Homework 1 Report: Text Normalization and Frequency Analysis

Swesan Pathmanathan
`pathmans@mcmaster.ca`

1 Data

The text used in this assignment is *Crime and Punishment* by Fyodor Dostoyevsky, obtained from the Project Gutenberg repository. Project Gutenberg provides public-domain literary works in plain-text format that are intended for human reading, making them well suited for natural language processing tasks.

The corpus consists of a full-length English novel and contains 206,544 total tokens before normalization. The text is composed primarily of narrative prose and dialogue, with clear sentence structure and vocabulary typical of 19th-century literature. Notable characteristics include frequent use of dialogue, proper nouns associated with characters, and stylistic punctuation, all of which influence the resulting token frequency distribution. This corpus provides a representative example of natural language text for analyzing word frequencies and examining Zipf's Law.

2 Methodology

2.1 Overview

The software processes an input text file through a pipeline consisting of file reading, whitespace-based tokenization, optional normalization steps, token frequency counting, and visualization. The program is executed from the command line and allows users to selectively enable normalization options via flags. Token counts are sorted from most frequent to least frequent, and Zipf plots are generated using a log–log scale.

2.2 Normalization Options

The following normalization options are available:

- **Lowercasing**: converts all tokens to lowercase.
- **Stopword Removal**: removes common English stopwords using NLTK's predefined list.
- **Stemming**: applies the Porter Stemmer to reduce tokens to stem forms.
- **Lemmatization**: applies WordNet-based lemmatization to map tokens to dictionary base forms.

Only one of stemming or lemmatization may be enabled at a time.

2.3 Additional Normalization Option

An additional option (`-myopt`) was implemented to remove punctuation-only tokens (e.g., “***”, “...”, “–”). This option was motivated by inspection of the raw output, which revealed many tokens consisting solely of punctuation. Removing these tokens improves the clarity of the frequency analysis without altering semantically meaningful words, making the results easier to interpret for literary text.

3 Sample Output

3.1 Token Counts

Examples of token–frequency pairs produced by the program are shown below.

Most frequent tokens (raw text):

```
the     7404
and     6053
to      5189
a       4433
of      3813
I       3424
he      3364
in      2985
was     2737
you     2734
```

Least frequent tokens (raw text):

```
are     798
my      734
from    709
what    684
```

After applying `-lowercase -stopwords -myopt -lemmatize`, the total number of tokens is reduced to 107,252, and the most frequent tokens shift toward content words:

```
would        566
raskolnikov  546
one          533
could        476
don't        417
```

3.2 Figures

Two Zipf plots were generated: one using the raw text and one after applying normalization. Both plots use a log–log scale with rank on the x-axis and frequency on the y-axis.

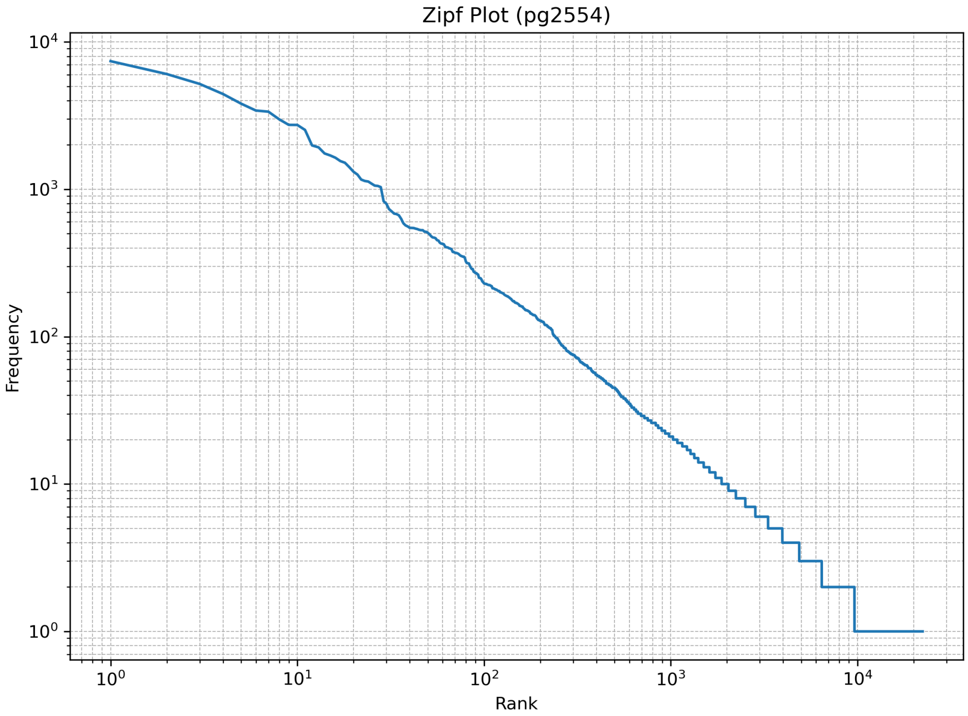

Figure 1: Zipf plot for the raw text of *Crime and Punishment*.

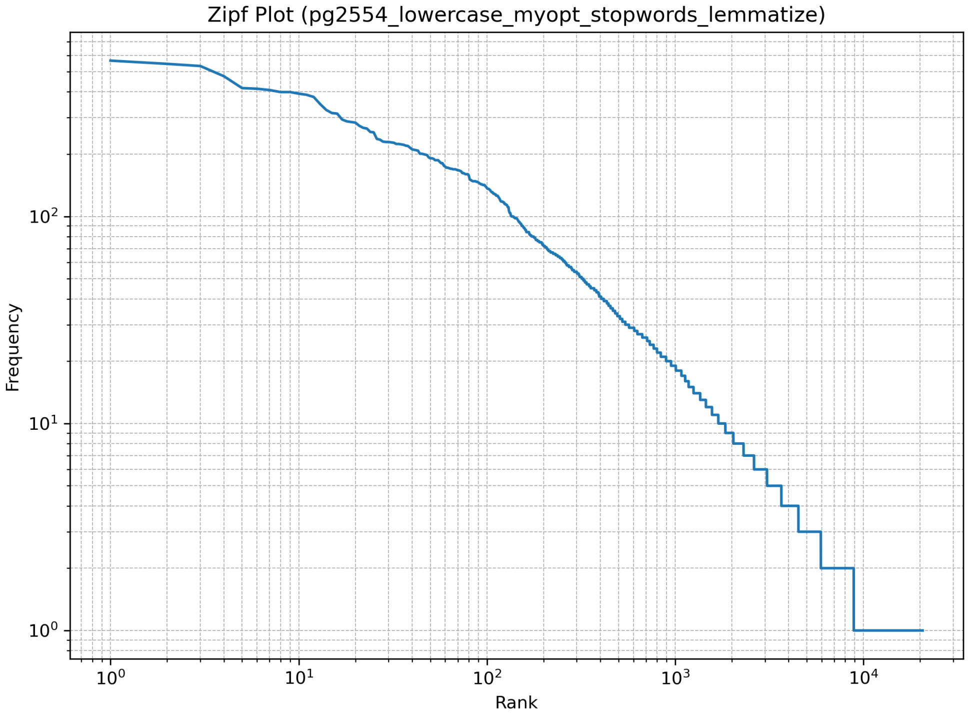

Figure 2: Zipf plot after lowercasing, stopword removal, punctuation-only token removal, and lemmatization.

4 Discussion

4.1 Zipf Analysis

In both the raw and normalized plots, the token frequency distributions follow an approximately linear trend on a log–log scale, which is consistent with Zipf's Law. A small number of tokens occur very frequently, while the majority of tokens appear rarely. In the raw text, the highest-frequency tokens are dominated by function words such as *the* and *and*. After applying normalization, the most frequent tokens shift toward semantically meaningful content words, while the overall Zipfian structure of the distribution remains intact.

4.2 Reflection

This assignment highlighted how strongly preprocessing decisions influence token statistics and interpretability. Simple normalization steps such as lowercasing and removing punctuation-only tokens resulted in noticeable changes in vocabulary size and frequency rankings. Additionally, working with a large literary corpus required careful handling of encoding and memory usage. Overall, the assignment provided practical experience with text processing, command-line interfaces, and data visualization, and reinforced the importance of thoughtful preprocessing in natural language analysis.

Generative AI Usage Disclosure

Generative AI tools (ChatGPT) were used to assist with code structuring, debugging, and drafting the report. All final content was reviewed and edited by the author.

Carbon Footprint Estimate An approximate carbon footprint was estimated using the published value of 4.32 g CO_2 per ChatGPT query. Approximately 25 queries were used, resulting in an estimated total of 108 g CO_2. This estimate is approximate and likely an underestimate due to limited transparency regarding model hardware and deployment region.