Code:

def mapper(self, \_, line):

words = WORD\_REGEX.findall(line)

for word in words:

yield (word.lower(), 1)

def reducer(self, word, counts):

yield(word,sum(counts))

Run:

bash-4.2$ python wordcount.py sherlock.txt > out.txt

The file out.txt contains the output of your MapReduce job. Open the results file, and answer the following question.

1. How many times does the word Sherlock appear in the file?

345 times

When you open the out.txt file now you should see substantially less words, and all of them should have at least 10 values if the code was implemented correctly.

def reducer(self, word, counts):

x = sum(counts)

if (x >=10):

yield word,x

Using Hadoop : bash-4.2$ python wordcount.py -r hadoop sherlock.txt > out.txt

2.How many words appear 10 times or more in the document?

3964 words appeared more than 10 times.

3.How many words in total (counting duplicates) has the document?

def mapper(self, \_, line):

words = WORD\_REGEX.findall(line)

for word in words:

a="Total"

yield (a, 1)

#and the reducer method goes after this line

def reducer(self, a, counts):

yield(a,sum(counts))

"Total" 549644

4. To find out the longest word from the text

def reducer(self, word, counts):

#yield(a,sum(counts))

x = len (word)

if (x >=15):

yield word,x

#yield word, sum(counts)