NLP PROJECT REPORT

Submitted by

Team *'Linguists'*

Aniket Shukla (19UCS057)

Anushka Vyas (19UCS041)

Harsh Kumar (19UCS185)

in partial fulfillment for the award of the degree of

B.Tech. *in* **CSE**

Github Repository Link -: https://github.com/DragoPhoenix/NLP-Project

ACKNOWLEDGEMENT

We would like to express our special thanks to our project guide **Dr. Sakthi Balan** who gave us the golden opportunity to do this wonderful project on the topic of **Natural Language Processing** which also helped us in doing a lot of research and it was a great learning experience.

Date: 21 Dec 2021

Aniket Shukla (19UCS057)

Anushka Vyas (19UCS041)

Harsh Kumar (19UCS185)

TABLE OF CONTENTS

ABSTRACT

Text Analytics is a very important aspect in the field of natural language processing and in this project, we worked on text preprocessing, PoS tagging, information extraction, document similarity measures, and various other operations on two of the very well-known books ***Pride and Prejudice*** and ***The Adventures of Sherlock Holmes*** and the third additional book (for Round 2) ***Emma***. Working on this project on these two books was particularly interesting because they are written from very different points of view and are completely different from each other in terms of genre. For all the operations performed, we have used NLTK (natural language ToolKit) to perform all the preprocessing, tokenization, and tagging as required. The project also describes the frequency distribution and WordCloud of both of the books and helped us understand some fields of text mining. The graphs that are plotted in the report also say a lot about the input text which is derived from the books and writing style of the authors, the words that they use frequently, main characters, genre, etc. This project can also be used in understanding vocabulary in a certain text file, frequency of words, difficulty in the text file, document analysis, document comparison, etc.

LITERATURE REVIEW

Many researchers worked on NLP, building tools and systems which make NLP what it is today. Tools like Sentiment Analyser, Parts of Speech (POS) Taggers, Chunking, Named Entity Recognition (NER), Emotion detection, Semantic Role Labelling made NLP a good topic for research.

Related Work :

- Sentiment analyser (Jeonghee et al.,2003) [26] works by extracting sentiments about a given topic.

- Parts of speech taggers for the languages like European languages, research is being done on making parts of speech taggers for other languages like Arabic, Sanskrit (Namrata Tapswi, Suresh Jain ., 2012) [27], Hindi (Pradipta Ranjan Ray et al., 2003)[28], etc. It can efficiently tag and classify words as nouns, adjectives, verbs, etc.

- The Sanskrit part of speech tagger specifically uses the treebank technique.

- Arabic uses the Support Vector Machine (SVM) (Mona Diab et al.,2004) [29] approach to automatically tokenize, tag parts of speech, and annotate base phrases in Arabic text.

PROJECT ROUND - 1

1. INTRODUCTION

In this Project, we imported two large books in text format in order to perform
text analysis using NLP techniques on them. We tokenized and lemmatized the imported text files, analyzed the frequency distribution, and performed PoS tagging on both of the text files, further we are going to compare these results for each text file and visualize the data which is going to be a good learning experience in the field of NLP.

To make it interesting we have chosen two very famous books from very different genres. These books are among the top downloaded books on the Project Gutenberg website.

- First is **Pride and Prejudice** which is a novel written by **Jane Austen.**
- Second book is **The Adventures of Sherlock Holmes** which is a collection of twelve short stories by **Arthur Conan Doyle.**

For the tasks given in this project, we have used *nltk* which is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language. And we have imported some modules and functions in order to perform different activities such as preprocessing, tokenizing, removing stop words etc. And after performing such operations on both of the text files we have plotted frequency distribution for both text files and created word clouds.

2. Python Libraries and Modules Used

Nltk.tokenizer package : Tokenizer divides strings into a list of substrings.

Nltk.stem package : Interfaces used to remove morphological affixes from words, leaving only the word stem.

Nltk.probability : A probability distribution specifies how likely it is that an experiment will have any given output.

Nltk.corpus : The modules in this package provide functions that can be used to read corpus files in a variety of formats.

Wordcloud package : Provides modules to create wordcloud in python.

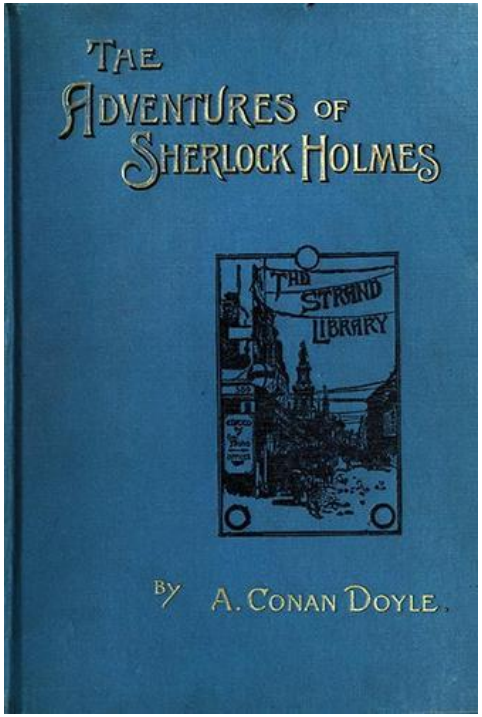
Collection modules : Provide different types of container data types.

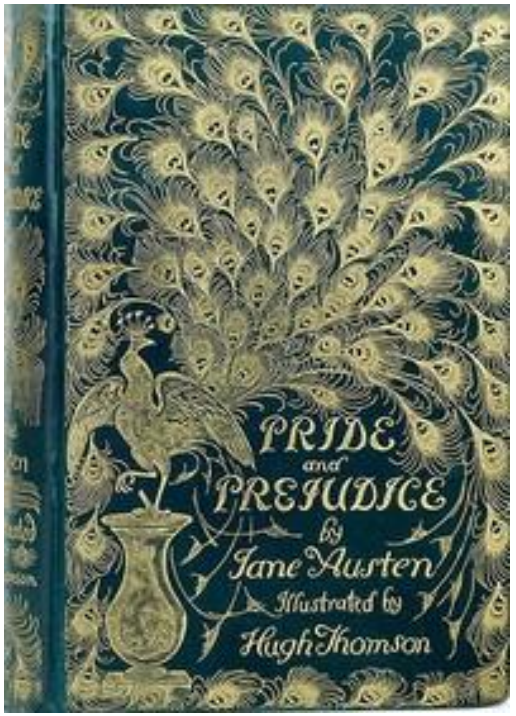
3. METHODOLOGY

Downloading Books :

- We downloaded the following two books in Plain Text UTF-8 format for text processing from Project Gutenberg (http://www.gutenberg.org)
    - **Pride and Prejudice** by **Jane Austen** (https://www.gutenberg.org/ebooks/1342)
    - **The Adventures of Sherlock Holmes** by **Arthur Conan Doyle** (https://www.gutenberg.org/ebooks/1661)

The downloaded files were .txt files.

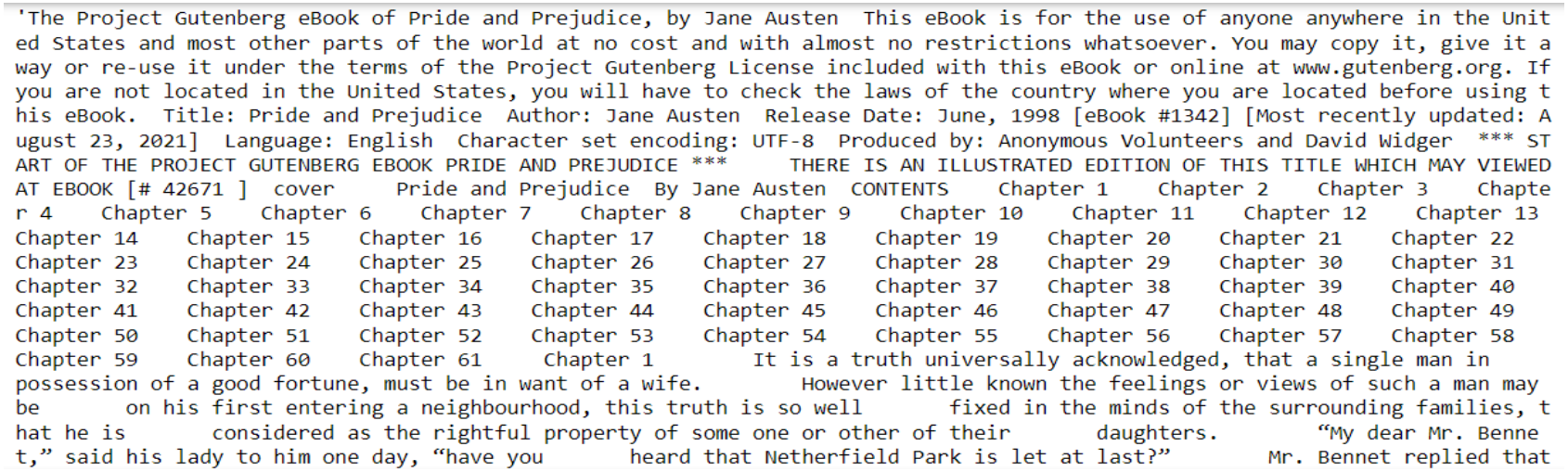




Covers of the two Books selected for Text Analysis

Importing the text :

- In this step we created a function (*txt_file_to_string)* to read the text imported from both the books and convert them into string for processing in python. This function takes as input the path of the file to be read and returns the content of the file in string format. We stored our first book “Pride and Prejudice” in the variable Original_T1 of string type. Similarly, we stored the text from the second book “The Adventures of Sherlock Holmes” in the variable Original_T2 of string type.



```
'The Project Gutenberg eBook of Pride and Prejudice, by Jane Austen  This eBook is for the use of anyone anywhere in the Unit
ed States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it a
way or re-use it under the terms of the Project Gutenberg License included with this eBook or online at www.gutenberg.org. If
you are not located in the United States, you will have to check the laws of the country where you are located before using t
his eBook.  Title: Pride and Prejudice  Author: Jane Austen  Release Date: June, 1998 [eBook #1342] [Most recently updated: A
ugust 23, 2021]  Language: English  Character set encoding: UTF-8  Produced by: Anonymous Volunteers and David Widger  *** ST
ART OF THE PROJECT GUTENBERG EBOOK PRIDE AND PREJUDICE ***     THERE IS AN ILLUSTRATED EDITION OF THIS TITLE WHICH MAY VIEWED
AT EBOOK [# 42671 ]  cover     Pride and Prejudice  By Jane Austen  CONTENTS    Chapter 1    Chapter 2    Chapter 3    Chapte
r 4    Chapter 5    Chapter 6    Chapter 7    Chapter 8    Chapter 9    Chapter 10    Chapter 11    Chapter 12    Chapter 13
Chapter 14    Chapter 15    Chapter 16    Chapter 17    Chapter 18    Chapter 19    Chapter 20    Chapter 21    Chapter 22
Chapter 23    Chapter 24    Chapter 25    Chapter 26    Chapter 27    Chapter 28    Chapter 29    Chapter 30    Chapter 31
Chapter 32    Chapter 33    Chapter 34    Chapter 35    Chapter 36    Chapter 37    Chapter 38    Chapter 39    Chapter 40
Chapter 41    Chapter 42    Chapter 43    Chapter 44    Chapter 45    Chapter 46    Chapter 47    Chapter 48    Chapter 49
Chapter 50    Chapter 51    Chapter 52    Chapter 53    Chapter 54    Chapter 55    Chapter 56    Chapter 57    Chapter 58
Chapter 59    Chapter 60    Chapter 61     Chapter 1        It is a truth universally acknowledged, that a single man in
possession of a good fortune, must be in want of a wife.        However little known the feelings or views of such a man may
be       on his first entering a neighbourhood, this truth is so well       fixed in the minds of the surrounding families, t
hat he is       considered as the rightful property of some one or other of their       daughters.        “My dear Mr. Benne
t,” said his lady to him one day, “have you       heard that Netherfield Park is let at last?”        Mr. Bennet replied that
```

The content of *Original_T1* after reading from .txt file

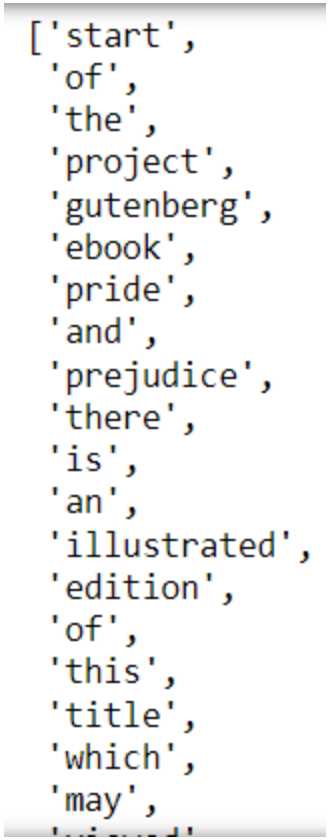
Text Pre-Processing and Tokenization :

1. Removing prefix and suffix to narrow down to text from eBook - Each book from the website had additional prefix and suffix in its .txt file, apart from the contents of the eBook. To narrow down our string to the relevant part only we considered only the substring of the original string which was marked with *** *START OF THE PROJECT* and *** *END OF THE PROJECT* in the .txt files.
2. Lowercase - We converted our string to lowercase using the string function *string.lower()*.
3. Expansion of some Contractions - We expanded some generic contractions. For example : *can’t* to *can not,* all instances of *‘ll* to *will* etc. This is not very accurate and will lead to some incorrect expansion since disambiguation to the right expansion is not deterministic but this will be correct for most cases.

4. Removal Of Punctuations - We removed all the punctuation using regular expressions in two steps by first replacing everything other than word and whitespace characters with empty string and then replacing _ (underscore, which is considered part of word in python) by empty string.
5. Removing unnecessary repeated words - We have removed the term *chapterXYZ* using regular expression substitution since it is not necessary and increases the frequency of the word 'chapter' if present in the chapter headings or index of the book.
6. Replacing one or more continuous white space characters with single space to make the string evenly spaced.
7. We tokenized the strings T1 and T2 into single words using *word_tokenize()* function imported from *nltk* and stored them into the variables *Tokenized_T1* and *Tokenized_T2* respectively. Then we lemmatized these lists as we plan to analyze word frequencies later, therefore reducing the words to their lemma form will be suitable.

```
' start of the project gutenberg ebook pride and prejudice there is an illustrated edition of this title which may viewed at
ebook 42671 cover pride and prejudice by jane austen contents it is a truth universally acknowledged that a single man in pos
session of a good fortune must be in want of a wife however little known the feelings or views of such a man may be on his fi
rst entering a neighbourhood this truth is so well fixed in the minds of the surrounding families that he is considered as th
e rightful property of some one or other of their daughters my dear mr bennet said his lady to him one day have you heard tha
t netherfield park is let at last mr bennet replied that he had not but it is returned she for mrs long has just been here an
d she told me all about it mr bennet made no answer do not you want to know who has taken it cried his wife impatiently you w
ant to tell me and i have no objection to hearing it this was invitation enough why my dear you must know mrs long says that
netherfield is taken by a young man of large fortune from the north of england that he came down on monday in a chaise and fo
ur to see the place and was so much delighted with it that he agreed with mr morris immediately that he is to take possession
before michaelmas and some of his servants are to be in the house by the end of next week what is his name bingley is he marr
ied or single oh single my dear to be sure a single man of large fortune four or five thousand a year what a fine thing for o
ur girls how so how can it affect them my dear mr bennet replied his wife how can you be so tiresome you must know that i am
thinking of his marrying one of them is that his design in settling here design nonsense how can you talk so but it is very l
ikely that he may fall in love with one of them and therefore you must visit him as soon as he comes i see no occasion for th
at you and the girls may go or you may send them by themselves which perhaps will be still better for as you are as handsome
as any of them mr bingley might like you the best of the party my dear you flatter me i certainly have had my share of beauty
but i do not pretend to be anything extraordinary now when a woman has five grownup daughters she ought to give over thinking
of her own beauty in such cases a woman has not often much beauty to think of but my dear you must indeed go and see mr bingl
```

The contents of T1 after pre-processing



The list of tokenized and lemmatized words of T1

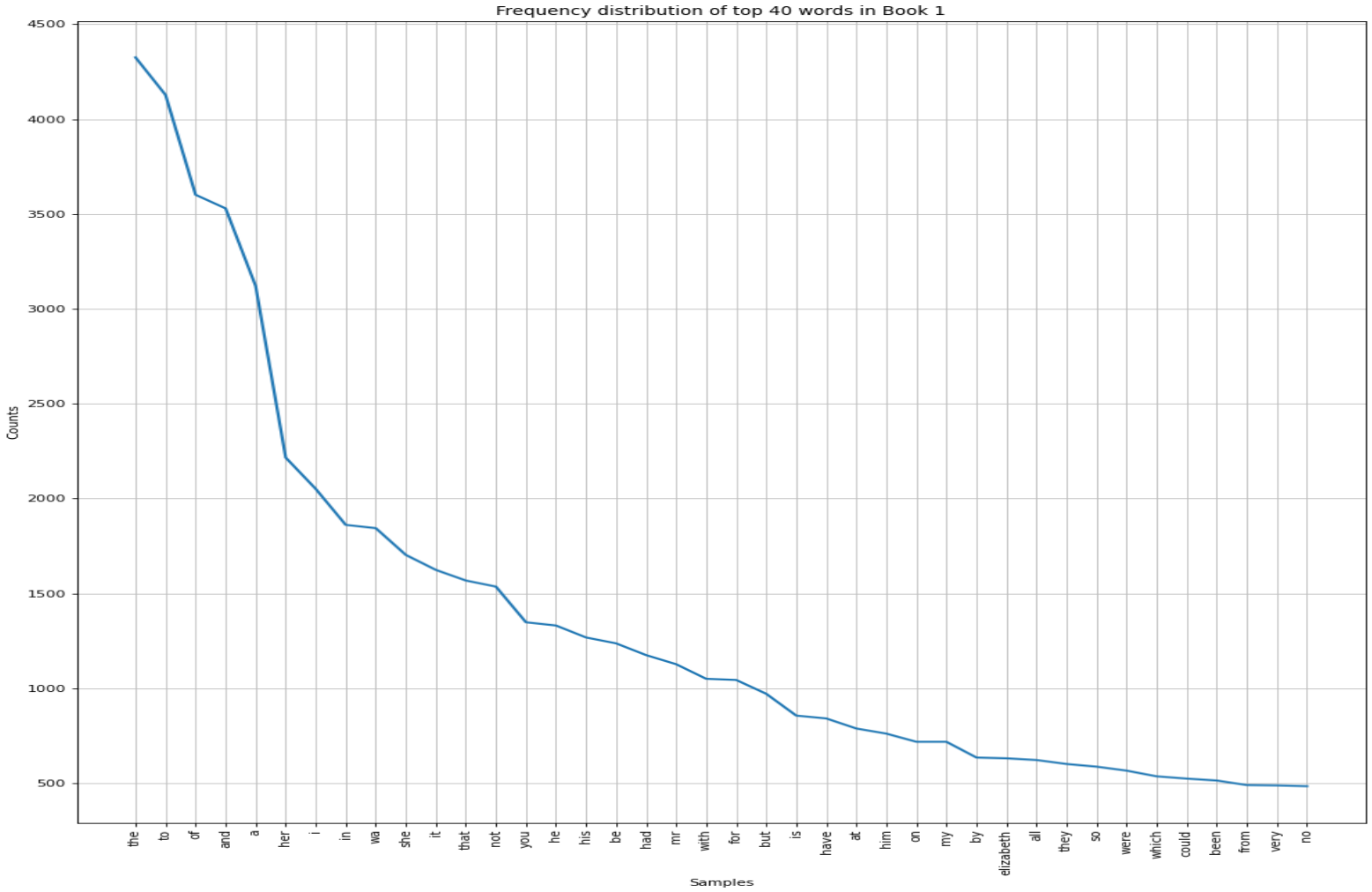
Plotting Frequency distribution of tokens :

- After tokenizing the text, we imported the function *FreqDist()* from the module *nltk.probability* which is helpful in probability calculations, where frequency distribution counts the number of times that each outcome of an experiment occurs. We stored the frequency distribution of the two strings in the *FreqDist* object by passing the tokenized and lemmatized lists to the above function.

```
FreqDist({'the': 4325, 'to': 4127, 'of': 3601, 'and': 3529, 'a': 3120, 'her': 2216, 'i': 2051, 'in': 1861, 'wa': 1844, 'she': 1
703, ...})
```

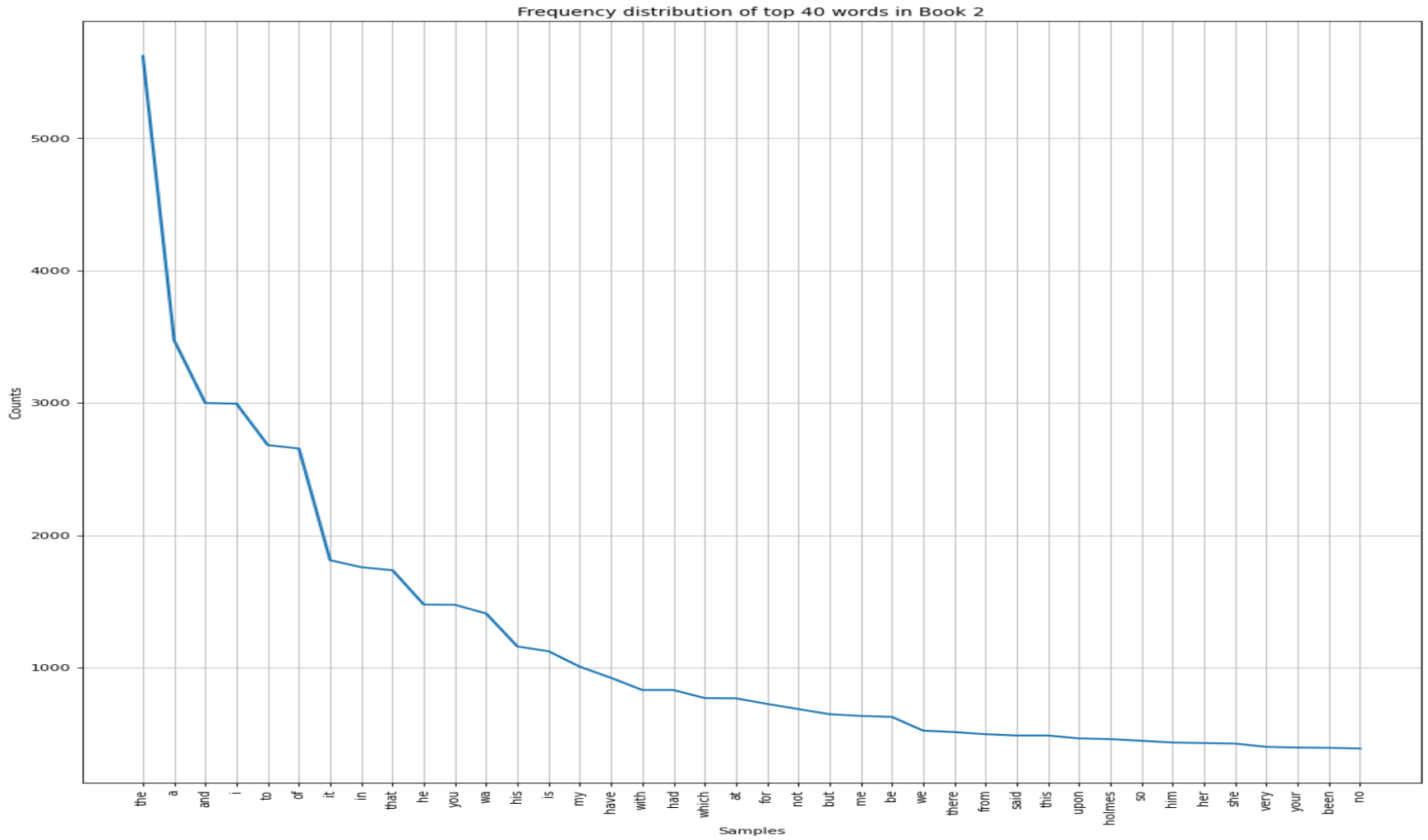
The *FreqDist* object for string T1

- For plotting the graph of frequency distribution we imported the function *figure()* from the module *matplotlib.pyplot* which is used to create a figure object. The whole figure is regarded as the figure object. Next we plotted frequency distribution graphs for both T1 and T2 which are as follows:



Inference from graph for Book 1 :

- We find that most frequent words in Book 1(Pride and Prejudice) are the, to, and, of, a etc., which are all stop words but we find that the word “elizabeth” is also amongst the top 40 most appeared words (631 times).
- A notable observation is that in T1 we find *her* has appeared 2000+ (2216) times
- Thus we can conclude that it is a female lead character driven book and Elizabeth is the main character in that book.
- In this frequency distribution we can note that stop words make up a substantial part of the total 121577 words.

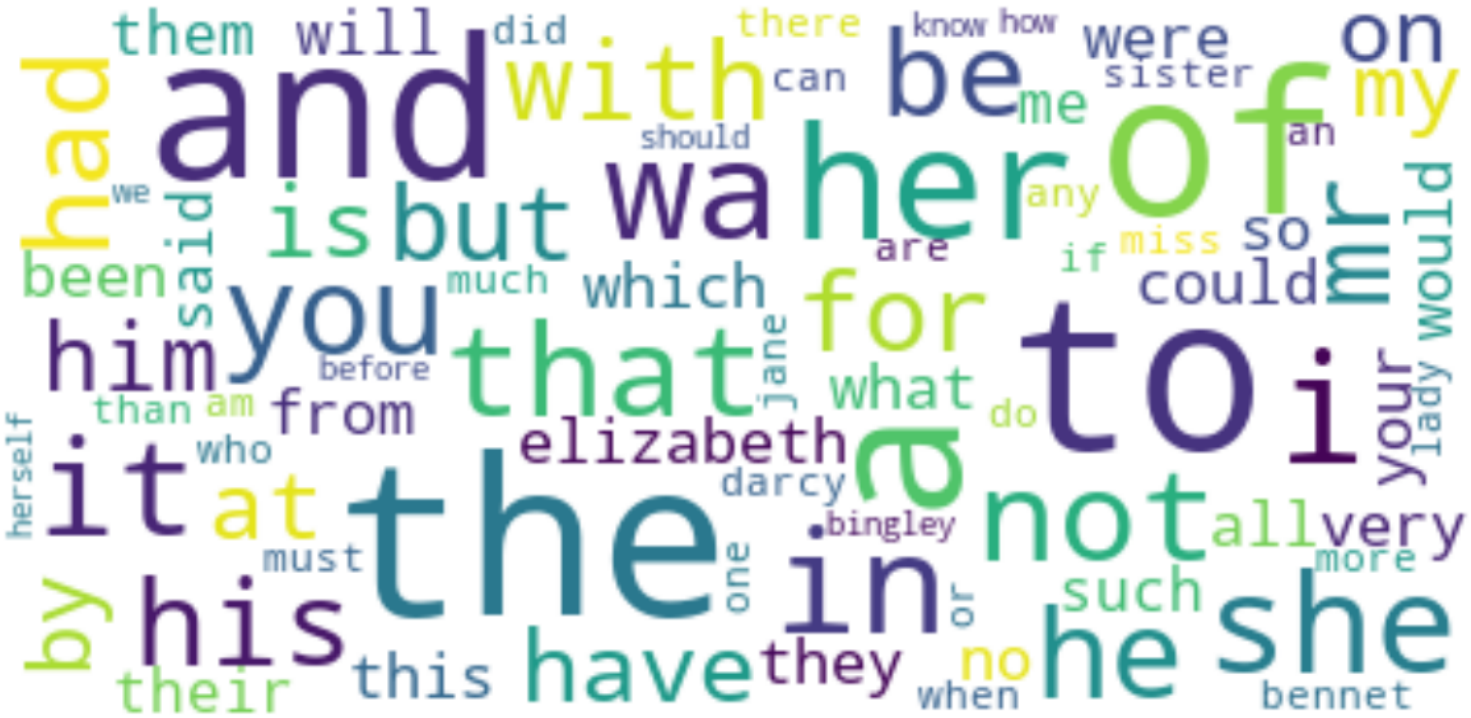


Inference from graph for Book 2 :

- In Book 2's distribution we can see *I* has a high frequency which shows that there are a lot of dialogues in the book.
- Also, in Book 2 *he*(1400+)*, his*(1200+) and *holmes*(400+) are present in the top 40 most frequent words.
- Thus we can conclude that it is a male lead character driven book and Holmes is the main character in that book.
- In this frequency distribution also we can note that stop words make up a substantial part of the total words.

Creating Word cloud :

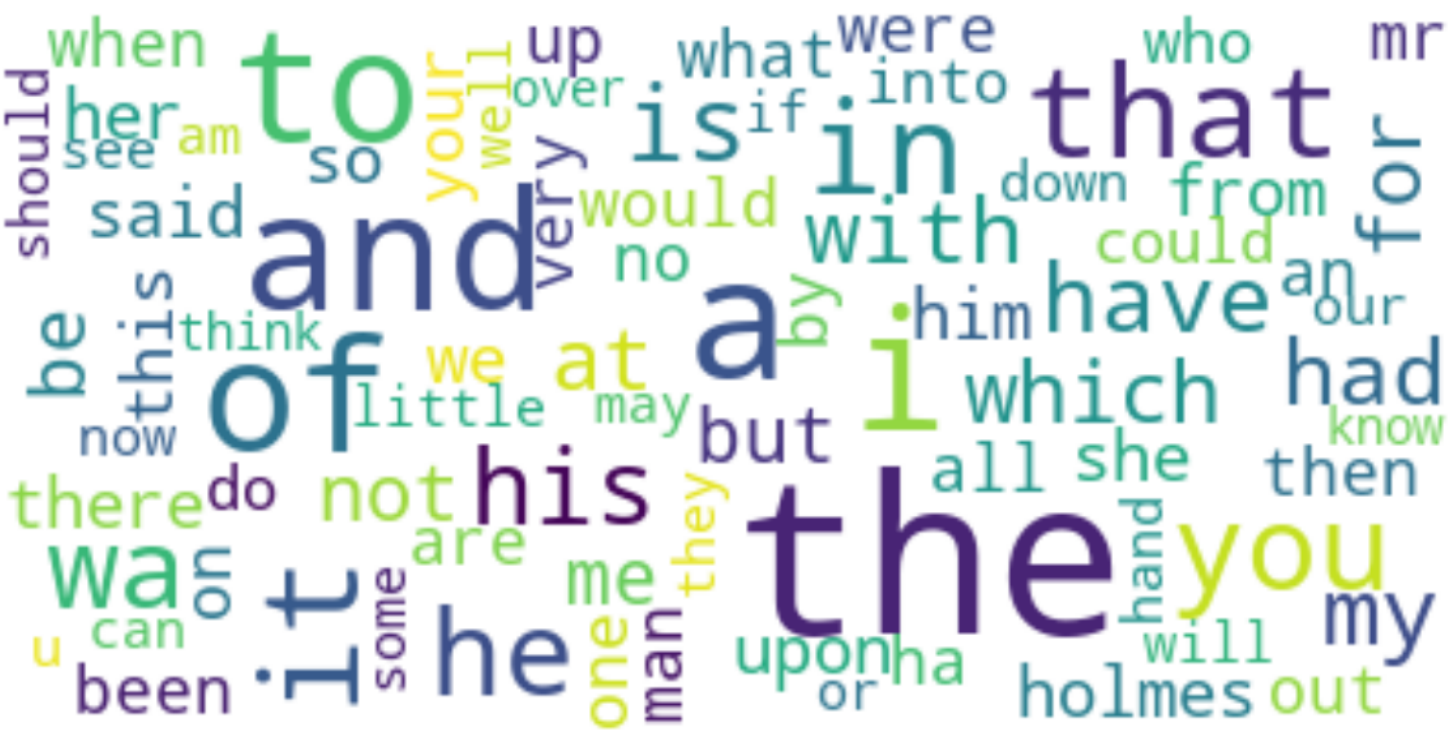
- For creating word cloud for both T1 and T2 we imported *Counter* from the module *Collections*. Then we imported *wordcloud* from the module *wordcloud*. Then we used functions *plt.figure() , plt.axis(), plt.show()* for the visualization of wordcloud. We made the word cloud for the top 80 most frequently used words which is attached below :



Word Cloud for Book 1 (with stop words)

Inference :

- It is giving the visual representation of the most frequent words in the book *Pride and Prejudice*.
- We observe that words like ***the, to, not, and*** are among the words that appear bigger and their frequency is also high in the previous plot.
- Infact, the overall word cloud is heavily dominated by stop words.
- Thus we can infer that the stop words occur in high frequency in the text.



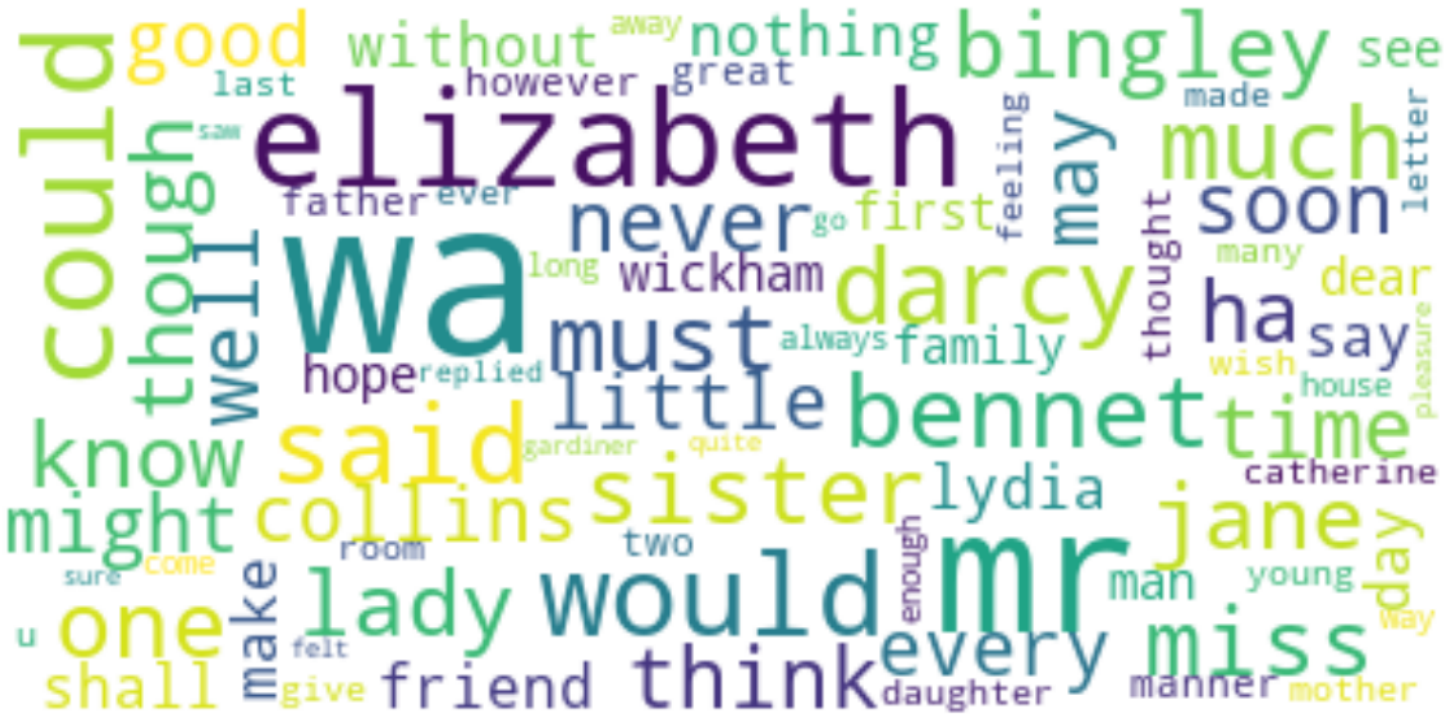
Word Cloud for Book 2 (with stop words)

Inference :

- It is giving the visual representation of the most frequent words in the book *The Adventures of Sherlock Holmes.*
- We observe that words like *the, and , his* are the among the words that appear bigger and their frequency is also high in the previous plot.
- Infact, the overall word cloud is heavily dominated by stop words.
- Thus we can infer that the stop words occur in high frequency in the text.

Removing stop words and creating word cloud again:

- We remove the stop words from the text. We used the *nltk.corpus stopwords* from English and removed them from the text. We updated the frequency distribution of words in the texts and repeated the above process to obtain a word cloud without stop words which is attached below :



Word Cloud for Book 1 (without stop words)

Inference :

- It is giving the visual representation of the most frequent words in the book *Pride and Prejudice* after removal of stopwords.
- We observe that words like *elizabeth, bennet, jane* and *bingley* are now amongst the words that appear bigger as their frequency is higher relatively after removal of stop words. These words are the names of the characters in the book.
- Words like *miss, lady, sister, daughter* etc. tell us about the dominant presence of female characters in the book.
- Words like *friend, family* etc. can be used to infer some sense of theme of the story.

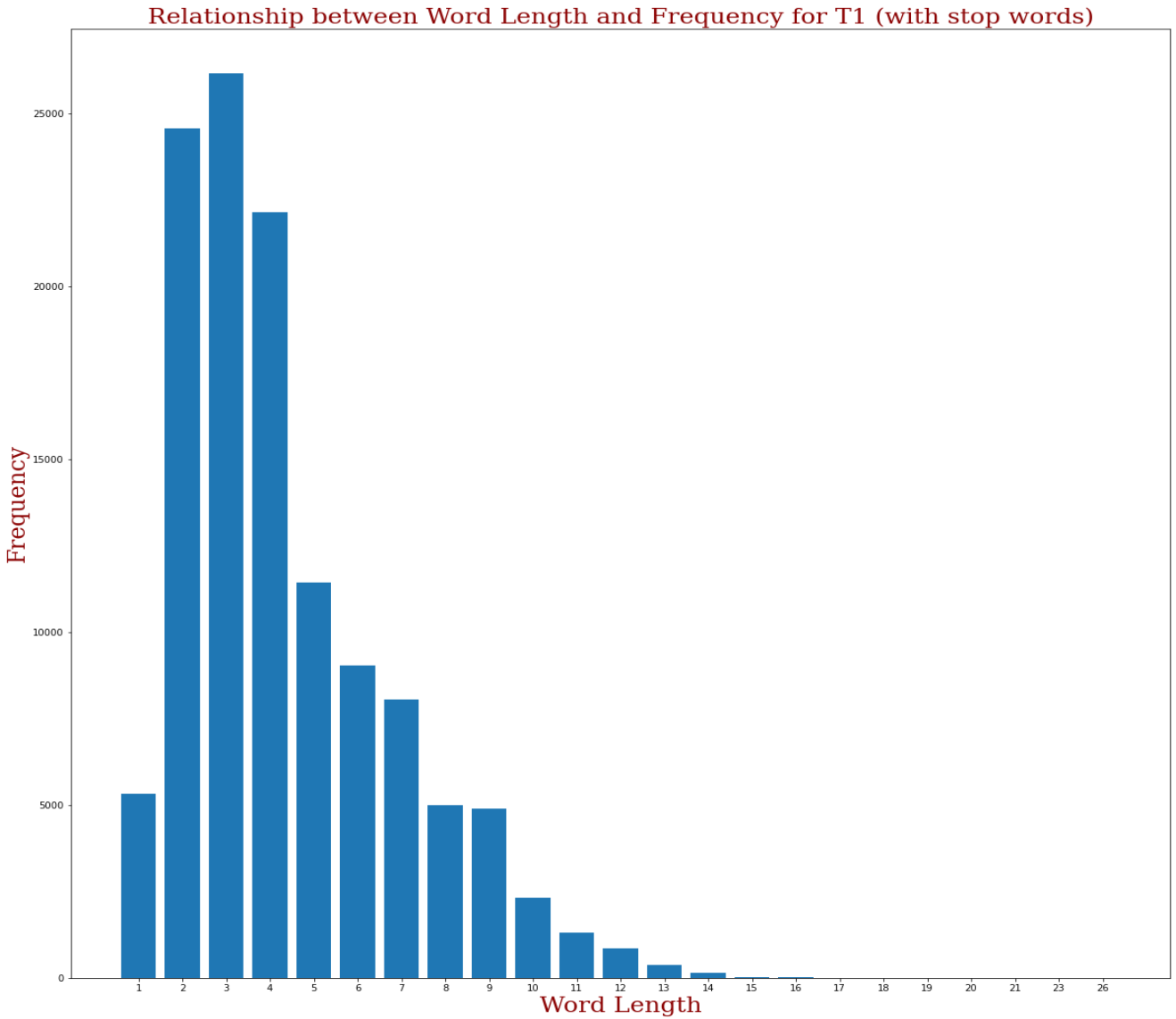
Word Cloud for Book 2 (without stop words)

Inference :

- It is giving the visual representation of the most frequent words in the book *The Adventures of Sherlock Holmes* after removal of stopwords.
- We observe that *holmes* has high relative frequency which confirms our belief of a character named Holmes in the book.
- Words like *said, shall, could, would* etc. suggest lots of dialogues in the book.

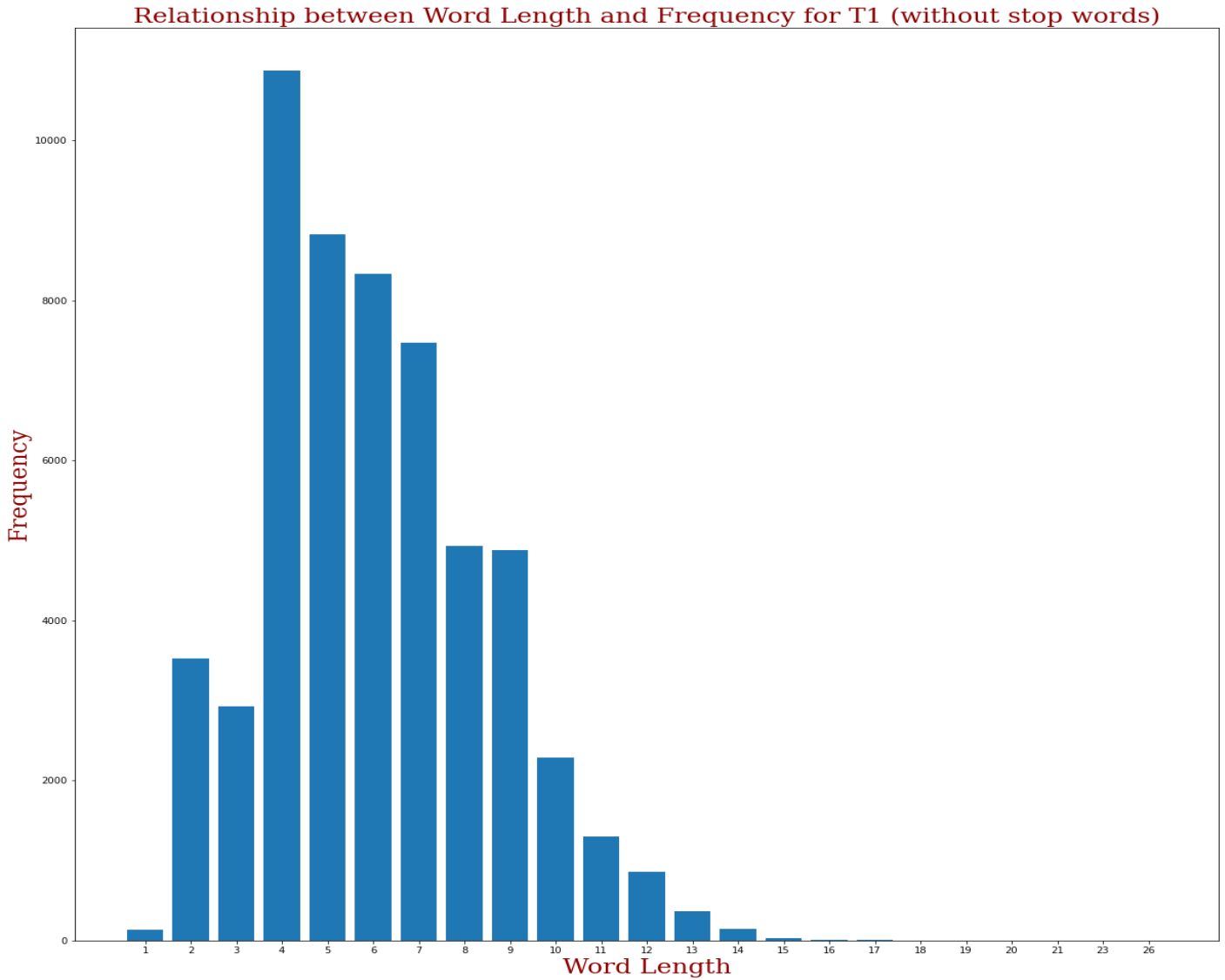
Relationship between the word length and frequency :

- We made an ordered dictionary to store the frequency of different word lengths in the text. The word lengths varied between 1 to 26. Next we plotted a bar chart showing the frequency of different word lengths. We did this for both the books twice, once with the stop words and once after removal of stop words. These plots are attached below :



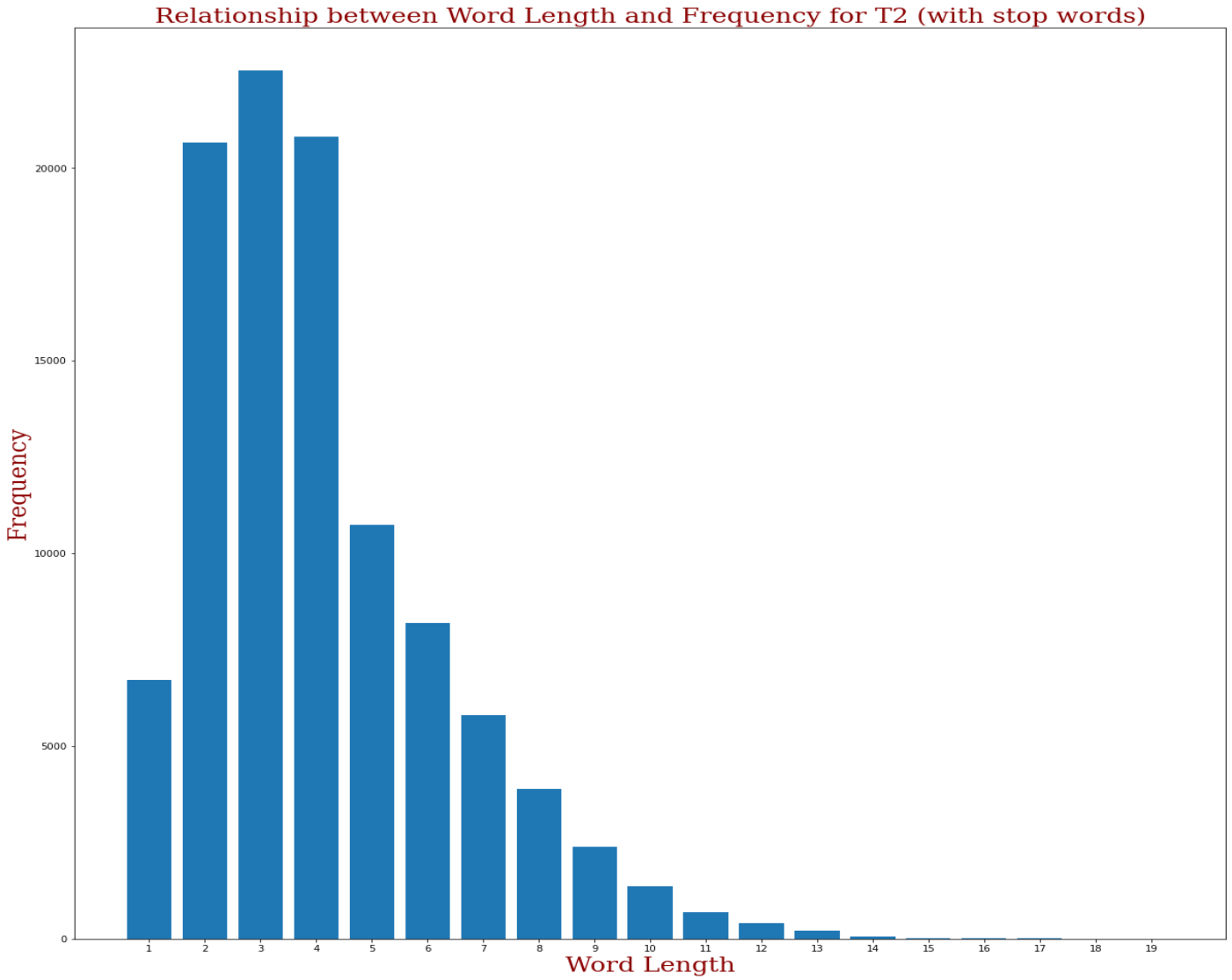
Inference :

- We have plotted a bar chart for the words of different length and their frequency. But in this chart we have included stop words, so the words of small length have large frequencies.
- Words of length 3 have the highest frequency.
- We can infer that words of lengths between 2 and 4 (inclusive) form the majority of the text.



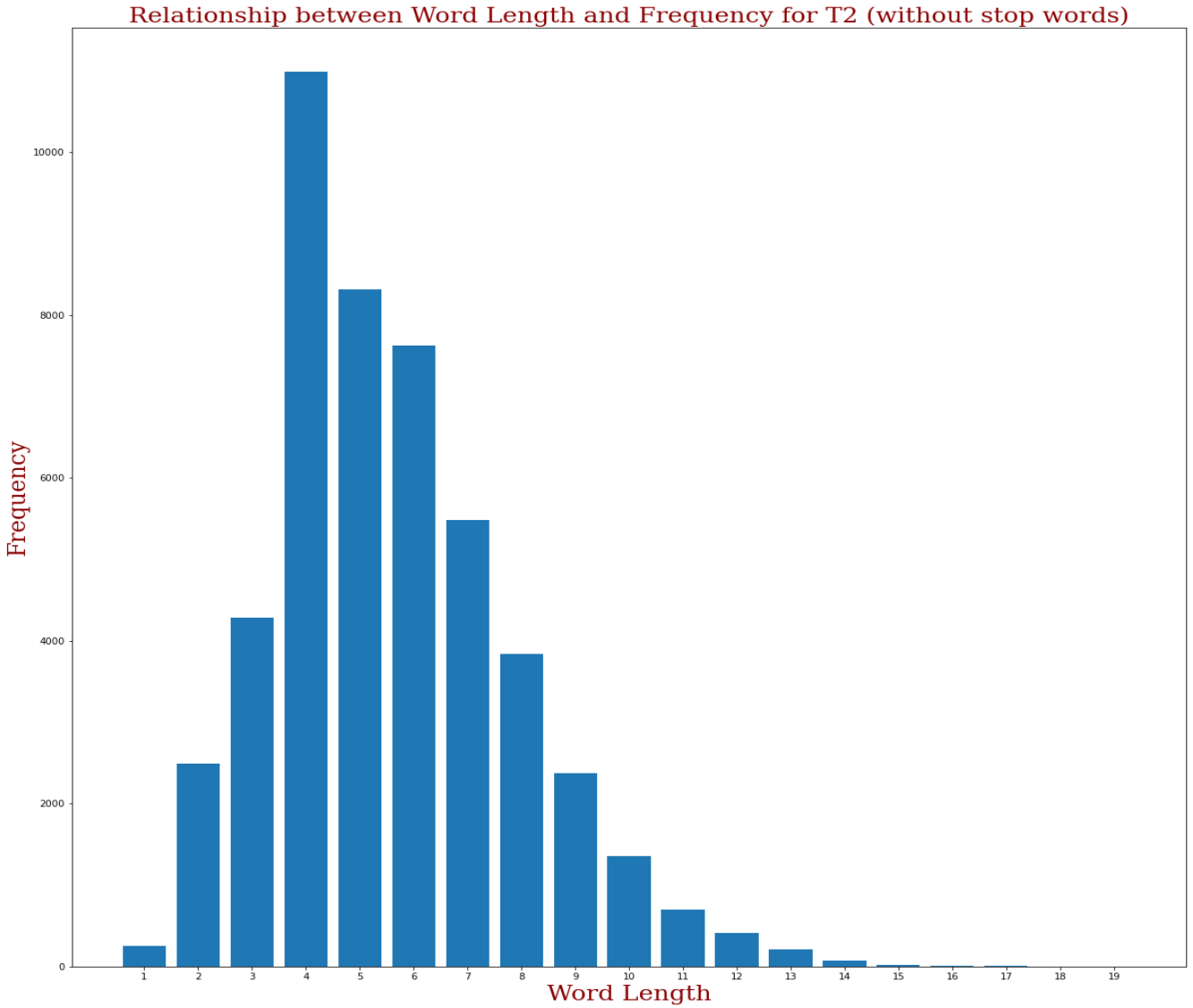
Inference :

- We have plotted a bar chart for the words of different length and their frequency. But for this chart we have not included stop words.
- Word length 4 has the highest frequency.
- We can infer that words of lengths between 4 and 7 (inclusive) form the majority of the text (after stop word removal).
- Comparing the above range with the chart for stop words included, we can also infer that stop words mainly have length between 2 and 3 (inclusive).



Inference :

- We have plotted a bar chart for the words of different length and their frequency. But in this chart we have included stop words, so the words of small length have large frequencies.
- Words of length 3 have the highest frequency.
- We can infer that words of lengths between 2 and 4 (inclusive) form the majority of the text.

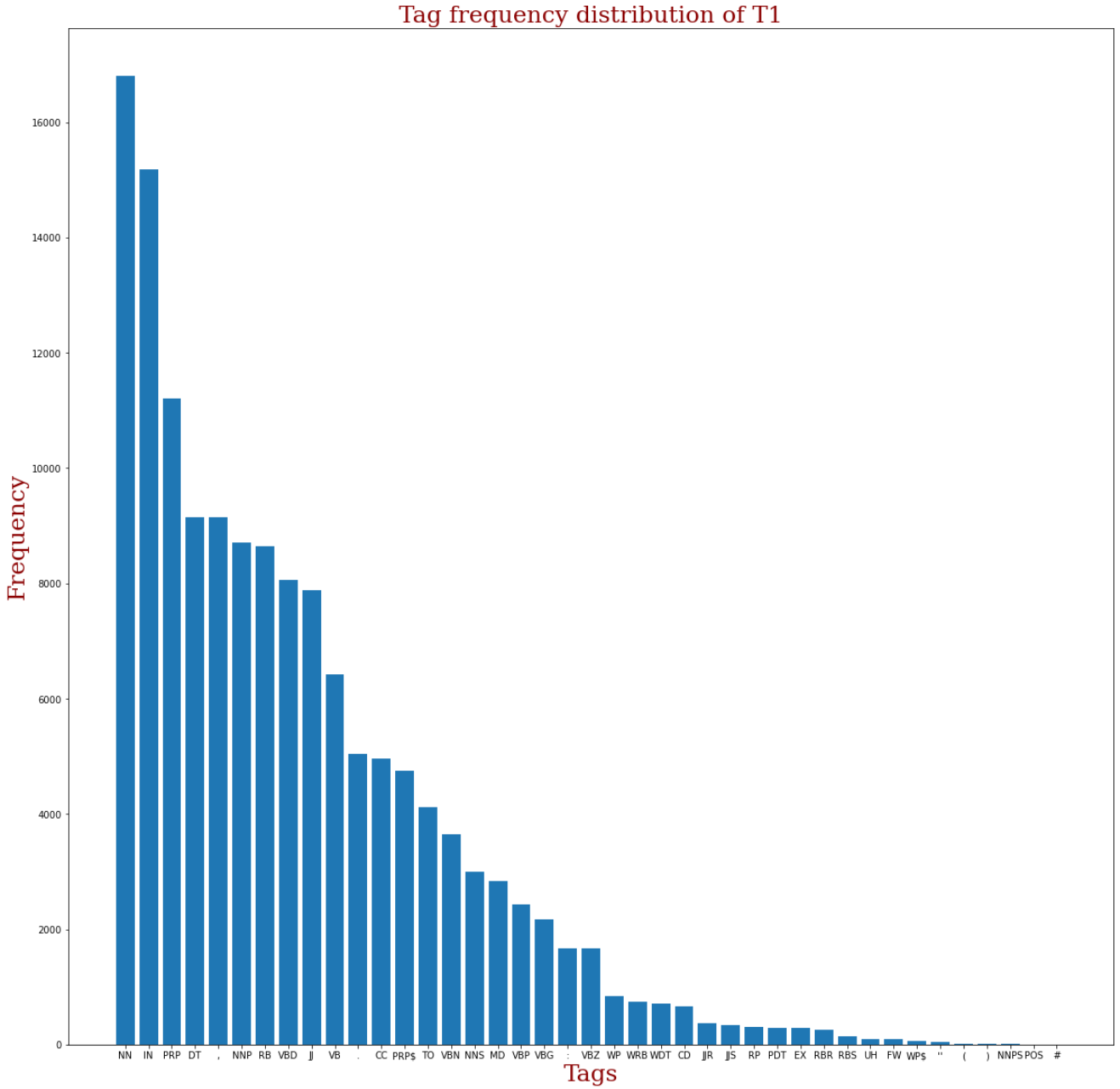


Inference :

- We have plotted a bar chart for the words of different length and their frequency. But for this chart we have not included stop words.
- Word length 4 has the highest frequency.
- We can infer that words of lengths between 4 and 6 (inclusive) form the majority of the text (after stop word removal).
- Comparing the above range with the chart for stop words included, we can also infer that stop words mainly have length between 2 and 3 (inclusive).
- Also note that the results are very similar for the two texts, meaning they may remain similar for all English texts.

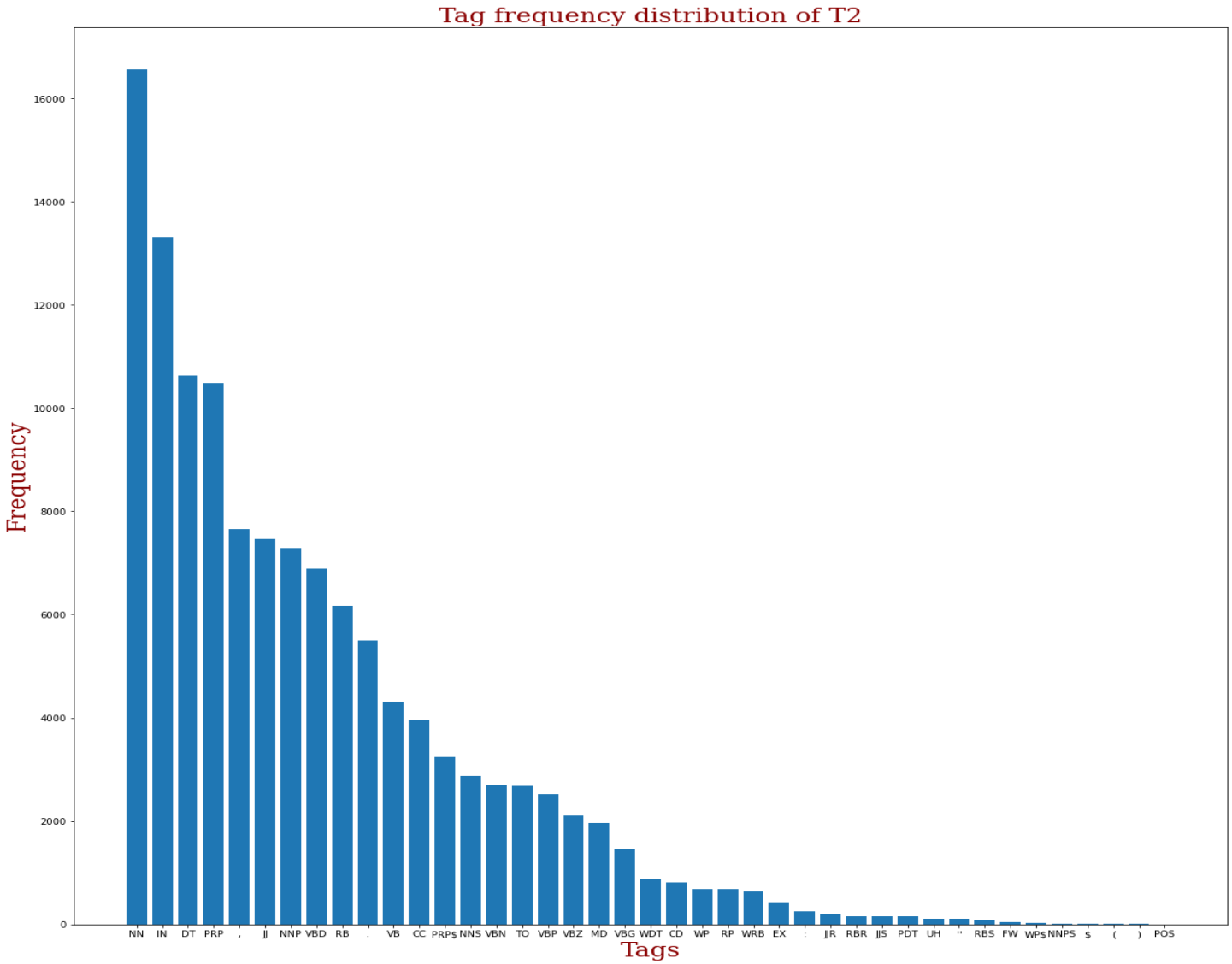
PoS Tagging and distribution of tags :

- For PoS tagging we used the original content of the eBooks without pre-processing it. We only removed the extra part to narrow down the string to the contents of the eBook. We used *nltk.pos-tag()* method for PoS tagging using the Penn Treebank Tagset. We word tokenized each sentence token before PoS tagging and then obtained the frequency distribution for the different tags for the texts. We plotted this distribution as a bar chart to show the frequency of occurrence of different PoS tags in the text.


Tag frequency distribution of T1
Frequency
Tags
0
2000
4000
6000
8000
10000
12000
14000
16000
NN IN PRP DT , NNP RB VBD JJ VB . CC PRP$ TO VBN NNS MD VBP VBG : VBZ WP WRB WDT CD JJR JJS RP PDT EX RBR RBS UH FW WP$ '' ( ) NNPS POS #

Inference :

- We find that most frequent are
  - NN : Nouns(singular) **(16798)**
  - IN : Preposition **(15176)**
  - PRP : Personal Pronoun **(11200)**
  - And thus we get the idea about the type of content written in the text file.



Inference :

- We find that most frequent are
  - NN : Nouns(singular) (16552)
  - IN : Preposition (13317)
  - DT : Determiner (10620)
  - And using this we can analyze the type of content that we see in this book .

- Also we can note that singular nouns and prepositions are among the most common Part of Speech tags for both the texts.

PROJECT ROUND - 2

1. INTRODUCTION

In this part of the Project we continue using the two large books imported in text format previously. We also use an additional third book in the last part of this Round of the project. We will perform PoS tagging on the data in the books, extract entities from the book and check the performance of this operation and finally we will use the third book and compare the books to find similarity if any. This will further be a good practical experience in the field of NLP.

As mentioned, two of the books remain the same as before. So the books used for this Round of project are the following-:

- First is **Pride and Prejudice** which is a novel written by **Jane Austen.**
- Second book is **The Adventures of Sherlock Holmes** which is a collection of twelve short stories by **Arthur Conan Doyle.**
- Third book is **Emma** by **Jane Austen**

For the tasks given in this project we have used *nltk* which is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language. And we have imported some modules and functions in order to perform different activities such as preprocessing, tokenizing, removing stop words, lemmatizing, information extraction etc. And after performing such operations on both of the text files we have plotted frequency distribution as required.

2. Python Libraries and Modules Used

Nltk.tokenizer package : Tokenizer divides strings into a list of substrings.

Nltk.stem package : Interfaces used to remove morphological affixes from words, leaving only the word stem.

Nltk.probability : A probability distribution specifies how likely it is that an experiment will have any given output.

Nltk.corpus : The modules in this package provide functions that can be used to read corpus files in a variety of formats.

Wordcloud package : Provides modules to create WordCloud in python.

Collection modules : Provide different types of container data types.

Nltk.stem.wordnet : It is a module used for stemming and lemmatization.

Sklearn.feature_extraction.text : This module can be used to extract features in a format supported by machine learning algorithms from datasets consisting of formats such as text and image.

3. METHODOLOGY

Downloading Books :

- We continued using the previous two books: **Pride and Prejudice** by **Jane Austen** and **The Adventures of Sherlock Holmes** by **Arthur Conan Doyle.** We further downloaded the following book in Plain Text UTF-8 format for text processing from Project Gutenberg (http://www.gutenberg.org)
    - **Emma** by **Jane Austen** (https://www.gutenberg.org/ebooks/158)

The downloaded file was a .txt file.

Cover of the new Book selected for Text Analysis

Importing the text :

In this step, we used the previously created function (*txt_file_to_string)* to read the text imported from both the books and convert them into string for

processing in python. This function takes as input the path of the file to be read and returns the content of the file in string format. We stored our first book “Pride and Prejudice” in the variable Original_B1 of string type. Similarly, we stored the text from the second book “The Adventures of Sherlock Holmes” in the variable Original_B2 of string type.

Text Preprocessing and Tokenization

- **Removing prefix and suffix to narrow down to text from Book -** Removing prefix and suffix to narrow down to text from eBook - Each book from the website had additional prefix and suffix in its .txt file, apart from the contents of the eBook. To narrow down our string to the relevant part only we considered only the substring of the original string which was marked with *** *START OF THE PROJECT* and *** *END OF THE PROJECT* in the .txt files.
- **Expanding some Contractions -**We expanded some generic contractions. For example: *can't* to *can not,* all instances of *'ll* to *will,* etc. This is not very accurate and will lead to some incorrect expansion since disambiguation to the right expansion is not deterministic but this will be correct for most cases.
- **Expanding more Contractions according to general assumption -** For example: ‘ve to have, ‘t to not, ‘m to am, etc.
- **Removing chapter number headings if any -** We remove the words named chapter because it increases the frequency of word chapter present in the text unnecessarily.
- **Replacing one or more continuous whitespace characters by space -** Replacing one or more continuous white space characters with a single space to make the string evenly spaced.

POS_Tagging

- We created a function to perform POS_Tagging of the text imported from the books.
- **What is POS_tagging -** It is the categorizing of words in the text with particular correspondence with the part of speech depending on the definition of the word and its context.
- For this we first sentence tokenized the entire text, then we word tokenized each sentence and finally we did POS Tagging for the words.
- We mainly used the *nltk.tokenize* library for this.
- **Function which extracts and returns lists of nouns and verbs in the Book whose pos_tag is given as parameter -** We created a function(*extract_nouns_and_verbs*) to find out the nouns and verbs present in the text with the help of their POS tags. We also find the categories that these words fall under in WordNet.
- To extract the nouns and verbs we first normalized the words in the tagged list above. In this process we change the words to lowercase and remove punctuations from it. Then we find the wordnet category for the word using its POS tag. Using this category we lemmatize the word using the *WordNetLemmatizer* and append the list in the proper list if it is a noun or a verb.

- We identify the word as a noun if its POS tag starts with an ‘N’, Similarly for a verb the tag starts with a ‘V’. This is true for the 36 Tags present in the **Penn Treebank Tagset** which we have used for tagging in our code.

  The noun tags in the Tagset are -

NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural

  The verb tags in the Tagset are -

VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present

  Thus we can see that our observation of identifying categories by using the first letter of the Tag is correct for this tagset.

- **Frequency Distribution Plots and Interpretation -** Next we plot the frequency distribution of the nouns and the verbs in the text under

different categories as per WordNet.

- The categories for nouns are as follows:

noun.Tops	unique beginner for nouns
noun.act	Nouns denoting acts or action
noun.animal	Nouns denoting animal
noun.artifact	nouns denoting man-made objects
noun.attribute	nouns denoting attributes of people and objects
noun.body	Nouns denoting body parts
noun.cognition	nouns denoting cognitive processes and contents
noun.communication	nouns denoting communicative processes and contents
noun.event	nouns denoting natural events
noun.feeling	nouns denoting feelings and emotions
noun.food	nouns denoting foods and drinks
noun.group	nouns denoting groupings of people or objects
noun.location	nouns denoting spatial position
noun.motive	Nouns denoting goals
noun.object	nouns denoting natural objects (not man-made)
noun.person	Nouns denoting people

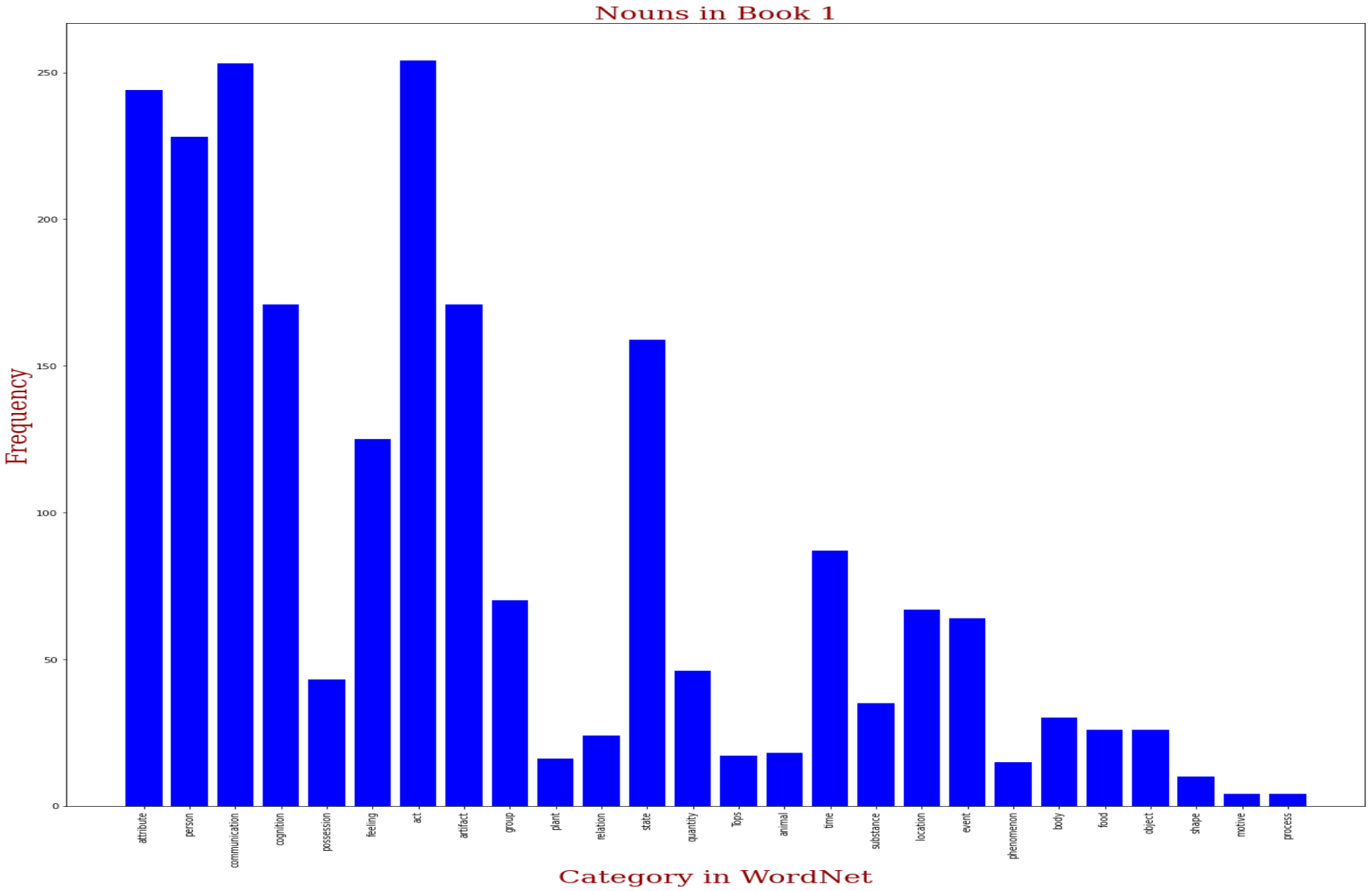
noun.phenomenon	nouns denoting natural phenomena
noun.plant	Nouns denoting plants
noun.possession	nouns denoting possession and transfer of possession
noun.process	nouns denoting natural processes
noun.quantity	nouns denoting quantities and units of measure
noun.relation	nouns denoting relations between people or things or ideas
noun.shape	nouns denoting two and three dimensional shapes
noun.state	nouns denoting stable states of affairs
noun.substance	nouns denoting substances
noun.time	nouns denoting time and temporal relations

- The categories for verbs are as follows:

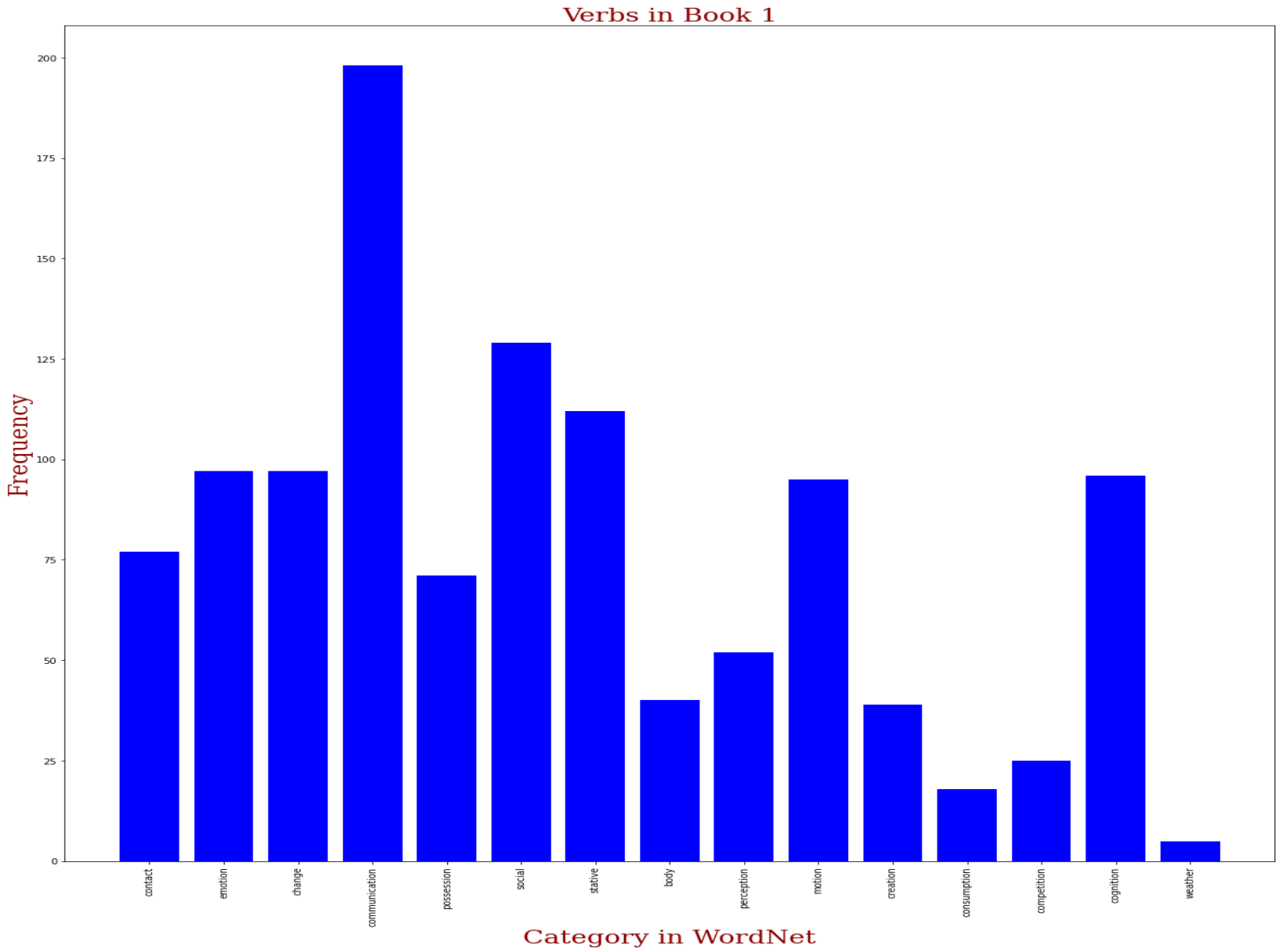
verb.body	verbs of grooming, dressing and bodily care
verb.change	verbs of size, temperature change, intensifying, etc.
verb.cognition	verbs of thinking, judging, analyzing, doubting
verb.communication	verbs of telling, asking, ordering, singing

verb.competition	verbs of fighting, athletic activities
verb.consumption	verbs of eating and drinking
verb.contact	verbs of touching, hitting, tying, digging
verb.creation	verbs of sewing, baking, painting, performing
verb.emotion	verbs of feeling
verb.motion	verbs of walking, flying, swimming
verb.perception	verbs of seeing, hearing, feeling
verb.possession	verbs of buying, selling, owning
verb.social	verbs of political and social activities and events
verb.stative	verbs of being, having, spatial relations
verb.weather	verbs of raining, snowing, thawing, thundering

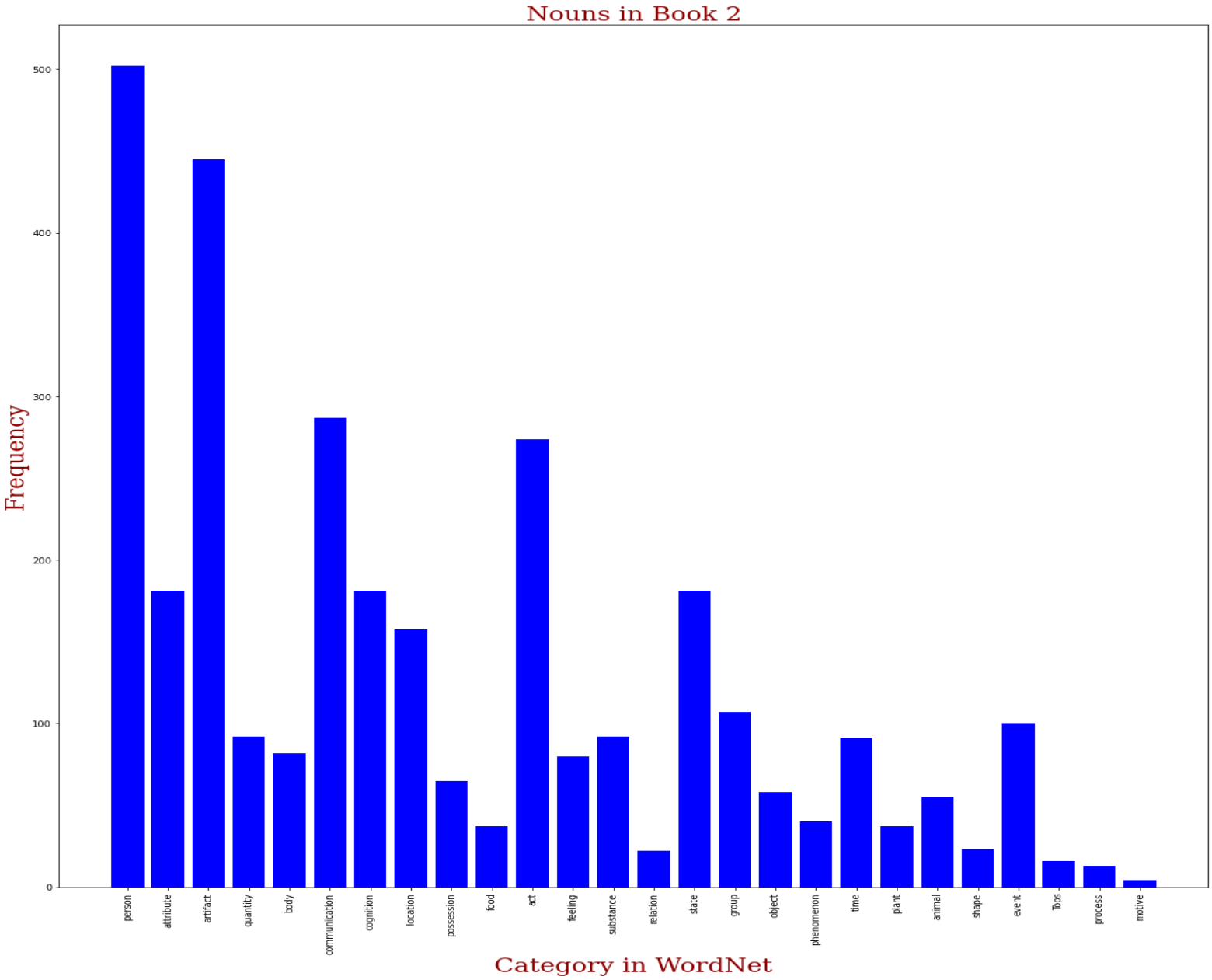
- When we plot the graph of the frequency distribution of nouns in the Book1- *Pride and Prejudice*, where we plot *Category in Wordnet* on the x-axis and we plot *frequency* for the category on the y-axis, we find that some of the most frequent noun categories are: act, attribute, communication, person



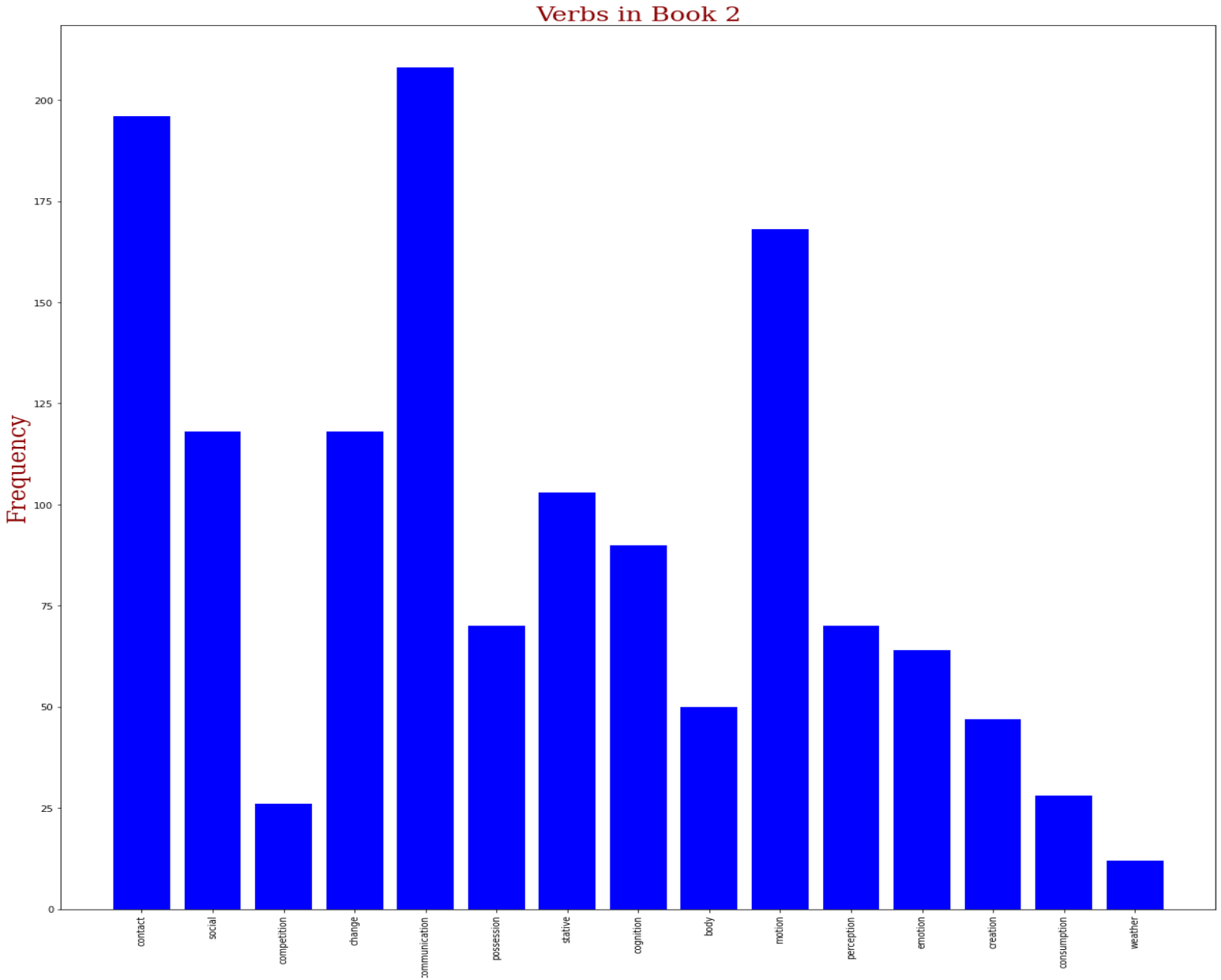
- When we plot the graph of the frequency distribution of the verbs in the book1 - *Pride and Prejudice*. Where we plot *Category in Wordnet* on the x-axis and we plot *frequency* for the category on the y-axis, we find that some of the most frequent verb categories are: communication, social.



- When we plot the graph of the frequency distribution of the nouns in the book2 - *The Adventures of Sherlock Holmes*, where we plot *Category in Wordnet* on the x-axis and we plot *frequency* for the category on the y -axis, we find that some of the modest frequent nouns are: person, artifact.



- When we plot the graph of the frequency distribution of the verbs in the book2 - *The Adventures of Sherlock Holmes*,, where we plot *Category in Wordnet* on the x-axis and we plot frequency on the y- axis, we find that some of the most frequent verbs are: Contact, motion, communication.



Extracting Entities:

- To identify all Persons, Location, and organisations in the novel we have used Named Entity Recognition (NER) in python. We used the nltk.ne_chunk function to do this.

- We pre-processed the text as above for this purpose, that is, we only reduced the text to the content of the book, expanded contractions, and removed chapter numbers from the text of the book.

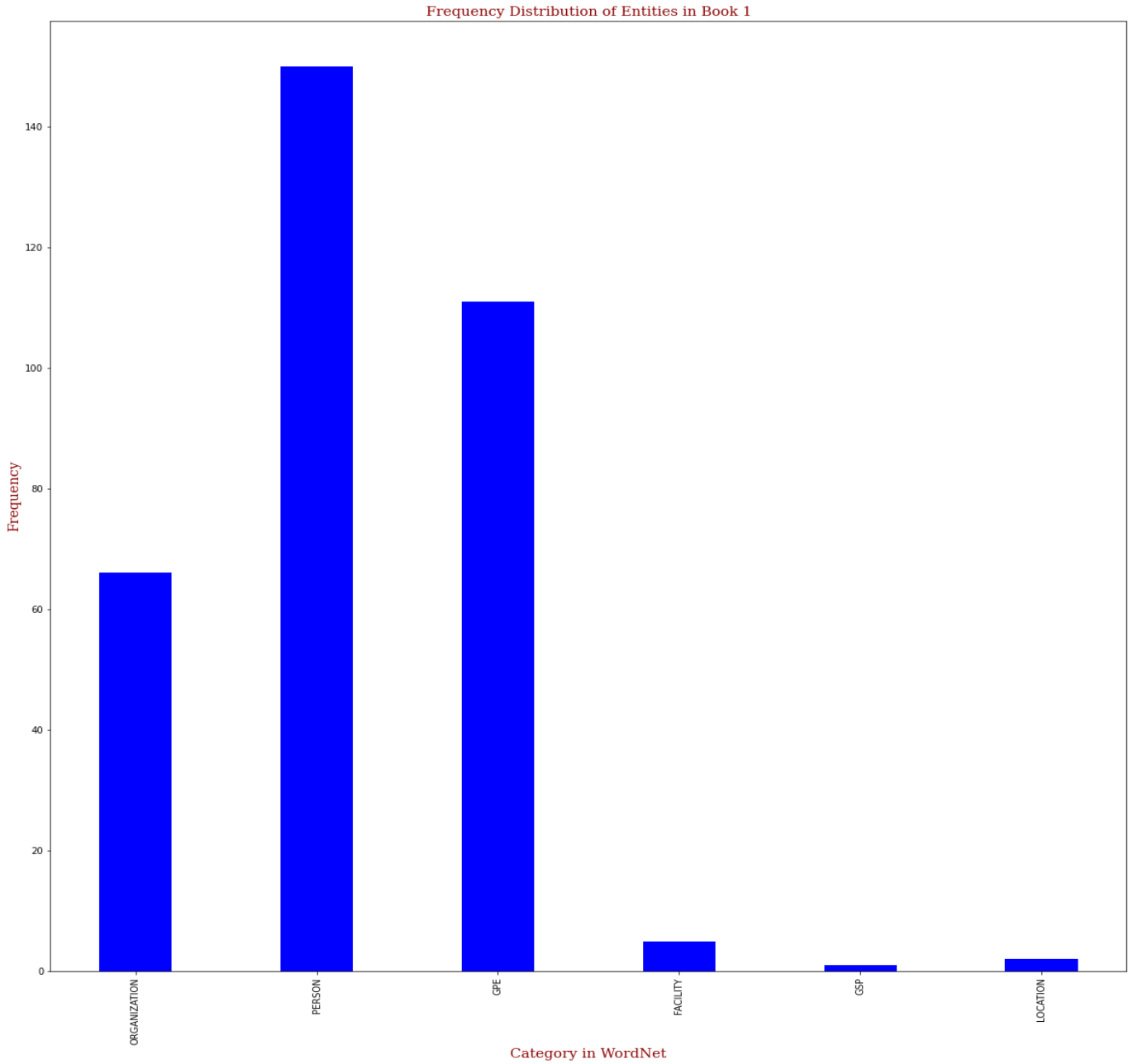
- Then we sentence tokenized the text using nltk.sent_tokenize(). We iterated over each sentence, word tokenizing it using nltk.word_tokenize() and found the part of speech tag for the words using nltk.pos_tag().

- Then we used the nltk.ne_chunk() function to chunk the list of tagged tokens. If the word was identified as an entity and was being encountered for the first time we inserted its label into the list entity_list.

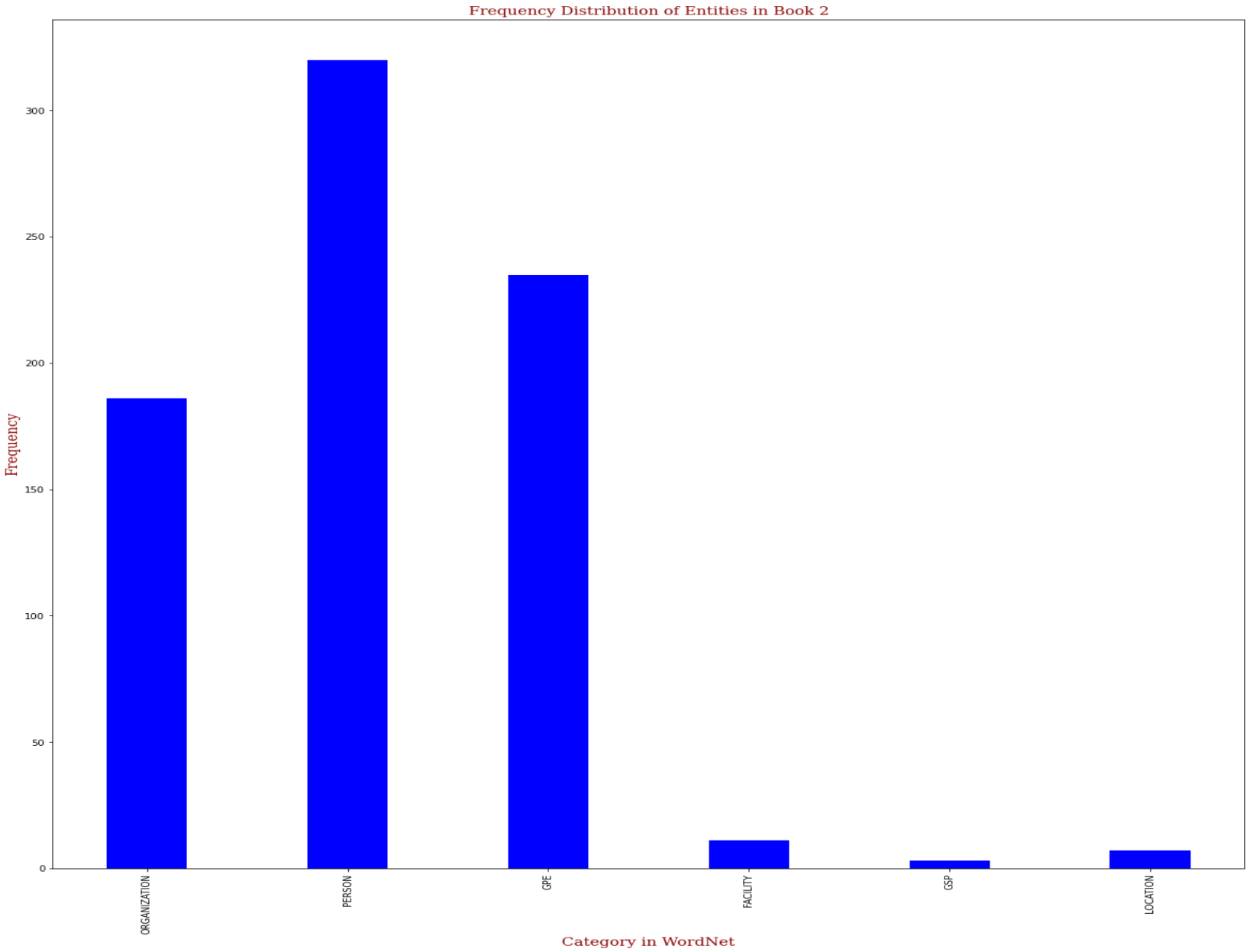
- The bar graph plotted using the FreqDist() of the above list was then used to indicate the number of different types of entities in the text.

- The named entity types reported are : ORGANIZATION, PERSON, LOCATION, GPE(Geo-Political Entity), FACILITY and GSP(Geographical-Social-Political Entity).

- The bar graphs are attached:



When we plot the graph of the frequency distribution of the entities in the book *The Adventures of Sherlock Holmes*, where we plot *Entity type* on the x-axis and we plot frequency on the y- axis, we find that some of the most frequent entity is of type: person, least frequent is of type: GSP



When we plot the graph of the frequency distribution of the entities in the book *Pride and Prejudice*, where we plot *Entity type* on the x-axis and we plot frequency on the y-axis, we find that some of the most frequent entity is of type: person, least frequent is of type : GSP.

Performance Evaluation of NER:

- We selected 6 random passages from the two books (3 from each) and manually identified the entities present in them.
- We then also extracted the entities from these passages using the nltk library as stated above.
- The passages can be found in the code.
- For example the manual labelling for passage 2 of *Pride and Prejudice* was as follows:

Passage 2 (Pride and Prejudice)

Miss Bingley's attention was quite as much engaged in watching Mr. Darcy's progress through *his* book, as in reading her own; and she was perpetually either making some enquiry, or looking at his page. She could not win him, however, to any conversation; he merely answered her question, and read on. At length, quite exhausted by the attempt to be amused with her own book, which she had only chosen because it was the second volume of his, she gave a great yawn and said, "How pleasant it is to spend an evening in this way! I declare after all there is no enjoyment like reading! How much sooner one tires of anything than of a book! When I have a house of my own, I shall be miserable if I have not an excellent library."No one made any reply. She then yawned again, threw aside her book, and cast her eyes round the room in quest for some amusement; when hearing her brother mentioning a ball to Miss Bennet, she turned suddenly towards him and said: "By the bye, Charles, are you really serious in meditating a dance at Netherfield? I would advise you, before you determine on it, to consult the wishes of the present party; I am much mistaken if there are not some among us to whom a ball would be rather a punishment than a pleasure." "If you mean Darcy," cried her brother, "he may go to bed, if he chooses, before it begins—but as for the ball, it is quite a settled thing; and as soon as Nicholls has made white soup enough, I shall send round my cards.""I should like balls infinitely better," she replied, "if they were carried on in a different manner; but there is something insufferably tedious in the usual process of such a meeting. It would surely be much more rational if conversation instead of dancing were made the order of the day.""Much more rational, my dear Caroline, I dare say, but it would not be near so much like a ball."

Persons : Miss Bingley, Mr. Darcy, Charles, Miss Bennet, Darcy, Nicholls, Caroline
Location : Netherfield
Organization : -----

	Predicted Positive	Predicted Negative
Actual Positive	8	0
Actual Negative	1	0

Precision (p) = 0.888
Recall (r) = 1
F1 Measure = 0.941

- The precision, recall and F1 measure values for the 6 passages are given in the table below:

Passage Number	Precision	Recall	F1 Measure
1 (Book 1)	0.75	0.43	0.55
2 (Book 1)	0.89	1.00	0.94
3 (Book 1)	0.66	0.80	0.73
4 (Book 2)	0.57	0.57	0.57
5 (Book 2)	0.60	0.38	0.37
6 (Book 2)	0.64	0.70	0.67

- The average values of above measures are as follows:

  Average Precision = 0.68
  Average Recall = 0.65
  Average F1 Measure = 0.64

- We can see that the nltk entity recognition is not very accurate, however its performance is quite decent.

Creating TF-IDF Vectors and Calculating Cosine Similarities:

- We first read the data from the third book *Emma* by *Jane Austen*. Then we pre-process this data as we did earlier for the two other books. We further also lower the case for all letters in all the books and remove punctuations from all of them.
- We then create a container of all the three files we have to read.
- The books in order are - **Emma** (Book1), **Pride and Prejudice** (Book2) and **The Adventures of Sherlock Holmes** (Book3).
- This order is due to the files being sorted by name in the drive.
- We append the string format of the files in a list and store it to make an index of tf-idf vectors.
- We then use the *TfidfVectorizer* from *sklearn.feature_extraction.text* to create tf-idf vectors for the three books.
- Then we convert the sparse matrix format to proper vector format for cosine similarity calculation.
- We then find the cosine similarity for different pairs as follows:

    Book1 & Book2 - 0.9675365224177660
    Book1 & Book3 - 0.9072784374389796
    Book2 & Book3 - 0.9202425576369526

- Then we lemmatize the text of each of the books.
- We again create tf-idf vectors and find the cosine similarities. The results for different pairs this time were:

    Book1 & Book2 - 0.9672514003773120
    Book1 & Book3 - 0.9062943952161336
    Book2 & Book3 - 0.9193122701206012

- We can see that the cosine similarity values change very slightly after lemmatization.
- We can also clearly see that the cosine similarity for the two books written by the same author **Jane Austen** (Book1 & Book2) have higher similarity values than for any other pair.

CONCLUSION

Working on this project was a great learning experience for us in understanding the subject as well as team coordination. We all had surface-level knowledge about all the processes in text analytics but this project has helped us gain a better understanding of text processing using NLP techniques in python. Using python libraries and in-built toolkits, we came to a conclusion that this project highlights the basic understanding of text preprocessing, PoS tagging, Tokenization,extracting particular part of speech tagged words such as nouns and verbs in this project , tf-idf vectors , cosine similarity to compare the similarity between two books , NER etc. The Graphs, Bar Charts, Word clouds, frequency distribution tables represented by using matplotlib helped us more in understanding the output and it is also beneficial for the visual representation of the data. Overall this was a great learning experience and it has encouraged us to explore more in the fields of NLP.

REFERENCES

- **NLP Class Lectures and Slides**
- https://www.researchgate.net/publication/319164243_Natural_Language_Processing_State_of_The_Art_Current_Trends_and_Challenges
- http://librarycarpentry.org/lc-tdm/index.html
- https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/
- https://www.mygreatlearning.com/blog/nltk-tutorial-with-python/#3
- https://www.geeksforgeeks.org/text-analysis-in-python-3/
- https://www.nltk.org/book/ch05.html
- https://www.nltk.org/api/nltk.chunk.html?highlight=ne_chunk#nltk.chunk.ne_chunk