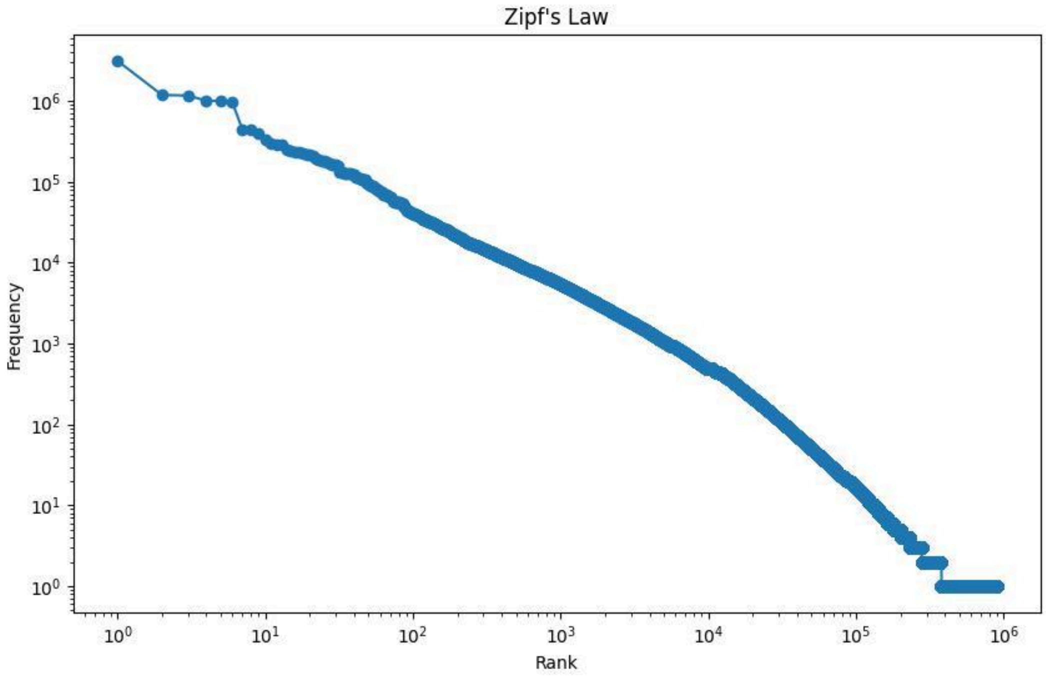
Assignment 1

Eshaan Aggarwal
21075030
Computer Science and Engineering, B.Tech

1. Zipf Law

I chose the English dataset for the analysis of word statistics. I implemented a script in Python to calculate the frequency and rank of the different words in the dataset. I plotted the graph between the logarithmic values of the rank and the frequency of occurrence.



The graph between the same is approximately a straight line with a negative slope, showing that the word's frequency and rank are approximately inversely proportional. Hence, we can say that this exercise empirically verifies Zipf's law.

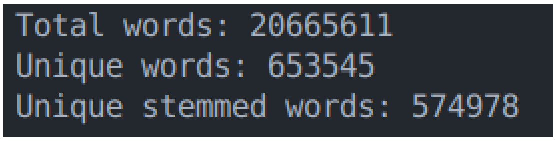
2. Bengali Stemmer

I have used the Benagli Stemmer library, which is available on PyPI. The library stems the rules based on some pre-defined rules, which primarily belong to 3 different categories:

1. When **X** appears at the end of a word, remove it **(X)**
2. When **Y** appears at the end of a word, replace it with **Z (Y -> Z)**

3. When **Y**, followed by some characters, followed by **Z** at the end of a word, replace it with **A.B (Y.Z -> A.B)**

The different words **X, Y, A,** and **B** are defined in a dictionary at the library's core and mapped according to the linguistics of the language. I ran a simple Python script to read all the provided files, used a simple whitespace tokenizer to split the content into words, and then tokenized it with the library's help. In the end, I obtained the following result for the count of tokens:



```
Total words: 20665611
Unique words: 653545
Unique stemmed words: 574978
```

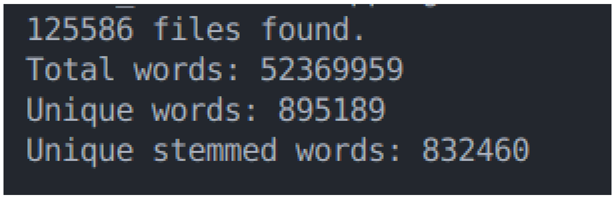
3. **English Stemmer (Porter's Algorithm)**

To implement the Porter's stemming algorithm I have used the nltk library from PyPI. This implementation of the algorithm follows a set of heuristic rules to systematically strip suffixes from words. The rules are designed to handle common English language suffixes and to produce a stem that captures the core meaning of a word.

Here are the main rules of the Porter Stemmer:

1. Remove the plurals from the words **(S, ES)**
2. Remove the past tense from the word **(ED, IED)**
3. Made some common substitutions to remove the adverbial forms **(eg. ATIONAL -> ATE, TIONAL -> TION)**
4. Remove the common verb endings to nouns **(eg. AL, ANCE, ENCE, IC)**

These rules are applied sequentially to a word until a rule matches, and the stemming process stops. In the case of the English dataset provided to us, when I applied the stemming algorithm to the same, the following results were found:



```
125586 files found.
Total words: 52369959
Unique words: 895189
Unique stemmed words: 832460
```