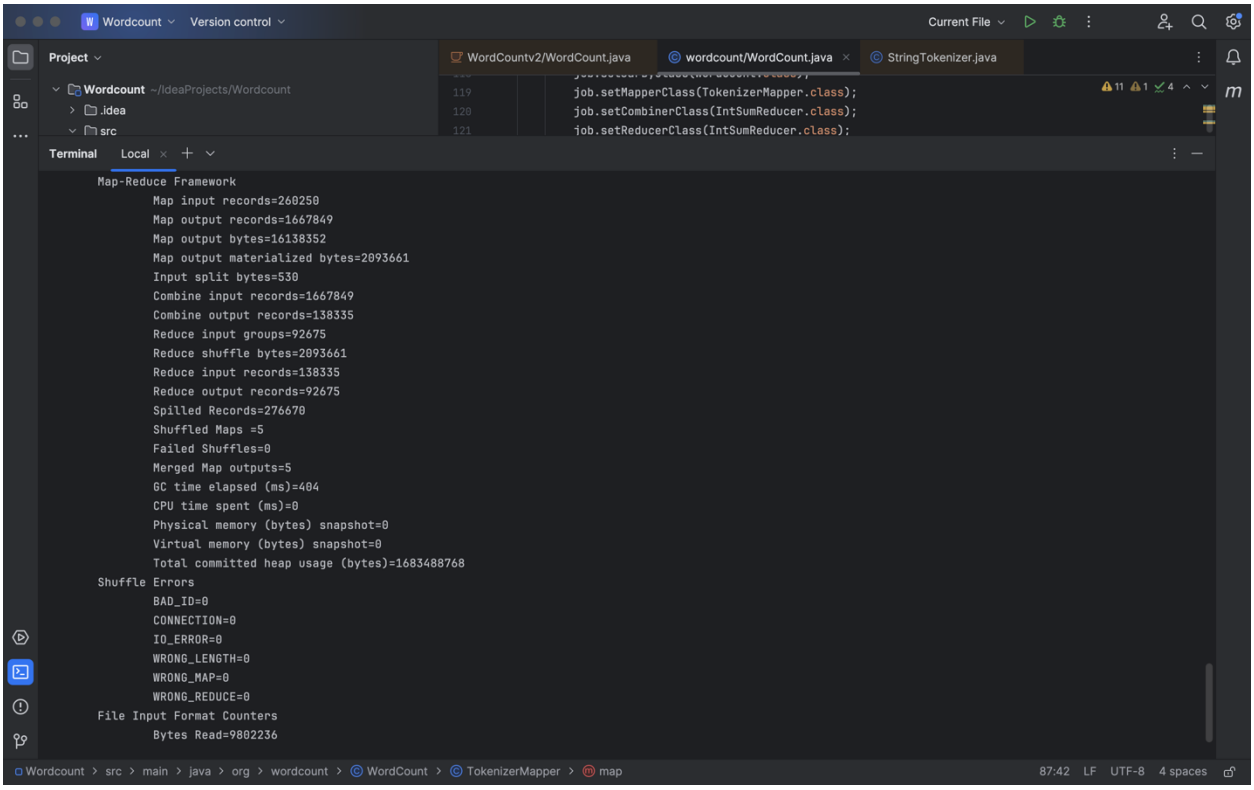
2. **What are the 25 most common words and the number of occurrences of each when you do remove stopwords?**

```
(base) rajkumarrathlavath@Rajkumars-Air ~ % hadoop fs -cat /output6/part-r-00000 | sort -k2,2nr | head -n 25
2024-03-25 18:04:01,887 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
i       24113
his     18356
you     15361
not     12900
have    9214
had     9186
your    7623
by      7246
her     7025
him     6693
all     6173
will    5943
what    5827
which   5785
so      5724
me      5540
they    5520
she     5410
said    5128
we      4992
if      4987
or      4435
who     4421
do      4375
one     4322
(base) rajkumarrathlavath@Rajkumars-Air ~ %
```

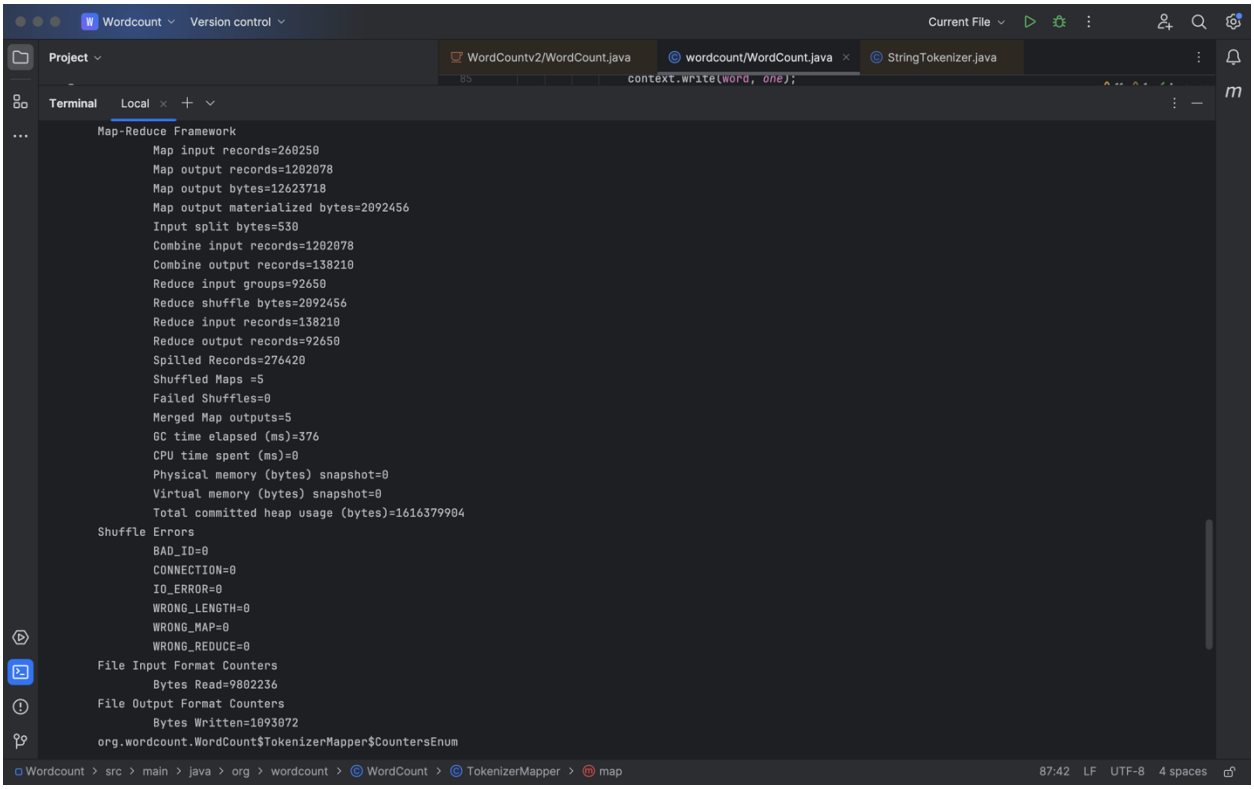
By adding a feature or a special functionality to the mapper code to exclude stop words from the output. I created a special text file (stopwords.txt) containing words that I want to skip. The mapper now checks each word against this list and skips it if there's a match. This way, I don't have to count the stop words in the final output. Then, after the reducer does its thing, I use command line sorting to sort the data by the values (instead of the keys). It's like organizing a big party and making sure the most popular guests get the attention they deserve.

3. **Based on the output of your application, how does removing stop words affect the total amount of bytes output by your mappers? Name one concrete way that this would affect the performance of your application**.

   The following is the output results with stop words included:



The following is the output results with stop words not included.

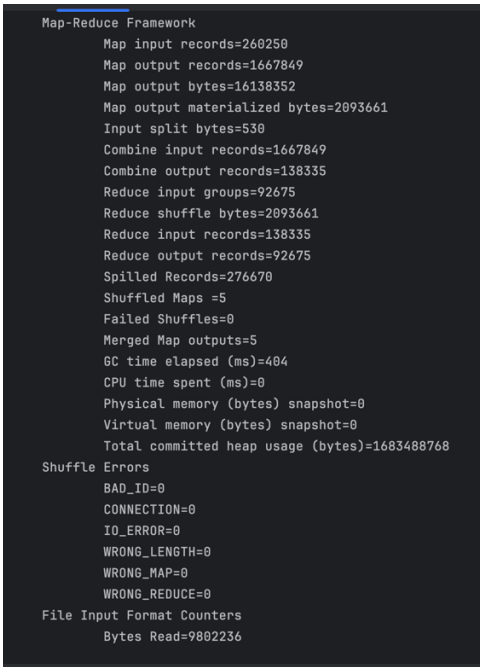


From the above both output observations we can observe that the mapper output size is significantly decreased after the stop words removal. Initially the size of the mapper output is 16138352 and after removal of the stop words it shrinks to 12623718. Thus, we can conclude that the removal of the stop keys from the mapper output the memory usage is reduced significantly. Reducing intermediate data will enhance performance by diminishing the necessity for file I/O and decreasing the volume of data transmitted via the network.

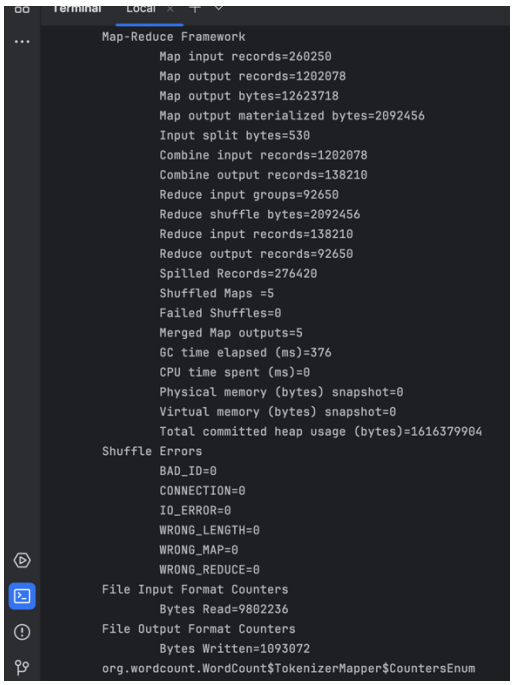
4. **Based on the output of your application, what is the size of your keyspace with and without removing stopwords? How does this correspond to the number of stopwords you have chosen to remove?**

The key space represents the total number of unique keys that can be generated from the data processor. In the case of the mapper-reducer configuration, this is reflected in the 'Reduce input/output groups' metric, which provides a summary of the types of keys generated.

The following is the comparison of both outputs with and without stop words, first let's observe that and conclude.



```
Map-Reduce Framework
        Map input records=260250
        Map output records=1667849
        Map output bytes=16138352
        Map output materialized bytes=2093661
        Input split bytes=530
        Combine input records=1667849
        Combine output records=138335
        Reduce input groups=92675
        Reduce shuffle bytes=2093661
        Reduce input records=138335
        Reduce output records=92675
        Spilled Records=276670
        Shuffled Maps =5
        Failed Shuffles=0
        Merged Map outputs=5
        GC time elapsed (ms)=404
        CPU time spent (ms)=0
        Physical memory (bytes) snapshot=0
        Virtual memory (bytes) snapshot=0
        Total committed heap usage (bytes)=1683488768
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
File Input Format Counters
        Bytes Read=9802236
```



```
Terminal  Local
Map-Reduce Framework
        Map input records=260250
        Map output records=1202078
        Map output bytes=12623718
        Map output materialized bytes=2092456
        Input split bytes=530
        Combine input records=1202078
        Combine output records=138210
        Reduce input groups=92650
        Reduce shuffle bytes=2092456
        Reduce input records=138210
        Reduce output records=92650
        Spilled Records=276420
        Shuffled Maps =5
        Failed Shuffles=0
        Merged Map outputs=5
        GC time elapsed (ms)=376
        CPU time spent (ms)=0
        Physical memory (bytes) snapshot=0
        Virtual memory (bytes) snapshot=0
        Total committed heap usage (bytes)=1616379904
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
File Input Format Counters
        Bytes Read=9802236
File Output Format Counters
        Bytes Written=1093072
org.wordcount.WordCount$TokenizerMapper$CountersEnum
```

By observing both the outputs of the map reducer we can conclude that there is a significant decrease in the reducer output record. When we include the stop words, we obtain 92675 records and when we do not include stop words, we obtain 92650 records. By this we can conclude that we have encountered with 25 stop words where we have 92650 unique records in text file. Thus, if we ignore stop words there is a decrease in key space.

5. **Let's now assume you were going to run your application on the entirety of Project Gutenberg. For this question, assume that there are 100TB of input data, the data is spread over 10 sites, and each site has 20 mappers. Assume you ignore all but the 25 most common words that you listed in question 2. Furthermore, assume that your combiners have been run optimally so that each combiner will output at most 1 keyvalue pair per key.**

(a) How much data will each mapper have to parse?

The overall dataset consists of 100 TB, distributed across 10 sites, with each site holding 10 TB (100/10TB, data at each site). Twenty mappers are assigned to each site, resulting in each mapper processing 500 GB (10 TB/20 mappers).

(b) What is the size of your keyspace?

As we are assuming to consider only 25 common words it equals to the size of Key space and hence the key size is 25.

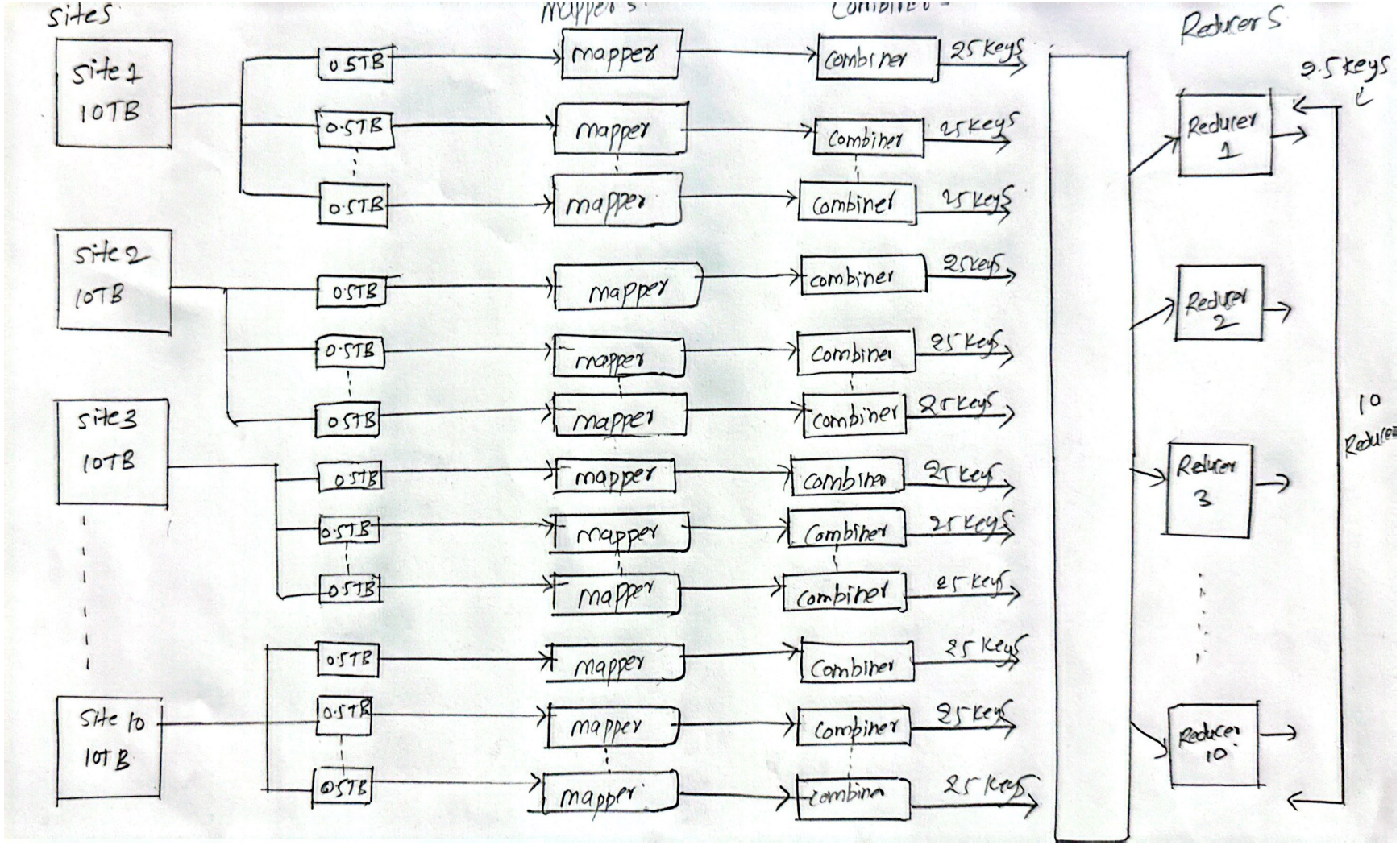
(c) What is the maximum number of key-value pairs that could be communicated during the barrier between mapping and reducing?

We have 10 sites, each with 20 mappers, totaling 200 mappers. The mapper reducer architecture shows that each mapper has a combiner, which feeds into a barrier. Assuming 25 key-value pairs per mapper, we get 25 x 200 = 5000 key-value pairs from all combiners, which are then passed through the barrier. This means the barrier can handle up to 5000 key-value pairs.

(d) Assume you are running one reducer per site. On average, how many key-value pairs will each reducer have to handle?

With 10 reducers and 5000 key-value pairs from the barrier, each reducer handles approximately 500 key-value pairs on average (5000 divided by 10). However, considering each team generates 2.5 keys worth of data (25 teams), each reducer will process about 500 key-value pairs worth of data, averaging 2.5 keys per team.

6. Draw the data flow diagram for question 5. The diagram should be similar to the diagram shown in the lecture. On your diagram, label the specific quantities you got for 5a,b,c, and d.


Sites
Site 1
10TB
Site 2
10TB
Site 3
10TB
Site 10
10TB
0.5TB
mapper
combiner
25 keys
Reducers
Reducer 1
Reducer 2
Reducer 3
Reducer 10
2.5 keys
10

Reference:

[1] https://medium.com/@MinatoNamikaze02/installing-hadoop-on-macos-m1-m2-2023-d963abeab38e

[2] https://www.oracle.com/java/technologies/downloads/