Fundamentals of NLP: Corpus Creation and Pre-Processing

Gülce Erdoğan 21802781 EEE

✦

Abstract - This article explores the fundamental laws of statistical natural language processing and focuses on the development of skills to devise language data, form a corpus, and perform basic pre-processing on the data. The collected textual data is analyzed using several techniques, including Zipf's Law, Heaps' Law, and clustering methodologies. The results of the assignment indicate that all corpora conform to Zipf's Law, and the relationship between increasing vocabulary size and the number of tokens is consistent with Heaps' Law. The findings also suggest that books of the same authors and type have similar slopes for the best fit, enabling the clustering of corpora based on these parameters. The impact of stop words on the corpus was not as expected, and creating a random text produced different results than an actual text. The article emphasizes the importance of preprocessing and tokenization, statistical analysis, and gaining a deeper understanding of corpus linguistics for building NLP models.

1 Introduction

The field of Natural Language Processing (NLP) has gained significant interest in recent years due to its applications in various areas such as speech recognition, sentiment analysis, and machine translation. One of the fundamental aspects of NLP is statistical language modeling, which involves the use of probabilistic models to learn the structure of language from data [1]. In this assignment, the purpose is to explore the fundamental laws of statistical natural language processing, and to develop skills to devise language data, form a corpus and perform basic preprocessing on this data.

The assignment consists of several parts, starting with the collection and preprocessing of textual data from the Gutenberg Project website by tokenization process. We then proceed to analyze the collected data by creating vocabulary files, removing stop words, and plotting Zipf's Law curves for both the author and type corpora. The relationship between token size and vocabulary size is also examined, and we analyze the behavior of the texts as the vocabulary size increases. Finally, we use our findings to explore the possibility of deriving simple clustering methodologies in which we automatically group books so that the members of each group belong to the same author or literary type. This report presents the results of the assignment and discusses our observations and findings.

2 Corpus Construction and Implementation

In the assignment, I will be creating 36 different corpora. 18 of them are the tokenized versions of the books that are downloaded from the Gutenberg project, and the remaining 18 are the stop-words removed versions of them. As I stated I used tokenized as the preprocessing technique and in the tokenization process, I removed all the punctuations and convert all the words to their lowercase. 18 books consist of 3 authors and 3 types having 3 books on each. In other words, I chose the authors according to their books to select text files above 1MB as suggested. Therefore, Fyodor Dostoyevsky, Leo Tolstoy, and Victor Hugo are my final authors due to their large-sized books. I chose Crime and Punishment, The Brothers Karamazov, The Idiot by Dostoyevsky; Anna Karenina, Resurrection, War and Peace by Tolstoy; and Notre-Dame de Paris, The Man Who Laughs, Les Misérables by Hugo. I also look at whether the language of the book is English to prevent unreadable characters and I shoot a glance before downloading them. I chose the types considering the size of the books and my final choices are Classic Novels, Horror, and Science-Fiction. I chose Emma, Jane Eyre, Moby Dick as Classic Novels; Dracula, The Mysteries of Udolpho, The Phantom Ship as Horror; and The Last Men, The Moon Maid, The Mysterious Island as Science-Fiction.

As the programming tool, I utilized Python with Jupiter Notebook in order to create corpora and plot the required plots. Additionally, I saved all the corpora into txt files and all frequency token tuples into a dictionary for further usage.

3 Results

3.1 Part A

In Part A, it is needed to choose the text files that will create the basis of the corpus. Therefore, 3 authors are selected from the http://www.gutenberg.org website and 3 e-books from each author are downloaded as UTF-8 text files [2]. I chose the authors according to their books to select text files above 1MB as suggested. Therefore, Fyodor Dostoyevsky, Leo Tolstoy, and Victor Hugo are my final authors due to their large-sized books. I chose Crime and Punishment, The Brothers Karamazov, The Idiot by Dostoyevsky; Anna Karenina, Resurrection, War and Peace by Tolstoy; and Notre-Dame de Paris, The Man Who Laughs, Les Misérables by Hugo. I also look at whether the language of the book is English to prevent unreadable characters and I shoot a glance before downloading them. Hence, I successfully have a total of 9 books of each author to create author corpora and examine the impacts of different authors on the corpus.

3.2 Part B

In Part B, I chose 9 books as in Part A; however, this time I chose them from 3 distinct types. As in Part A, I chose the types considering the size of the books and my final choices are Classic Novels, Horror, and Science-Fiction. I chose Emma, Jane Eyre, Moby Dick as Classic Novels; Dracula, The Mysteries of Udolpho, The Phantom Ship as Horror; and The Last Men, The Moon Maid, The Mysterious Island as Science-Fiction. Hence, I successfully have a total of 9 books of each type to create type corpora and examine the impacts of different types on the corpus.

3.3 Part C

In Part C, I started to preprocess the text by removing the Gutenburg information at the beginning and the end of the text files. After obtaining the core of the book, I started to tokenize and preprocess the text files. At first, replace the punctuations with empty spaces to eliminate them. I utilized regular expression [^\w\s]|_ meaning any character that is not a word character or a whitespace character. This includes all the punctuation marks, as well as any other non-alphanumeric characters. I also took words with apostrophes separately by removing them (isn't transformed to isnt). Additionally, I cast every word into lowercase to finalize the tokenization process. In the end, I saved the tokenized corpora into a folder as txt files for ease.

3.4 Part D

In Part D, I search different common English Language stop-word lists to create the stop-word-removed version of the corpora. I found that there are many stop-word lists available for the English language, but one of the most widely used is the NLTK (Natural Language Toolkit) stop-word list. The NLTK stop-word list contains 179 common English language stop-words, including words like ”the”, ”of”, ”and”, ”to”, and ”in”. It can easily download this list in Python by installing the nltk library and using the stopwords module [3]. There are also other stop-word lists available for the English language, such as the SMART stop-word list and the Google stop-word list. However, due to its accessibility, I utilized this module. I also remove the punctuation from the stop-word list for its compatibility with the corpora. Then I remove the stop-words from the corpus and saved them into another folder as txt files.

In the end, I obtained a corpus consisting of 18 books in total, encompassing 3 books from three different authors and 3 books from three different literary types. For each of the 18 books, I had two versions - one with stop-words included and one with stop-words removed - resulting in a total of 36 versions of the corpus.

3.5 Part E

In Part E, vocabulary files were generated containing word types and their respective frequencies. This process will be repeated for all books individually, both before and after stop-word removal. While creating vocabulary files, the word type and frequency pairs are saved into dictionaries for further usage.

3.6 Part F

In Part F, I created 3 bigger author corpura by merging all three books written by each author. After obtaining all the author corpora, I plotted the Zipf's Law curves in order to examine the corpora on a linear scale.

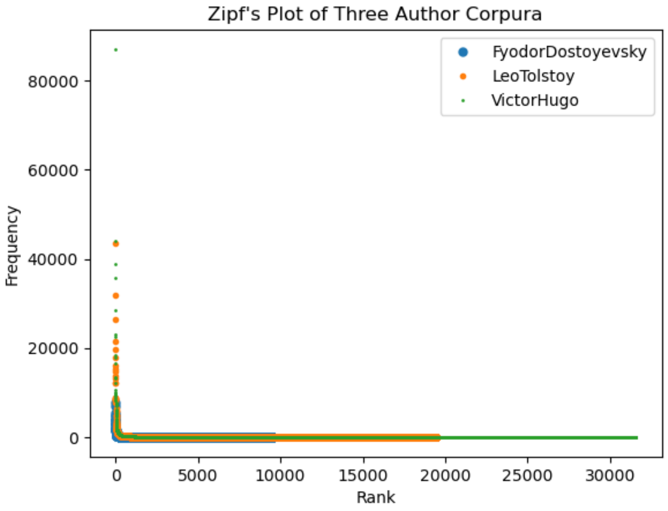

Fig. 1. Zipf's Law Curves for the Three Author Corpora in Linear Scale

In Figure 1, Zipf's Law curves for all author corpora can be examined. As can be observed, all the curves are similar to each other, they even cannot be differentiated. This is because all author corpora are consistent with Zipf's Law. I utilized vocabulary files to plot the rank vs. frequency graph. While utilizing them, I sorted different word types according to their frequency in descending order. Therefore, in Figure 1, the left side of the plot shows the most frequent words and the right side illustrates low frequencies. It is evident that there are certain events that occur with a high frequency, while a vast number of events occur rarely. As a result, the corpora of all authors conform to Zipf's Law, as the pattern observed in the data aligns with the principle established by this law.

Afterward, log-log curves for each of the three books written by each author are plotted as a combination of them into a single figure for each author. In NLP, the log-log curve that follows Zipf's Law represents the frequency distribution of words in a corpus. Zipf's Law states that in a given text corpus, the frequency of any word is inversely proportional to its rank in the frequency table. When the frequency of words in a corpus is plotted against their rank on a log-log scale, the resulting curve follows a straight line [4].

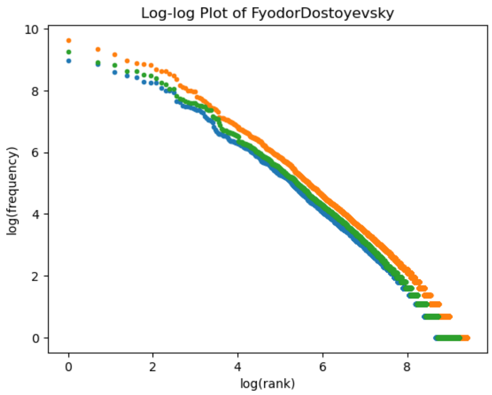

Fig. 2. Log-Log Curves for All Three Books for Dostoyevsky

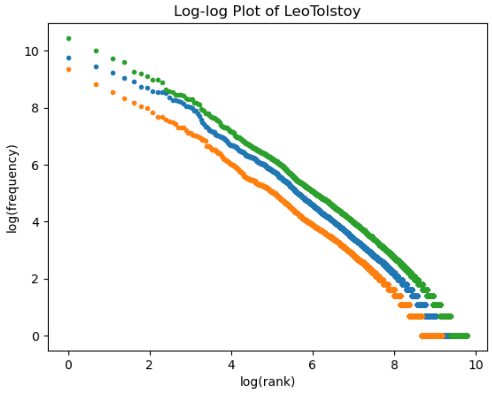

Fig. 3. Log-Log Curves for All Three Books for Tolstoy

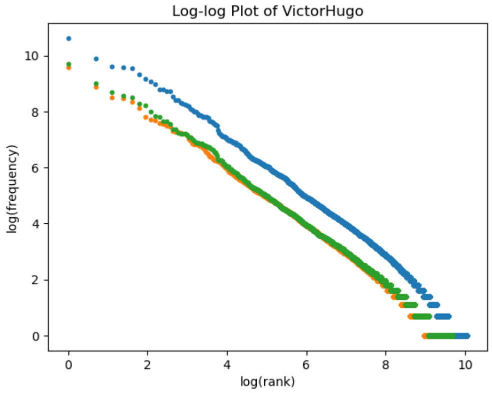

Fig. 4. Log-Log Curves for All Three Books for Hugo

As can be seen in Figures 2-4, all of the books of the authors obey Zipf's Law. Although there are some deviations at low and high parts, the plots show mostly a linear pattern as desired. This is proof that almost all balances corpora that are collected from a wide range of sources comply with Zipf's Law.

3.7 Part G

In Part G, the focus is on analyzing the relationship between the token size and vocabulary size in the corpora of 3 different authors. To fulfill this purpose, I kept track of the number of word types with respect to the increasing token size as I traversed along the corpora. I first tried to observe the relation for every 5000 words and its plots can be examined in Figures 5 and 6.

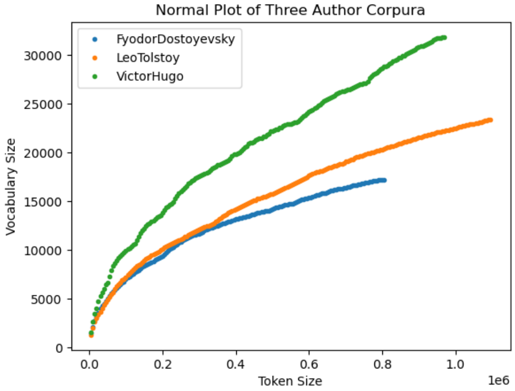

Fig. 5. Relationship Between Vocabulary Size and Token Size of 3 Author Corpora - Normal Version - Step Size = Every 5000 Words

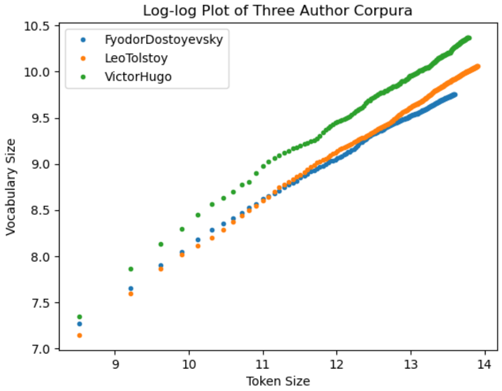

Fig. 6. Relationship Between Vocabulary Size and Token Size of 3 Author Corpora - Log-Log Version - Step Size = Every 5000 Words

I also tried to observe the relation for every 10000 words to better see the relation. Its plots can be examined in Figures 7 and 8.

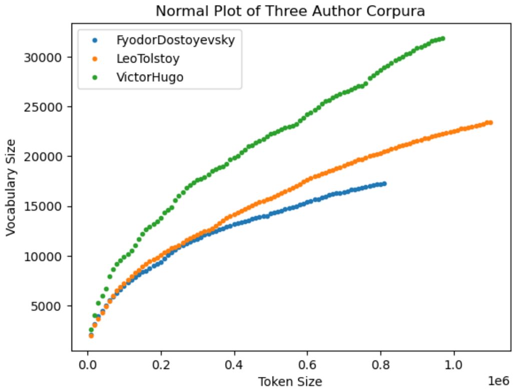

Fig. 7. Relationship Between Vocabulary Size and Token Size of 3 Author Corpora - Normal Version - Step Size = Every 10000 Words

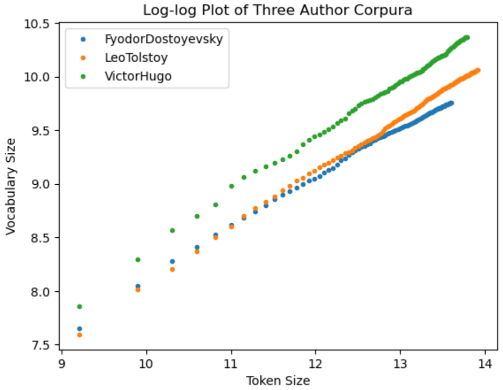

Fig. 8. Relationship Between Vocabulary Size and Token Size of 3 Author Corpora - Normal Version - Step Size = Every 10000 Words

In NLP, the ”vocabulary growth curve” or ”type-token curve” is a graphical representation of the relationship between the size of a corpus (in terms of tokens or words) and the size of its vocabulary (in terms of unique words or types). The curves typically show that as the size of a corpus increases, the number of unique words in the vocabulary also increases, but at a decreasing rate. This means that adding more text to a corpus leads to the discovery of new words, but with diminishing returns. Hence, it can be stated that one common pattern observed in type-token curves can be ”vocabulary saturation” meaning a steep initial increase in vocabulary size as more text is added to the corpus, followed by a more gradual increase at higher levels of the token count. This pattern reflects the fact that as a corpus becomes larger, there are fewer and fewer new words to discover, and many of the words that do appear are rare or infrequent [5].

Another pattern observed in type-token curves is known as ”Heaps’ Law” and states that the vocabulary size of a corpus grows as a power law function of its token count. Heaps’ law has been shown to hold for a wide range of natural language texts, including written texts, speech transcripts, and even computer code.

$$V_R(n) = Kn^{\beta} \tag{1}$$

The two parameters of Heaps’ law are K, the vocabulary richness or the rate at which new words are added to the vocabulary as the corpus size increases, and β, a measure of the rate at which the vocabulary grows as the corpus size increases. The scaling exponent, β typically falls in the range of 0.4 to 0.6, which means that the vocabulary size grows sublinearly with the token count, but faster than a logarithmic function. This implies that as a corpus becomes larger, the vocabulary size will continue to grow, albeit at a slower rate, and that there will always be new and infrequent words to discover. Thus, on one side, all of these patterns are reflections of Zipf’s Law as the law mentions the need of waiting an arbitrarily long time to get valid statistics on low-frequency events [6].

The parameters of the relationship between token size and vocabulary size are the slope and intercept of the linear regression line that is fit to the log-log plot of the data. The slope represents the rate at which the vocabulary size increases with increasing token size, while the intercept represents the vocabulary size when the token size is zero. Additionally, the type-token ratio, which is the ratio of the number of unique words (types) to the total number of words (tokens) in a text, can also be considered a parameter.

3.8 Part H

In Part H, different from Part G each author’s curve was drawn separately in log-log format which can be seen in Figure 9.

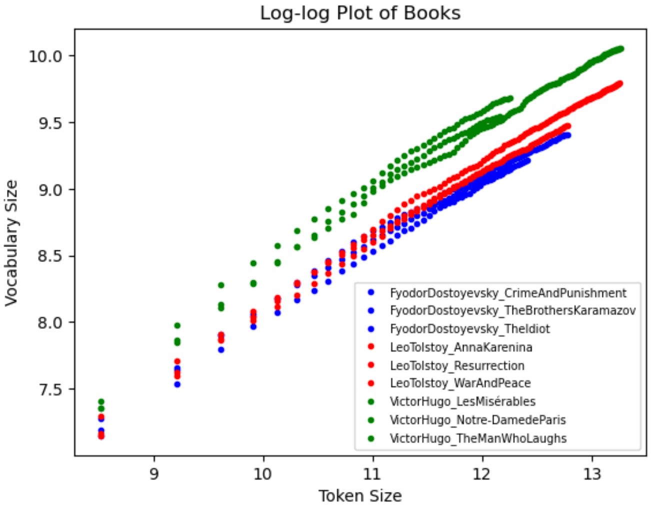

Fig. 9. Relationship Between Vocabulary Size and Token Size of 9 Books of 3 Authors - Log-Log Version - Step Size = Every 5000 Words - Every Author is Represented by a Distinct Color

As in Part G, all the books show a similar pattern with similar parameters. However, as can be observed the plots of books by the same author are close to each other. Therefore, it can be stated that although the slopes and intercepts are similar it is possible to compare different corpora by utilizing such vocabulary growth curve since it provides insights into the linguistic properties of a corpus, such as its lexical richness or the diversity of its vocabulary.

3.9 Part I

In Part I, slopes of the best-fitting lines for all the curves in Part H are found and some of the plots can be seen in below figures.

TABLE 1
Slopes of Best-Fitting Lines for Author Corpora

Author	Book	Best-Fit Book	Best-Fit Author
Dostoyevsky	Crime and Punishment	0.4905	0.4567
	The Brothers Karamazov	0.4778	0.4567
	The Idiot	0.5199	0.4567
Tolstoy	Anna Karenina	0.5228	0.5103
	Resurrection	0.5159	0.5103
	War and Peace	0.5200	0.5103
Hugo	Notre-Dame de Paris	0.5285	0.5201
	The Man Who Laughs	0.5880	0.5201
	Les Misérables	0.5711	0.5201

As can be observed from the table, books by the same author have similar slopes.

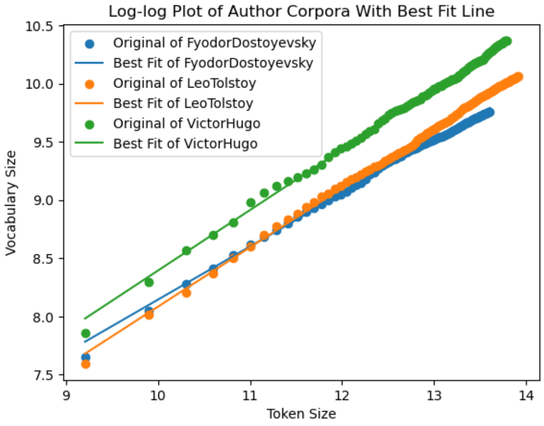

Fig. 10. Log-Log Plot of the Author Corpora With Its Best Fit Line

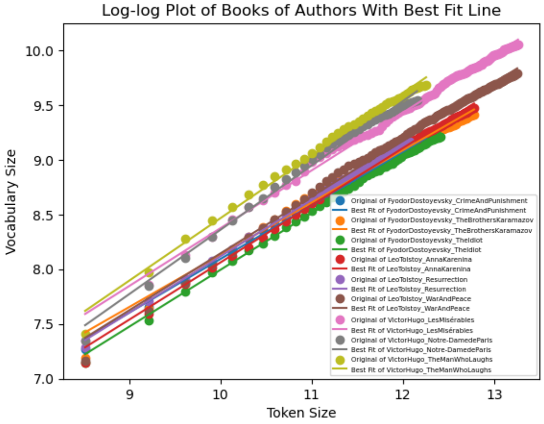

Fig. 11. Log-Log Plot of the Books of Authors With Its Best Fit Line

3.10 Part J

In Part J, I did the same process of Part H, and I for type corpora. The results can be seen as below.

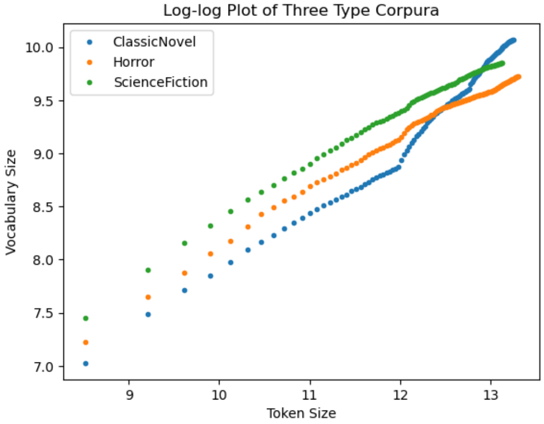

Fig. 12. Relationship Between Vocabulary Size and Token Size of 3 Type Corpora - Log-Log Version - Step Size = Every 5000 Words

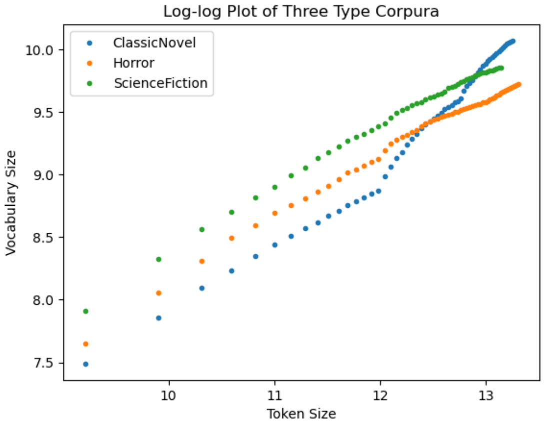

Fig. 13. Relationship Between Vocabulary Size and Token Size of 3 Type Corpora - Log-Log Version - Step Size = Every 5000 Words

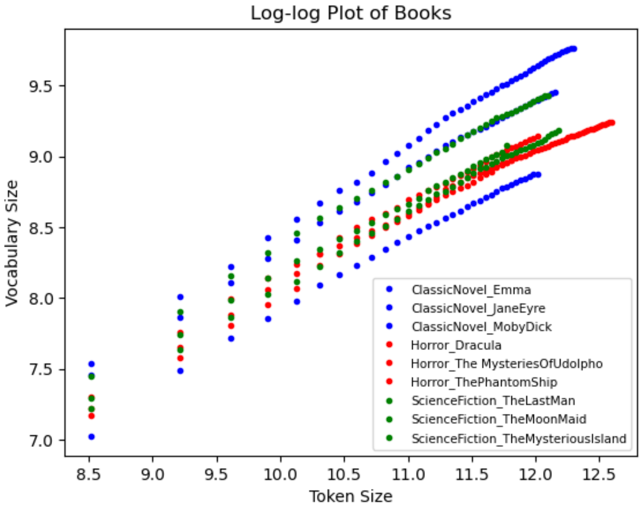

Fig. 14. Relationship Between Vocabulary Size and Token Size of 9 Books of 3 Types - Log-Log Version - Step Size = Every 5000 Words - Every Type is Represented by a Distinct Color

TABLE 2
Slopes of Best-Fitting Lines for Type Corpora

Type	Book	Best-Fit Book	Best-Fit Type
Classic Novels	Emma	0.5077	0.6952
	Jane Eyre	0.5405	0.6952
	Moby Dick	0.5817	0.6952
Horror	Dracula	0.5340	0.4780
	Mysteries of Udolpho	0.4385	0.4780
	The Phantom Ship	0.5376	0.4780
Science-Fiction	The Last Men	0.5328	0.4758
	The Moon Maid	0.5185	0.4758
	Mysterious Island	0.5238	0.4758

As can be observed from the table, books by the same type have similar slopes; however, this similarity is less than the one in author corpora.

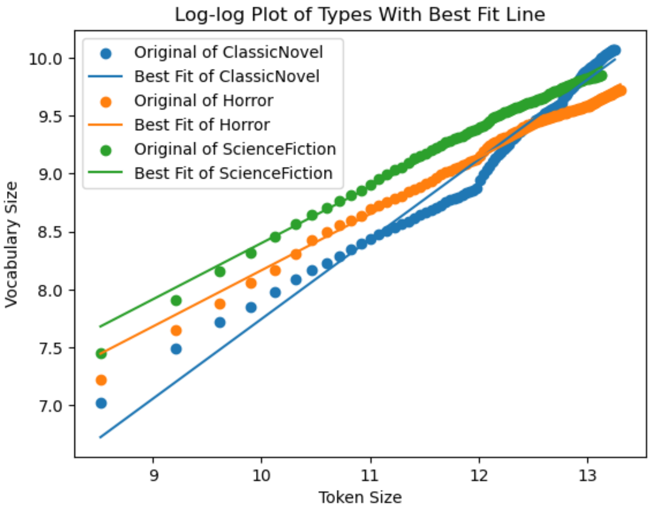

Fig. 15. Log-Log Plot of the Type Corpora With Its Best Fit Line

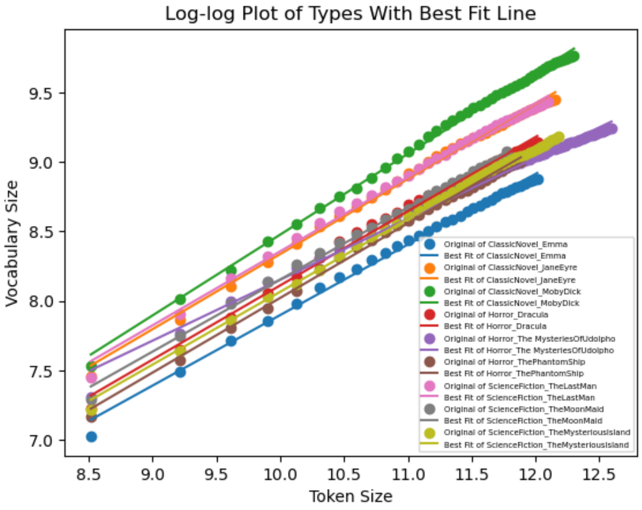

Fig. 16. Log-Log Plot of the Books of Types With Its Best Fit Line

3.11 Part K

Based on the type-token curves, it may be possible to derive simple clustering methodologies to group books automatically based on their authors or literary types. By comparing the slopes and shapes of the curves for each book, we can gain insights into their lexical richness and complexity, and identify patterns that are characteristic of certain authors or literary types.

For example, we may observe that books by the same author tend to have similar slopes or shapes, indicating that they share common vocabulary or linguistic features. Similarly, we may notice that books of the same literary type exhibit similar patterns, such as a faster initial growth rate or a slower saturation point, which reflect their genre-specific characteristics.

To cluster the books based on these patterns, we could use unsupervised learning algorithms such as K-means or hierarchical clustering, and use the type-token curve parameters (e.g., slope, saturation point) or other linguistic features as input variables. By evaluating the resulting clusters against the ground truth labels (i.e., the actual authors or literary types), we can assess the effectiveness and accuracy of the clustering methodologies.

Regarding the question of doing this with five different authors, it may be feasible depending on the availability and representativeness of the data. If we have sufficient data from each author to form meaningful clusters, and if the authors have distinct and recognizable writing styles, then we could potentially apply similar clustering techniques to group the books by the author. However, it may be more challenging to cluster books by literary type if the genres are too similar or diverse. Further experimentation and analysis would be necessary to determine the optimal clustering methodologies and parameters for each scenario.

3.12 Part L

Removing stop-words can significantly change the findings and observations in natural language processing. When stop words are removed, the type-token curve may become steeper, indicating a larger vocabulary size with fewer tokens. This is because the removal of stop words reduces the frequency of certain words in the corpus, making less frequent words more prominent. Additionally, the overall shape of the curve may be altered, as the removal of stop words can affect the distribution of word frequencies in the corpus. Therefore, it is important to carefully analyze the impact of stop word removal on the specific dataset and task at hand before deciding whether or not to remove stop words. However, in my current corpora, I could not be able to see such sharp alterations in my type-token curves. The required curves with the corpus whose stop-words are removed can be observed in the following figures.

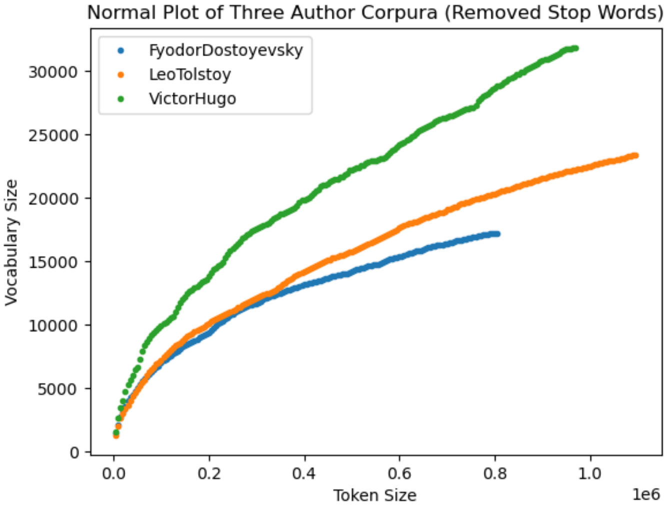

Fig. 17. Relationship Between Vocabulary Size and Token Size of 3 Author Corpora - Normal Version - Step Size = Every 5000 Words - Stop Words Removed Version

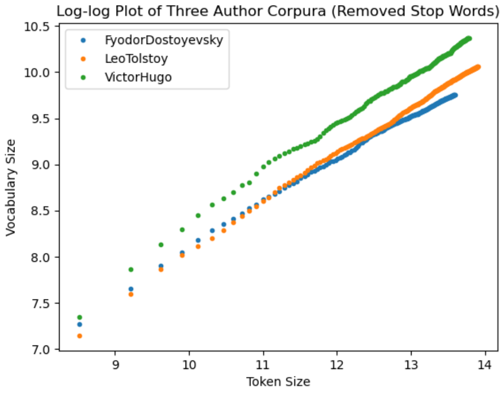

Fig. 18. Relationship Between Vocabulary Size and Token Size of 3 Author Corpora - Log-Log Version - Step Size = Every 5000 Words - Stop Words Removed Version

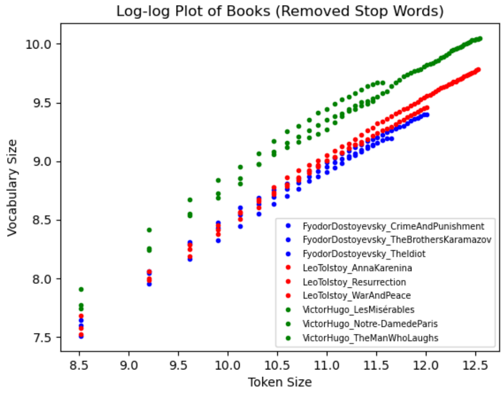

Fig. 19. Relationship Between Vocabulary Size and Token Size of 9 Books of 3 Authors - Log-Log Version - Step Size = Every 5000 Words - Every Author is Represented by a Distinct Color - Stop Words Removed Version

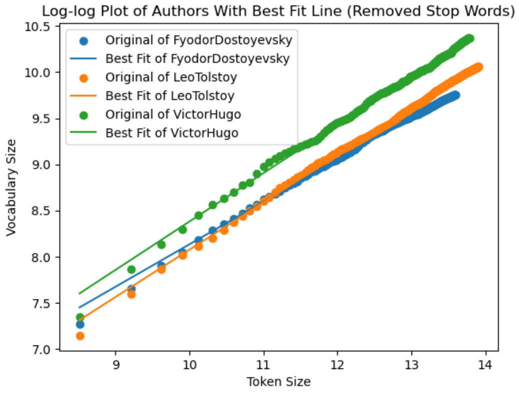

Fig. 20. Log-Log Plot of the Author Corpora With Its Best Fit Line - Stop Words Removed Version

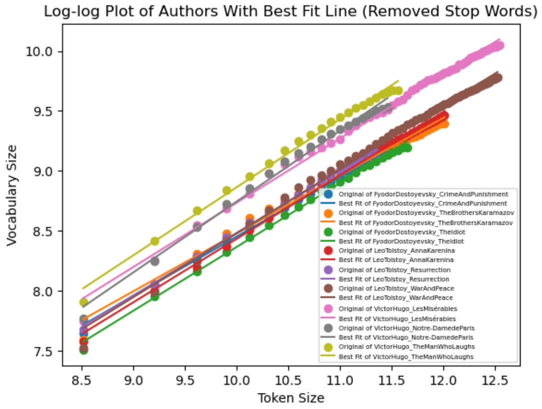

Fig. 21. Log-Log Plot of the Books of Authors With Its Best Fit Line - Stop Words Removed Version

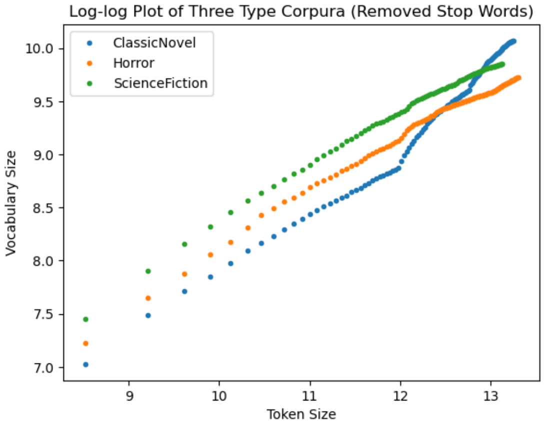

Fig. 22. Relationship Between Vocabulary Size and Token Size of 3 Type Corpora - Log-Log Version - Step Size = Every 5000 Words - Stop Words Removed Version

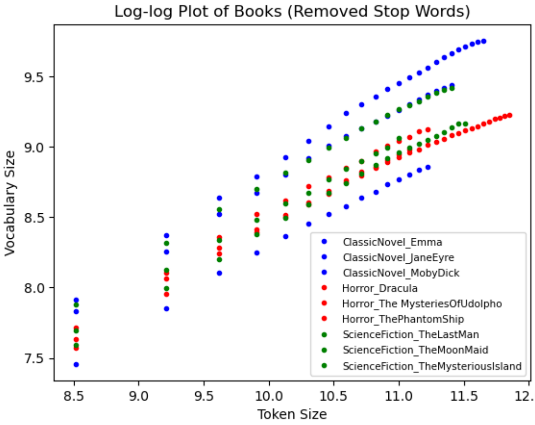

Fig. 23. Relationship Between Vocabulary Size and Token Size of 9 Books of 3 Types - Log-Log Version - Step Size = Every 5000 Words - Every Type is Represented by a Distinct Color - Stop Words Removed Version

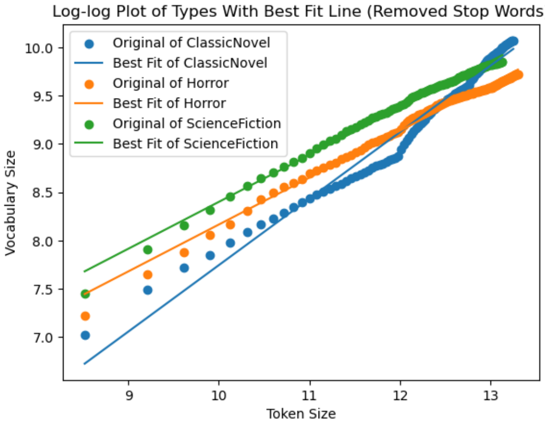

Fig. 24. Log-Log Plot of the Type Corpora With Its Best Fit Line - Stop Words Removed Version

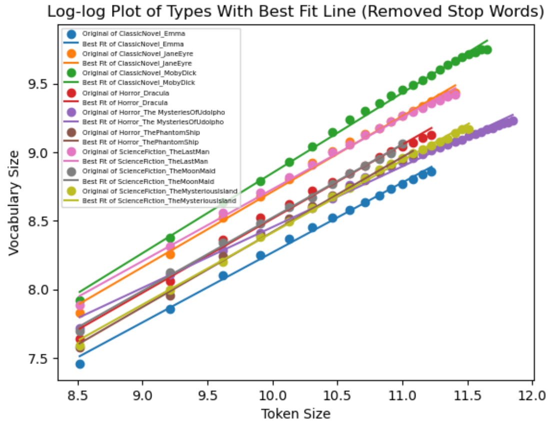

Fig. 25. Log-Log Plot of the Books of Types With Its Best Fit Line - Stop Words Removed Version

3.13 Part M

The paper ”Random texts exhibit Zipf’s-law-like word frequency distribution” by W. Li, published in IEEE Transactions on Information Theory in 1992, explores the phenomenon of Zipf’s law in randomly generated texts. The study finds that random texts exhibit a word frequency distribution that is similar to Zipf’s law, suggesting that this behavior may be a natural consequence of the statistical properties of language [7].

To test this idea, you can generate a random corpus of text and analyze its word frequency distribution. If the corpus is large enough and the words are generated randomly, you should observe a Zipfian distribution. However, it is important to note that the specific parameters of the random text generation process can affect the distribution, so it is important to carefully control these factors.

In comparison to the findings in Part G, where the word frequency distribution of literary texts was found to deviate from a strict Zipfian distribution, the behavior of a randomly generated corpus may show more strict adherence to Zipf’s law. This is because the random text does not contain the complex linguistic patterns and contextual factors present in natural language texts. However, it is still important to carefully analyze the specific properties of the generated corpus to ensure that any observed patterns are not simply artifacts of the random text generation process.

4 Discussions & Conclusions

In conclusion, this assignment provided valuable insights into various statistical techniques for analyzing corpora. The findings clearly demonstrate that all the corpora conform to Zipf’s Law in terms of frequency rank graphs, and the relation between increasing vocabulary size and the number of tokens is consistent with Heaps’ Law. The observation that books of the same authors and type have similar slopes for the best fit enables the clustering of corpora based on these parameters. However, the impact of stop words on the corpus was not as expected, and creating a random text produced different results since a random text it is far from the flow and content of a text. Overall, this assignment provided a solid foundation for building NLP models by emphasizing the importance of preprocessing and tokenization, statistical analysis, and gaining a deeper understanding of corpus linguistics.

References

[1] Wikipedia, *Natural language processing — Wikipedia, the free encyclopedia*, [Online; accessed 11-March-2023], 2023. [Online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing.

[2] *Project gutenberg*, https://gutenberg.org/, Accessed on: March 11, 2023.

[3] Natural Language Toolkit, *Nltk 3.6.3 documentation: Stopwords*, https://www.nltk.org/search.html?q=stopwords&check_keywords=yes&area=default, Accessed: March 11, 2023, 2021.

[4] Wikipedia, *Zipf’s law*, https://en.wikipedia.org/wiki/Zipf%27s_law, Accessed March 11, 2023.

[5] W. Lewis and S. Eetemadi, “Dramatically reducing training data size through vocabulary saturation,” Aug. 2013, pp. 281–291.

[6] Wikipedia, *Heaps’ law*, Accessed: March 11, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Heaps%27_law.

[7] W. Li, “Random texts exhibit zipf’s-law-like word frequency distribution,” *IEEE Transactions on Information Theory*, vol. 38, no. 6, pp. 1842–1845, Nov. 1992.