EEE586: Statistical Foundations of Natural Language Processing Assignment 1

Atakan Topcu, *Bilkent University*

***Abstract*—The process of finding trends in a written piece, such as a book, is still a widely discussed topic in various fields, such as computer science, natural language processing, and media and advertising applications. However, most texts are in their raw and unstructured form. It can be burdensome for engineers and data scientists to analyze them quickly. Therefore, it requires an investigation and processing of texts to increase efficiency. In this assignment, I started by preprocessing and breaking down the given raw texts into smaller units called tokens. I made every word lowercase and removed special characters, punctuation marks, and stop words. I also replaced words with contractions, such as "isn't," with their expanded forms, such as "is not." After the pre-processing stage, I analyzed the texts' properties, such as word frequency, rank ratio (i.e., Zipf's Law), and the ratio of the number of unique words to the total number of tokens (i.e., Heap's law). I performed these experiments on each book of each author or genre individually (18 books in total), as well as on a merged version of three books for each author/genre. Furthermore, these properties go beyond the English language, as randomly created texts display patterns similar to Zipf's law and Heap's law distribution. Given this frequency distribution, the Type-Token Relation (TTR) technique can be utilized to categorize texts.**

***Index Terms*—Natural Language Processing, Computational Linguistics, Zipf's Law, Clustering**

I. Introduction

Rank-frequency relation (RFR) and type-token relation (TTR) are the most common early methods to quantify a text. Before the advance of deep learning, these methods were utilized for natural language processing (NLP) tasks. The most common models for RFR/TTR are Zipf's Law and Heaps' Law, respectively [1], which are also the highlight of this assignment.

In this assignment, we will investigate how these classic models can be utilized and their usability on basic computing tasks such as clustering. As for the dataset, I have used Project Gutenberg, an online library containing a large number of books in the electronic form [2].

In Corpus Construction and Implementation section, the corpora used will be explained. To begin with, we will provide an overview of the corpora, including information about the authors, texts, and sources. Then, we will discuss the techniques used to preprocess the texts and prepare them for analysis. Next, we will provide information on the quantitative properties of the corpora, such as their size, content, and other relevant details. Finally, we will summarize the implementation details, including the programming infrastructure used, the libraries utilized, and the implementation decisions made.

The Results section of this study is presented in four subsections, where the findings are arranged systematically. Visual aids, such as figures and tables, will be utilized to display the findings and will be explained in detail.

The Discussions & Conclusions section will provide an interpretation of the experimental findings. A comparison will be made between the expected results and the actual experimental results. The methodology of the study will also be reviewed. Lastly, a brief and succinct summary of the project's main idea will be presented.

Overall, we have two main tasks to investigate:

- Examine the relationship between the frequency of word types and their corresponding ranks.
- Examine the relationship between the vocabulary size ratio and token size, which refers to the total number of words in the corpus while iterating through the text.
- Examine the usability of such relationships in the context of identifying Genre or Author.

II. Corpus Construction and Implementation

Here, the corpora used for the assignment are further discussed. The corpora that are constructed to be used in this study include 18 books from Project Gutenberg. Nine books were chosen, with three books from each of the three authors, and another nine books were selected, with three books from each of the three different literary genres, as shown in Table I and Table II.

TABLE I
Author Corpora

Author	Book
Leo Tolstoy	War and Peace Anna Karenina The Kingdom of God
Fyordor Dostoyevski	The Brothers Karamazov Crime and Punishment The Idiot
Charles Dickens	A Tale of Two Cities Oliver Twist Great Expectations

Preprocessing

To ensure consistency in the experiments, each book included in this study underwent the same preprocessing steps.

A. Topcu is with the Department of Electrical and Electronics Engineering, Bilkent University, Turkey,

TABLE II
GENRE CORPORA

Genre	Book
Horror	Varney the Vampire The Night Land The Lady of the Shroud
Mystery	The Moonstone The Women in White The Beetle
Sci-Fi	The Mysterious Island Twenty Thousand Leagues Under the Seas The Country of the Blind

As a result of this preprocessing, there were two versions of each of the 18 books, one before and one after stop-word removal, for a total of 36 versions. The process for preprocessing is as follows:

1) The books are obtained in the UTF-8 text format from the Project Gutenberg website.
2) Top and bottom parts of the ebooks are removed (i.e., they contained irrelevant project Gutenberg information)
3) Every word was cast to lowercase.
4) The punctuation marks and apostrophes are removed.
5) Common stop words that are from NLTK [3] are removed.
6) Two versions of each book (with/without common stop words) are saved as CSV files.

As for the resulting quantitative properties, we can see Table III. When we remove stop words from a collection of texts, the number of individual words or tokens decreases significantly, but the reduction in the overall vocabulary is only minor which suggest removing unnecessary words result in significant reduction in size.

Implementation Details

In this assignment, Python is used as programming language and Jupyter Notebook is used as an IDE. The preprocessing of the text data was carried out using Python's String library [4]. To manipulate the multiple CSV files required for the study, Pandas [5] and Numpy [6] were used. The results of the analysis were visualized using Matplotlib [7]. The data was stored in CSV format because of its efficient performance and strong compatibility with the Pandas data frame. Furthermore, python's default libraries have been used for ease of operation such as OS, Copy, collections, random. Finally, KMeans model has been used from scikit-learn library [8] for a simple clustering application to show the utility of Zipf's law and Type-Token relations.

III. RESULTS

In this section, we conduct a series of testing for Zipf's law and Heap's Law both qualitatively and quantitatively.

Part A & B & C & D & E

After acquiring the data, a class named 'Preprocess' has been defined for the preprocessing of the books. The main algorithm of this defined class is given in Algorithm 1

Algorithm 1 A Simple Pseudo of Preprocess(dict, dictFiltered, save, SW)

```
for $k, v$ in $dict.items()$ do
    for $sk, sv$ in $v.items()$ do
        $text \leftarrow$
        for $bk$ in $sv$ do
            $text \leftarrow$ removeSpecialChar(tokenize($bk$))
            $text \leftarrow$ removeSW($text$, get SW() if $SW$ else [])
            if $save$ then
                makeDir('SW' if $SW$ else 'noSW') if not exists
                writeCSV(createFreqFile($bk$, $text$, $save$, $SW$))) if $save$
            end if
        end for
        $dictFiltered[k][sk] \leftarrow text$
    end for
end for
return copy($dictFiltered$)
```

With such a class, I could preprocess all the steps from part A to part E, which included creating vocabulary files carrying the word types along with their frequencies. A detailed version of the class can be found in the code. An example of the created CSV file is also given in Table IV

TABLE IV
EXEMPLARY GENERATED CSV FILE FOR WAR AND PEACE

Word	Frequency	Rank	F*R	log(Freq)	log(Rank)
pierre	1963	1.0	1963.0	7.582229	0.000000
prince	1928	2.0	3856.0	7.564238	0.693147
natásha	1213	3.0	3639.0	7.100852	1.098612
man	1189	4.0	4756.0	7.080868	1.386294
...	...	...	...	...	...
dappled	1	16822.0	16822.0	0.000000	9.730443
clanging	1	16823.0	16823.0	0.000000	9.730502
unreal	1	16824.0	16824.0	0.000000	9.730562

Part F

According to the frequency ranking of words in the selected books [see Table IV], an inverse power law with an exponent close to 1 can be observed. Therefore, it can be concluded that the corpora used in this study provide clear evidence that Zipf's Law is applicable to the English language. The resulting plot can be seen in Fig. 1.

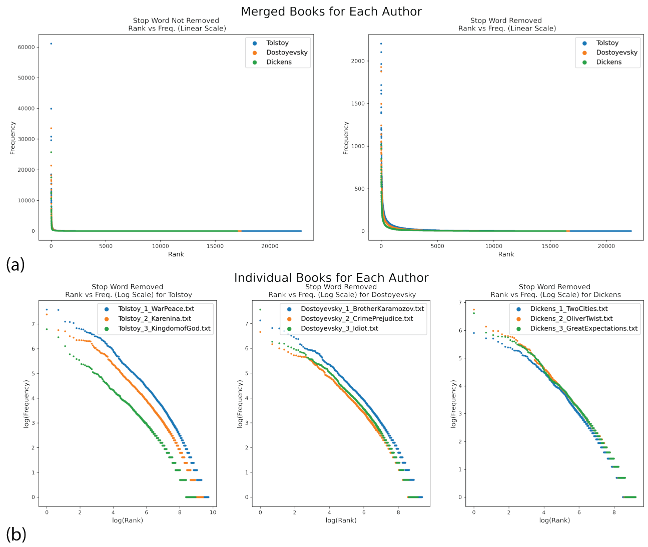

Fig. 1. Zipf's Law curve for authors. (a) Zipf's Law curve for each author's merged books with and without stop words. (b) Individual books of each author with stop words removed.

TABLE III
THE BOOKS WITH THEIR TOKEN AND TYPE COUNTS.

Author/Genre	Book	without SW / Token	without SW / Type	with SW / Token	with SW / Type
Horror	Varney the Vampire	100347	11149	334052	11762
	The Night Land	54115	5096	198349	5572
	The Lady of the Shroud	40862	7966	129392	8571
Mystery	The Moonstone	62955	8189	198218	8771
	The Women in White	82371	9462	250480	10037
	The Beetle	33010	7649	114105	8221
Sci-Fi	The Mysterious Island	71539	8985	194342	9554
	Twenty Thousand Leagues Under the Seas	60728	11419	144910	12025
	The Country of the Blind	58166	11731	164182	12340
Leo Tolstoy	War and Peace	207127	16824	567890	17468
	Anna Karenina	116720	12101	354839	12708
	The Kingdom of God	43689	7736	123754	8318
Fyodor Dostoyevsky	The Brothers Karamazov	112892	11782	354624	12389
	The Crime and Punishment	64863	8736	206230	9325
	The Idiot	75787	9235	246259	9847
Charles Dickens	A Tale of Two Cities	47462	9118	137668	9702
	Oliver Twist	57216	9543	160313	10138
	Great Expectations	57134	10143	187743	10739

In Fig. 1(a), the corpus for each author was compiled by merging all three books of each author. Next, I calculated the frequency of each word in the corpus and sorted them based on their frequency. Then, I plotted the frequency of words against their rank for each author, and Fig. 1(a) shows that the corpus follows Zipf's Law, which means that the frequency of words is inversely proportional to their rank. Even though the sizes of the corpora differ, the word frequency distribution remains the same. Removing stop words from the corpus removed the most common words, resulting in a slightly altered plot, but the corpus still follows Zipf's Law.

In Fig. 1(b), I analyzed the trend for each author and each of their book individually. Thus, I have directly used the CSV file that was generated by Preprocess class. Fig. 1(b) illustrates that the texts composed by the authors conform to Zipf's Law, which states that the frequency of a word is inversely proportional to its rank in the frequency table. The graph presented on a logarithmic scale indicates that the product of frequency and rank is relatively constant, resulting in a linear relationship. However, there is a noticeable difference in the constant value, which remains consistent across authors. The removal of stop words, which are the most frequently used words, has a minor impact on the curves and reduces the frequency values' range. Nonetheless, the corpora continue to follow Zipf's Law.

Part G

Similar to Zipf's Law, there is a classical law that models the relationship between the token size in the corpus and the vocabulary size. This law is called Heaps' Law [9], with the formulation of

$$V = KN^{\beta} where (0 < \beta < 1). \quad (1)$$

Here, V represents the vocabulary (i.e., the number of unique words), and N is the token size. K and β are hyper-parameters determined with experimentation. For the English language, a typical value for K is between 10 and 100, and is between 0.4 and 0.6 [10].

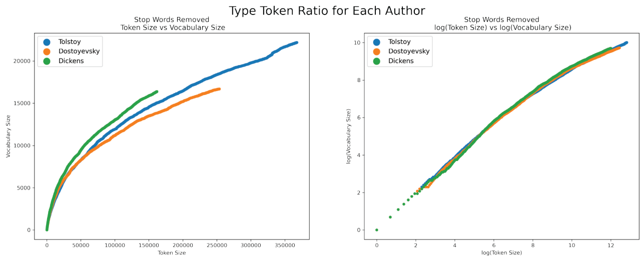

Fig. 2. Heap's Law curve for authors.

Both Zipf's Law and Heaps' Law is a subset of Power Law [11]. However, it is outside the scope of this project.

The resulting plots showing the Heaps' Law can be seen in Fig. 2. Every curve exhibits Heap's Law with different empirical parameters. Thus, there are slight deviations in their trend. Nonetheless, all of them obey the Heaps' Law.

Part H

This time, we will do the same plot for each book of each author. The results can be seen in Fig. 3. Each author exhibits a linear relationship on the logarithmic scale. However, the slopes of these lines are different from one author to another, which also be used for the identification of authors. This is also visible in Fig. 3(d). Since each author is represented with different colors, we can see there are some overlapped areas, but overall, each author is distinguishable.

Part I

Slopes of each author from Fig. 3 given in Table V. Books from the same author seem to have similar slopes, especially for Dostoyevski and Dickens. This suggests that the slopes of each book can be used for author identification.

Part J

Instead of using authorship, we repeat the same procedures for different Genres. The resulting plots are given in Fig. 4

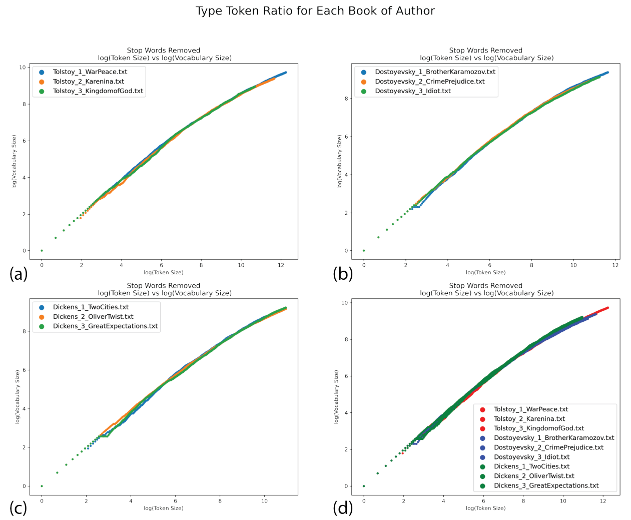

Fig. 3. Heap's Law curve for each book of each author. (a) For Tolstoy, (b) For Dostoyevski, (c) For Dickens, and (d) all of the authors combined with different colors.

TABLE V
AUTHOR'S CORPORA AND THEIR INDIVIDUAL SLOPES

Corpora	without SW / Slope	with SW / Slope
War and Peace	0.571	0.528
Anna Karenina	0.596	0.541
Kingdom of God	0.681	0.607
The Brothers Karamazov	0.575	0.513
Crime and Punishment	0.601	0.533
The Idiot	0.609	0.540
A Tale of the Two Cities	0.676	0.600
Oliver Twist	0.648	0.582
Great Expectations	0.682	0.601

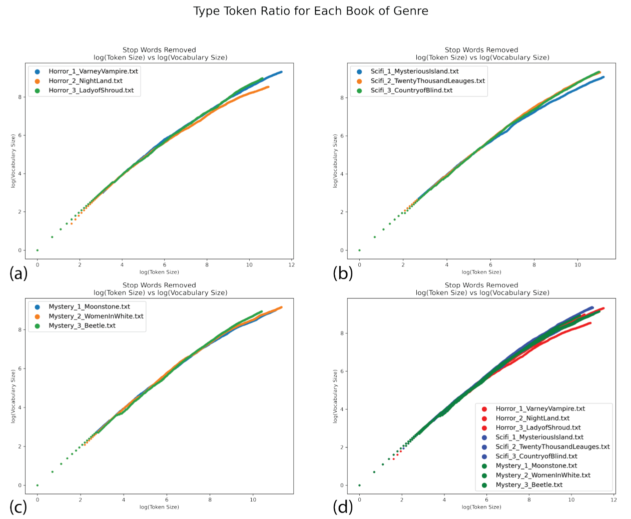

Fig. 4. Heap's Law curve for each book of each Genre. (a) For Horror, (b) For Sci-Fi, (c) For Mystery, and (d) all of the Genres combined with different colors.

TABLE VI
GENRE'S CORPORA AND THEIR INDIVIDUAL SLOPES

Corpora	without SW / Slope	with SW / Slope
Varney the Vampire	0.614	0.542
The Night Land	0.553	0.475
The Lady of the Shroud	0.711	0.623
The Moonstone	0.602	0.537
The Women in White	0.569	0.508
The Beetle	0.714	0.624
The Mysterious Island	0.609	0.551
Twenty Thousand Leagues Under the Seas	0.657	0.593
The Country of the Blind	0.692	0.624

TABLE VII
AUTHOR'S CORPORA AND THEIR CLUSTERING RESULTS.

Corpora	without SW / Group	with SW / Group
War and Peace	0	0
Anna Karenina	2	0
Kingdom of God	1	1
The Brothers Karamazov	0	2
Crime and Punishment	2	0
The Idiot	2	0
A Tale of the Two Cities	1	1
Oliver Twist	1	1
Great Expectations	1	1

Slopes for each Genre are given in Table VI. Similar to the case Table V, the slope of each Genre differs from each other, especially after stop words are removed. Nonetheless, this difference between each Genre seems to be more subtle compared to Author's case.

Part K

Differences in writing complexity among authors and their books can be differentiated by analyzing Zipf's Law and Heaps' Law curves. Books by the same author seem to have similar curves/slopes to those by different authors. This phenomenon can be used to cluster similar texts automatically.

To show the applicability, I have used the slopes of the log(Token Size) vs. log(Vocabulary Size) curve's best-fitting lines as a 1-dimensional feature and passed it to a KMeans model with k=3. The results are given in Table VII and Table VIII.

It can be concluded that simple clustering can be done using the slopes of the log(Token Size) vs. log(Vocabulary Size). Even though this method is rather primitive, it can still correctly cluster books, especially for Authors rather than Genres. Furthermore, the effect of removing stop words is also visible in both tables.

Part L

To investigate the effect of stop words, I have done all the parts with and without stop words. In Table VII and Table VIII, we can see the effect of stop words on the clustering of the books. If we don't discard the stop words, the differentiability of the Authors/Genres decreases, resulting in false clusters as well. Though their effect is not as visible as the slopes, stop words also change the distribution trend for each book which can be visible in Fig. 5 and Fig. 6.

TABLE VIII
GENRE'S CORPORA AND THEIR CLUSTERING RESULTS.

Corpora	without SW / Group	with SW / Group
Varney the Vampire	2	2
The Night Land	0	0
The Lady of the Shroud	1	1
The Moonstone	2	2
The Women in White	2	1
The Beetle	1	1
The Mysterious Island	2	2
Twenty Thousand Leagues Under the Seas	0	0
The Country of the Blind	1	1

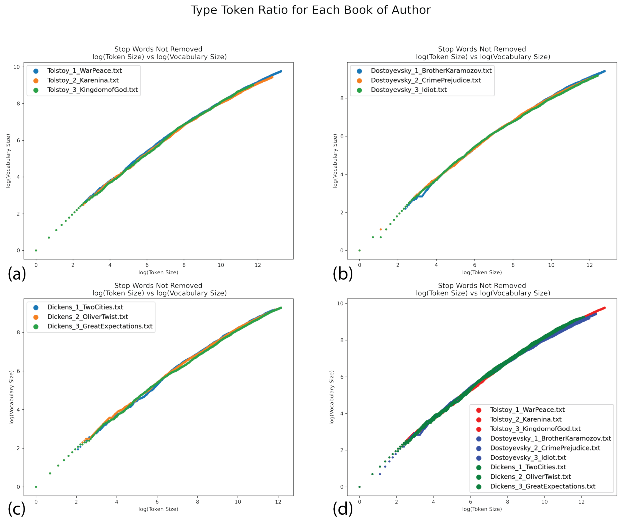

Fig. 5. Heap's Law curve for each author's book with stop words. (a) For Tolstoy, (b) For Dostoyevski, (c) For Dickens, and (d) all of the authors combined with different colors.

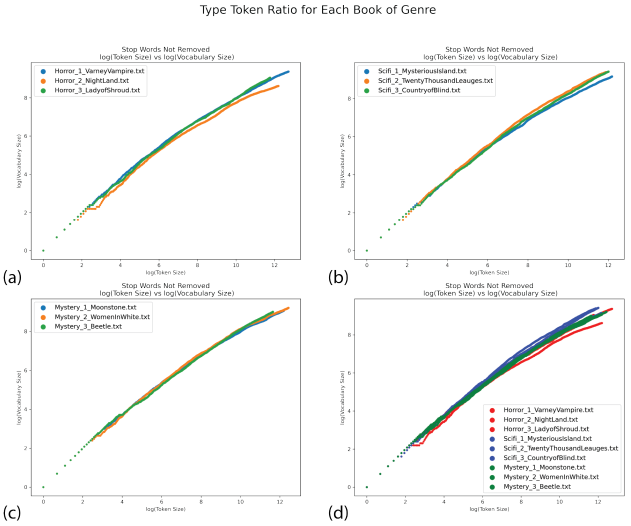

Fig. 6. Heap's Law curve for each book of each Genre with stop words. (a) For Horror, (b) For Sci-Fi, (c) For Mystery, and (d) all of the Genres combined with different colors.

Part M

According to previous research, the shallowness of Zipf's law was demonstrated with random corpus [12]. To test their conclusion, I have generated a random corpus of 750000 with a maximum word limit of 7. I have generated random lowercase, random words using English characters. Then, I computed the frequency of each word along with the overall type-token relation. The result is given in Fig. 7.

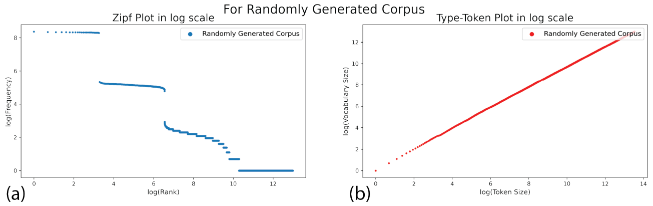

Fig. 7. (a) Zipf's Law curve for random corpus, (b) Heap's Law curve for random corpus.

It is seen that the overall generated random corpus follows and obeys both Zipf's Law and Heaps' Law. Even though there are gaps in Fig. 7(a), the overall trend is still similar to the trend of the original corpus. Similarly, in Fig. 7(b), we see a linear trend, while in the original corpus, we had a slight curve on the trend. Nevertheless, the overall trend is still preserved for both Fig. 7(a) and Fig. 7(b).

IV. DISCUSSIONS & CONCLUSIONS

All in all, the main idea of this investigation was to explore and understand two power laws which are Zipf's and Heaps' Law. Using such laws, the distribution of words and their applicability in Author/Genre detection was investigated. A total of 18 books were analyzed, including 3 books from 3 different authors and 3 books from 3 different literary genres. The project results showed that the selected books obey both Zipf's Law and Heaps' Law. Using the type-token ratio (TTR) to group the books by the author showed some success, but attempting to group them by genre was not as successful. Furthermore, the study demonstrated that a randomly generated corpus exhibited Zipf's Law and Heaps' Law, similar to natural languages, which showed the shallowness of these power laws.

Though the usage of such laws and methods was useful initially, they alone do not provide the means to develop practical applications. To use them in applications, they must be supported by techniques such as deep learning.

REFERENCES

[1] J. Milička, "Type-token hapax-token relation: A combinatorial model," *Glottotheory. International Journal of Theoretical Linguistics*, vol. 2, pp. 99–110, 01 2009.
[2] "Project gutenberg." [Online]. Available: https://www.gutenberg.org/
[3] [Online]. Available: https://www.nltk.org/
[4] "String - common string operations¶." [Online]. Available: https://docs.python.org/3.7/library/string.html
[5] "Pandas." [Online]. Available: https://pandas.pydata.org/
[6] [Online]. Available: https://numpy.org/
[7] "Visualization with python." [Online]. Available: https://matplotlib.org/index.html
[8] "Learn." [Online]. Available: https://scikit-learn.org/stable/

[9] John, "Heaps law: Estimating vocabulary size from total words," Aug 2021. [Online]. Available: https://www.johndcook.com/blog/2019/08/27/heaps-law/

[10] L. Egghe, "Untangling herdan's law and heaps' law: Mathematical and informetric arguments," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 5, p. 702–709, 2007.

[11] L. Lü, Z.-K. Zhang, and T. Zhou, "Deviation of zipf's and heaps' laws in human languages with limited dictionary sizes," *Scientific Reports*, vol. 3, no. 1, 2013.

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