

Big Data Analytics CPSC 5330

Lab 01

Fariha Shah

April 12, 2025

1 How much time you spent on the assignment

1. Environment Setup

To begin the lab, I followed the instructions provided on Canvas and the lab document. I dedicated approximately 3–4 hours to setting up the environment. This included launching an EC2 instance using the publicly available AMI named **CPSC_5330_S25**, generating a key-value pair to create the necessary .pem file, and establishing a secure connection between my local machine and the EC2 instance. Additionally, I ensured proper integration between the Ubuntu environment and HDFS for distributed file system operations.

2. Lab Demo Video Guidance

I invested around 4–5 hours watching and following the lab demo video. It was particularly helpful in understanding how to copy files from the local machine to the EC2 instance, create appropriate directories in HDFS, and compile and run MapReduce jobs effectively. This video served as a vital resource in bridging the gap between theoretical understanding and hands-on application.

3. Documentation and Lab Implementation

To implement the assignment instructions accurately, I spent about 2–3 hours each day thoroughly reading the lab documentation. As someone with limited prior experience in shell scripting using the Nano editor, this part of the lab posed a learning curve. Writing and debugging shell scripts to compile, run, and manage MapReduce jobs required additional time and attention, but ultimately strengthened my scripting skills.

2 If parts of the assignments are not fully working, which parts and what the problem(s) are?

All components of the assignment are functioning successfully on my end. This includes:

- **Part 1: word-count-shakespeare**
- **Part 2: word-count-shakespeare-clean and wordcountjava**
- **Final Output:** The generated Vocabulary.pdf containing the analysis and visual content

Each script executed as expected, and the outputs were validated and compiled into the final document. There were no unresolved issues or failures during execution.

3 Were there aspects of the assignment that were particularly challenging? Particularly confusing?

Yes, several parts of the assignment were challenging. Setting up and managing file paths between local and HDFS required careful attention, especially when dealing with wildcards and directory

structures in shell scripts. Another tricky part was debugging shell scripts and ensuring the correct permissions for execution. Writing and testing custom MapReduce logic for vocabulary richness and longest words also took effort, particularly in cleaning the text properly and formatting outputs. Additionally, merging and formatting results into a final PDF with code, output, and images presented some technical hurdles, but overall helped me improve my tool integration skills.

4 What were the main learning take-aways from this lab – that is, did it introduce particular concepts or techniques that might help you as an analyst or engineer in the future?

This lab provided hands-on experience with Hadoop's MapReduce framework, deepening my understanding of distributed computing, parallel data processing, and big data pipelines. Here are my key takeaways:

1. Understanding of the MapReduce Programming Model

I learned how to write custom Mapper and Reducer classes to process large-scale text data. This lab helped me internalize the separation of concerns in MapReduce: mapping (processing and emitting key-value pairs) and reducing (aggregating values by key), which is a powerful paradigm for scalable data processing.

2. Working with Real-World Text Data

Using texts from Shakespeare, Jane Austen, and the King James Bible, I explored challenges related to natural language data, such as:

- Tokenizing and cleaning words
- Counting frequencies and unique terms
- Calculating vocabulary richness
- Identifying the longest words in large corpora

This experience reinforced the importance of data cleaning and normalization in NLP-related tasks.

3. Shell Scripting & Hadoop Commands

I practiced automating MapReduce job submission, compilation, and result merging via shell scripts. This included:

- Using HDFS commands (`hdfs dfs -put`, `-mkdir`, `-getmerge`)
- Packaging and running MapReduce jobs
- Extracting and redirecting output for further analysis This exposure to command-line tools and scripting is crucial for working in production environments.

4. Working with Multi-Step Pipelines

In later parts of the lab, I saw how different scripts and processes come together:

- Running MapReduce for word counts
- Merging and parsing results in Python
- Calculating statistical measures (like vocabulary richness)
- Visualizing output using PDF tools It taught me how to think modularly and chain different tools together to form complete data workflows.

5. Big Data Mindset

Perhaps most importantly, this lab emphasized a shift in mindset: instead of bringing data to the code, I learned to bring code to the data. Processing at scale requires efficiency, distribution, and often simplicity—qualities inherent in MapReduce design

5 Output Interpretation

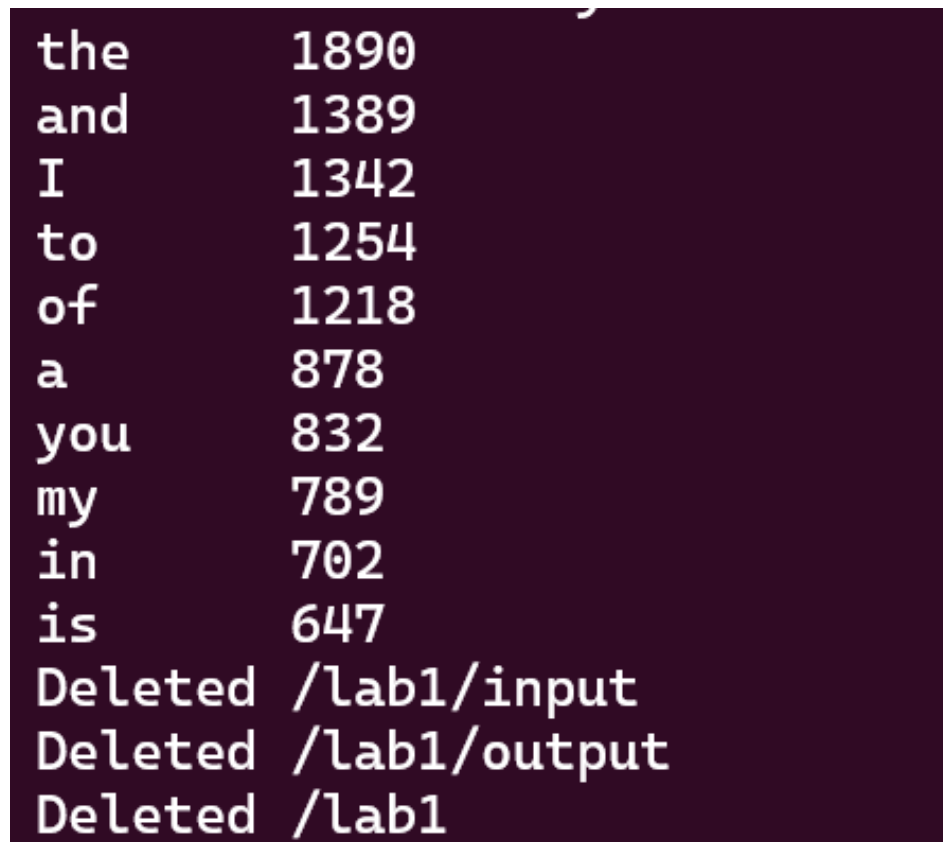
For reference, the following screenshots illustrate the output generated at each stage of the lab assignment. These visual results reflect the successful execution and processing of the MapReduce jobs for all three parts:

- **Part 01: Word Count (word-count-shakespeare)**

The output image displays the word count results for the `shakespeare.txt` file. It shows the total number of occurrences for each word after applying text preprocessing techniques such as:

- Lowercasing all words
- Removing all punctuation

This ensures accurate and standardized word frequency analysis.



```
the      1890
and      1389
I        1342
to       1254
of       1218
a        878
you      832
my       789
in       702
is       647
Deleted /lab1/input
Deleted /lab1/output
Deleted /lab1
```

- **Part 02: Cleaned Word Count (word-count-shakespeare-clean)**

The output image displays the top 15 most frequent words from the `shakespeare.txt` corpus. Before counting, the text was cleaned by:

- Removing all punctuation
- Converting all words to lowercase

This preprocessing ensures consistency in word counting by treating different forms of the same word (e.g., "The" and "the") as one. The result offers insights into the most frequently used words in Shakespeare's works after normalization.

Top 15 most frequent cleaned words:	
the	2219
and	2034
to	1512
i	1408
of	1302
you	1115
a	994
my	914
that	873
in	808
it	774
is	745
not	721
his	588
with	556

- **Part 03: Richness of Vocabulary**

The output image generated in Part 3 visually represents the richness of vocabulary for the three analyzed corpora—Shakespeare, Jane Austen, and the King James Bible. Richness is computed as the ratio of unique words to total words in each corpus, reflecting lexical diversity.

- **Shakespeare:** 11.5% richness
- **Jane Austen:** 3.81% richness
- **King James Bible:** 2% richness

This comparison highlights Shakespeare’s significantly richer vocabulary, suggesting a greater lexical variety in his writing compared to the other two authors.

```
ubuntu@ip-172-31-37-140:~$ python3 Vocab.py
Hello
Richness of Vocabulary for Shakespeare: 0.1157
Richness of Vocabulary for Austen: 0.0381
Richness of Vocabulary for King James bible: 0.0202
```

- **Part 04: Longest Word in the Corpus**

This output highlights a fascinating linguistic pattern in Shakespeare’s writing. Many of the longest words are compound or hyphenated words from original texts (cleaned of punctuation), reflecting Shakespeare’s creative language structure. For instance, words like *pastoricallycomicalhistoricallypastorally* and *tragicallycomicalhistoricallypastorally* are likely poetic or theatrical descriptors, showcasing the elaborate and stylistic nature of the corpus. Interestingly, as word length decreases, the frequency of occurrence increases—which is expected, as longer words are often more specific or context-dependent, while shorter words are more commonly used in general prose.

```
Top 10 longest word lengths with samples:
39      pastoricallyhistoricallypastorally 1
37      tragicallyhistoricallypastorally 1
20      tragicallyhistorically 1
19      tempestdroppingfire 1
18      lowcrookedcurtsies 2
16      ayredrawn dagger 2
15      voluptuousness 6
14      selflaughter 8
13      trumpet tongue 64
12      satisfaction 117
Deleted /lab1/input
Deleted /lab1/output
```