# 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**

LNMIIT

The LNM Institute of
Information Technology

**Github Repository Link -:** [https://github.com/DragoPhoenix/NLP-Project](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 **Text Analysis [NLP Project Round -1]** which also helped us in doing a lot of research and it was a great learning experience.

**Date : 20 Nov, 2021**

**Aniket Shukla ( 19UCS057 )**
**Anushka Vyas ( 19UCS041 )**
**Harsh Kumar ( 19UCS185 )**

# TABLE OF CONTENTS

TITLE

# 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, and various other operations on two of the very well-known books ***Pride and Prejudice*** and ***The Adventures of Sherlock Holmes***. Working on this project on these two books was particularly interesting because they are both 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. The project also described 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.

# 1. 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.

# 2. 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.

# 3. 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.

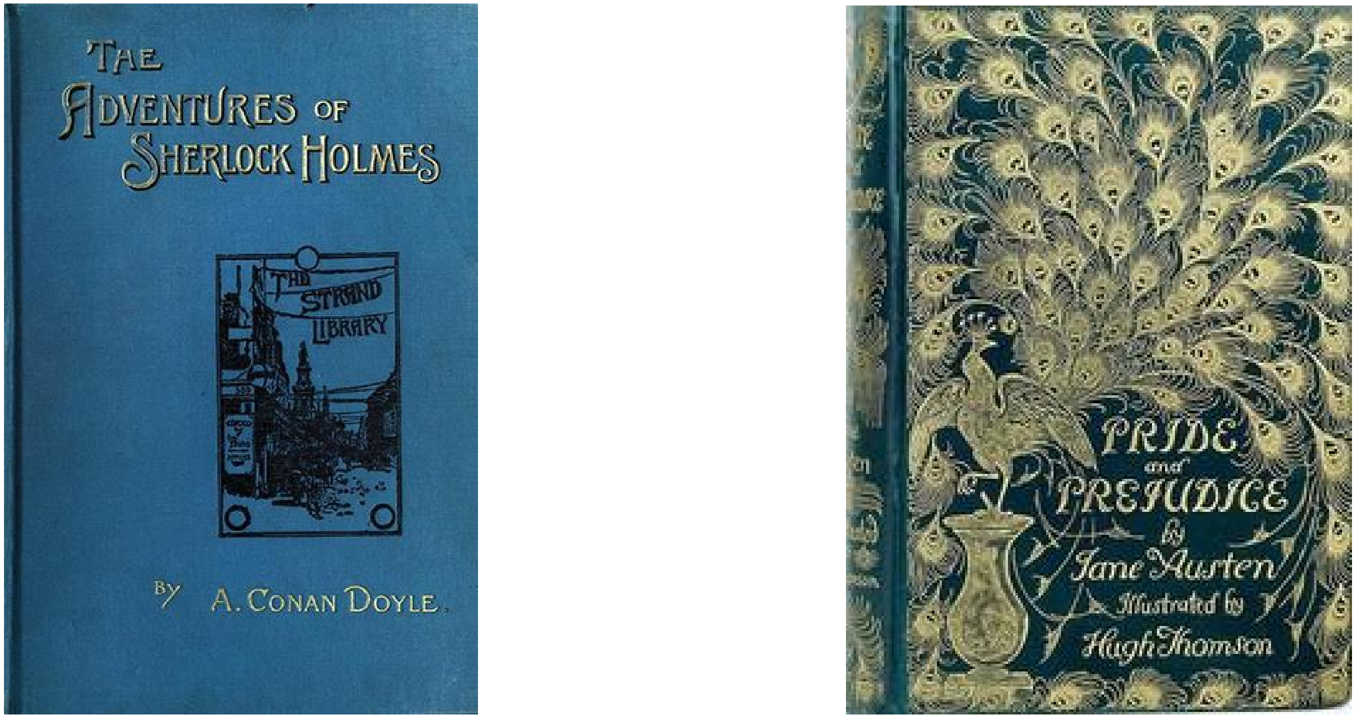
**Collection modules :** Provide different types of container data types.

# 4. 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)

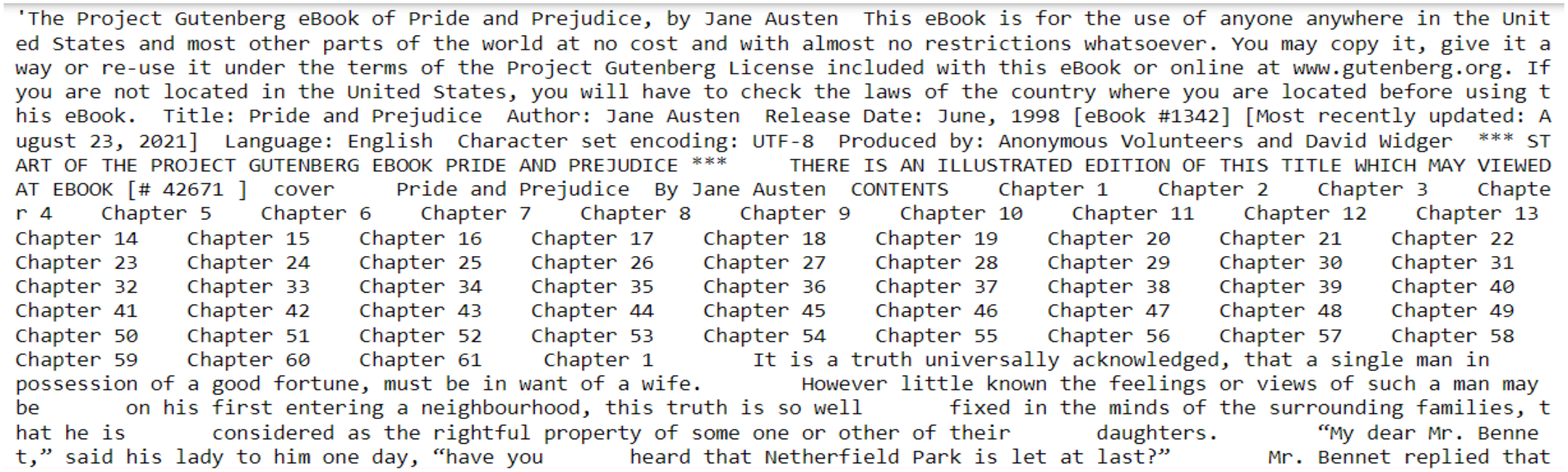
The downloaded files were .txt files.


THE ADVENTURES OF SHERLOCK HOLMES
THE STRAND LIBRARY
BY A. CONAN DOYLE.
PRIDE and PREJUDICE by Jane Austen
Illustrated by Hugh Thomson

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.



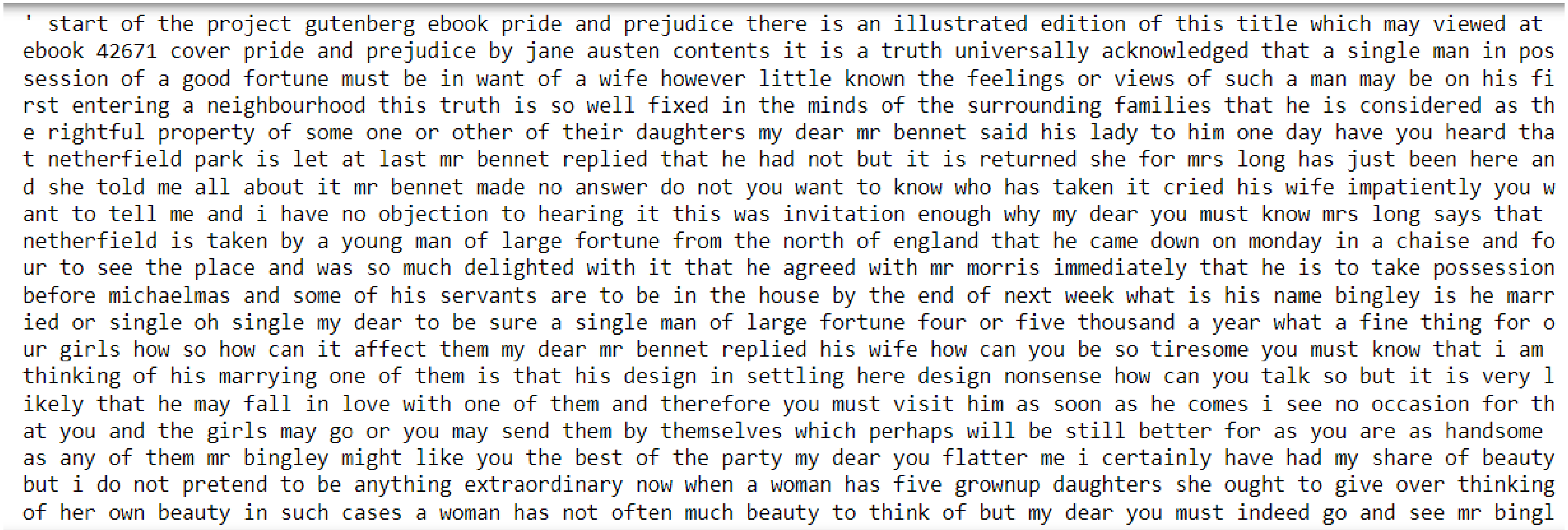
```
'The Project Gutenberg eBook of Pride and Prejudice, by Jane Austen  This eBook is for the use of anyone anywhere in the Unit
ed States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it a
way or re-use it under the terms of the Project 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 :

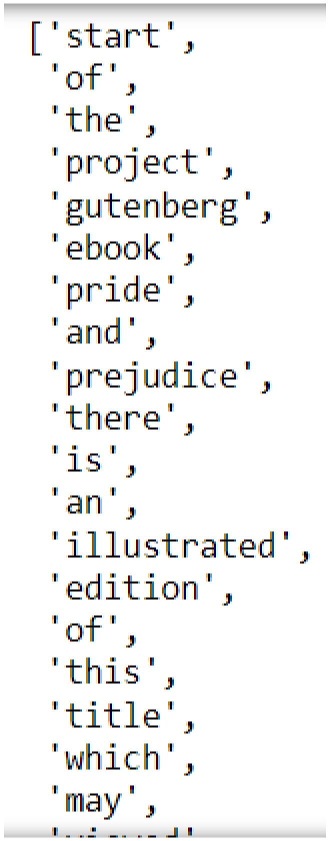
1. Removing prefix and suffix to narrow down to text from eBook - Each book from the website had additional prefix and suffix in its .txt file, apart from the contents of the eBook. To narrow down our string to the relevant part only we considered only the substring of the original string which was marked 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

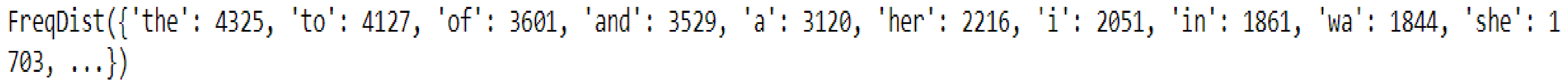


```
['start',
 'of',
 'the',
 'project',
 'gutenberg',
 'ebook',
 'pride',
 'and',
 'prejudice',
 'there',
 'is',
 'an',
 'illustrated',
 'edition',
 'of',
 'this',
 'title',
 'which',
 'may',
```

The list of tokenized and lemmatized words of T1

## Plotting Frequency distribution of tokens :

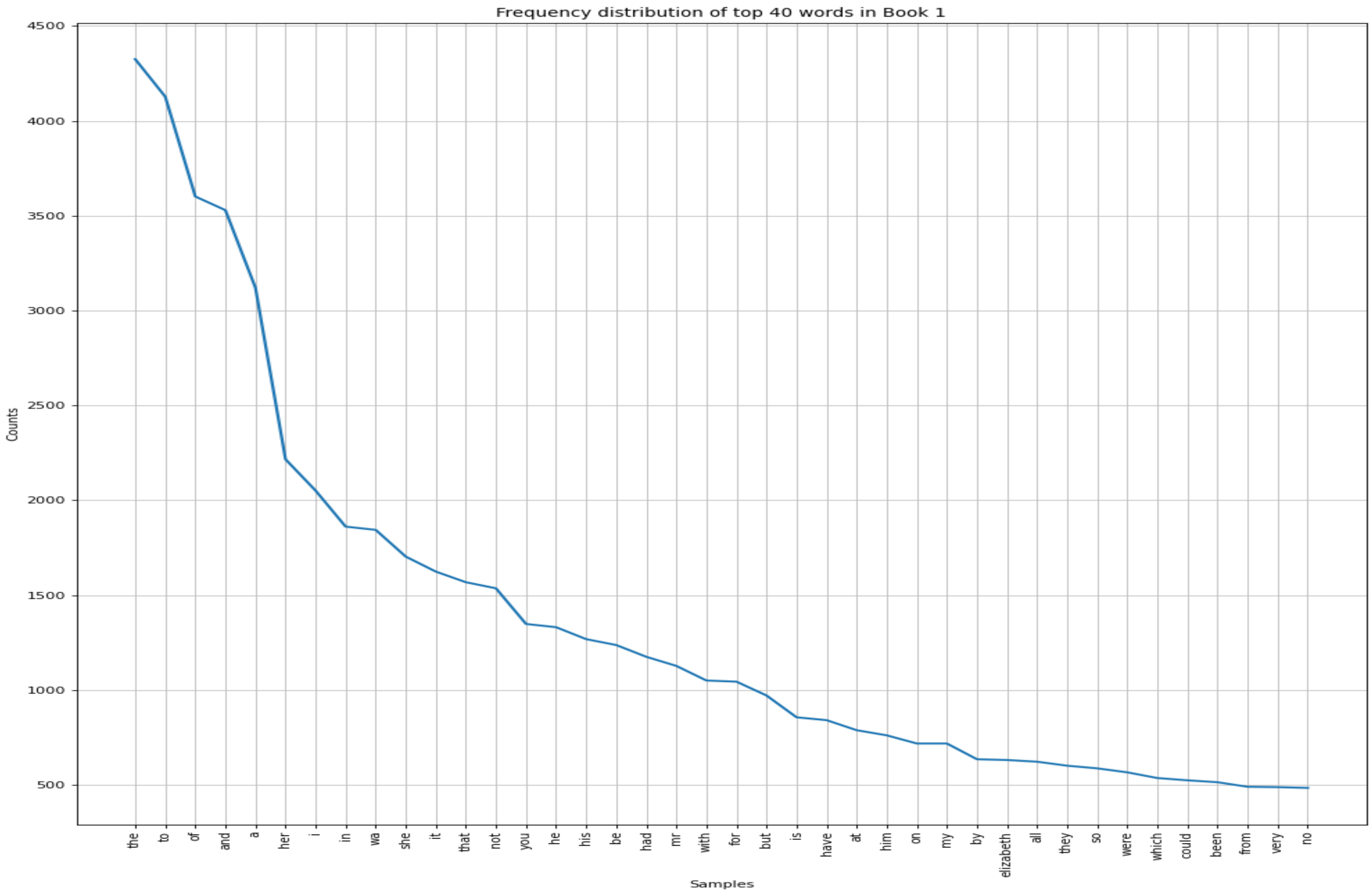
- After tokenizing the text, we imported the function *FreqDist()* from the module *nltk.probability* which is helpful in probability calculations, where frequency distribution counts the number of times that each outcome of an experiment occurs. We stored the frequency distribution of the two strings 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, ...})
```

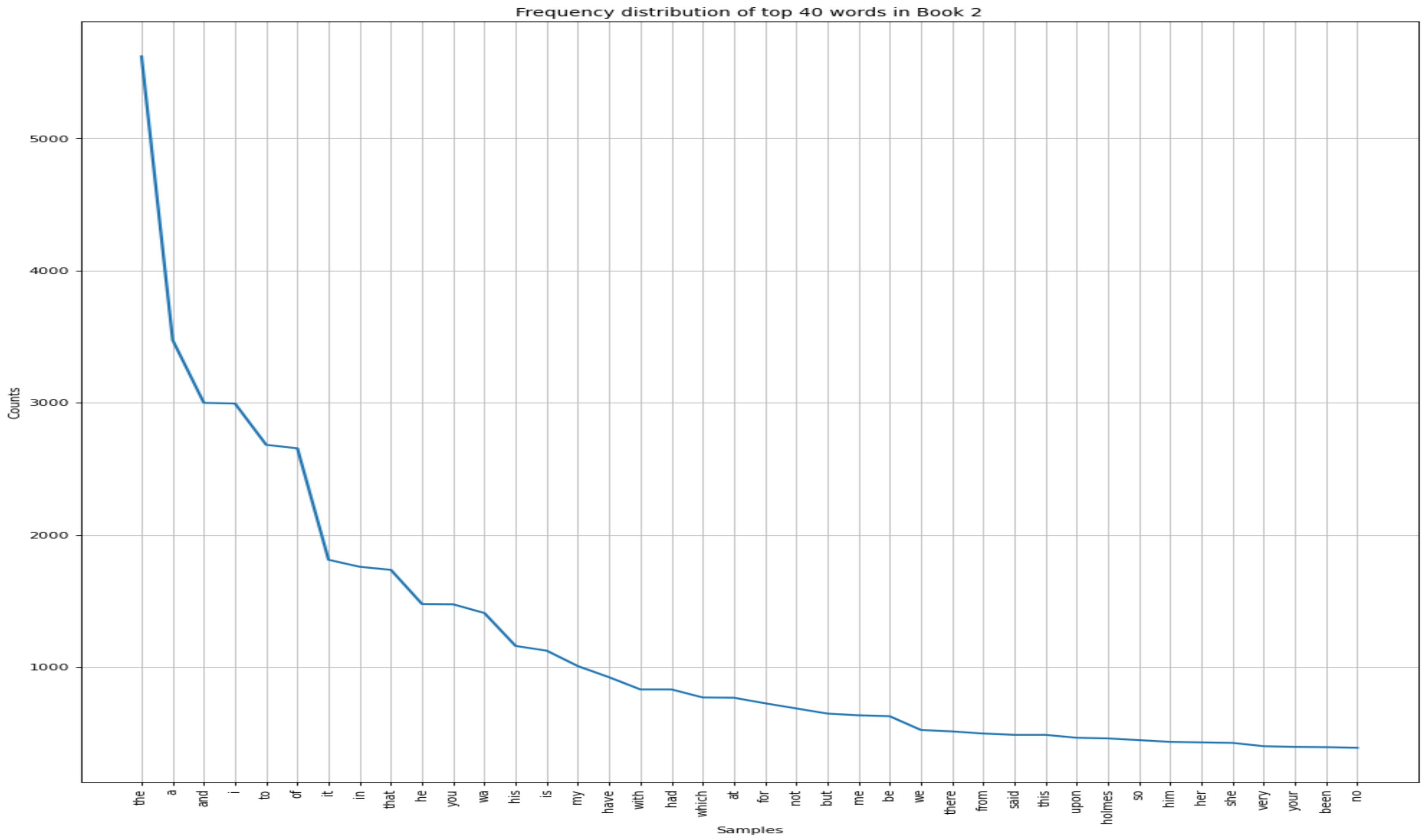
The *FreqDist* object for string T1

- 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.

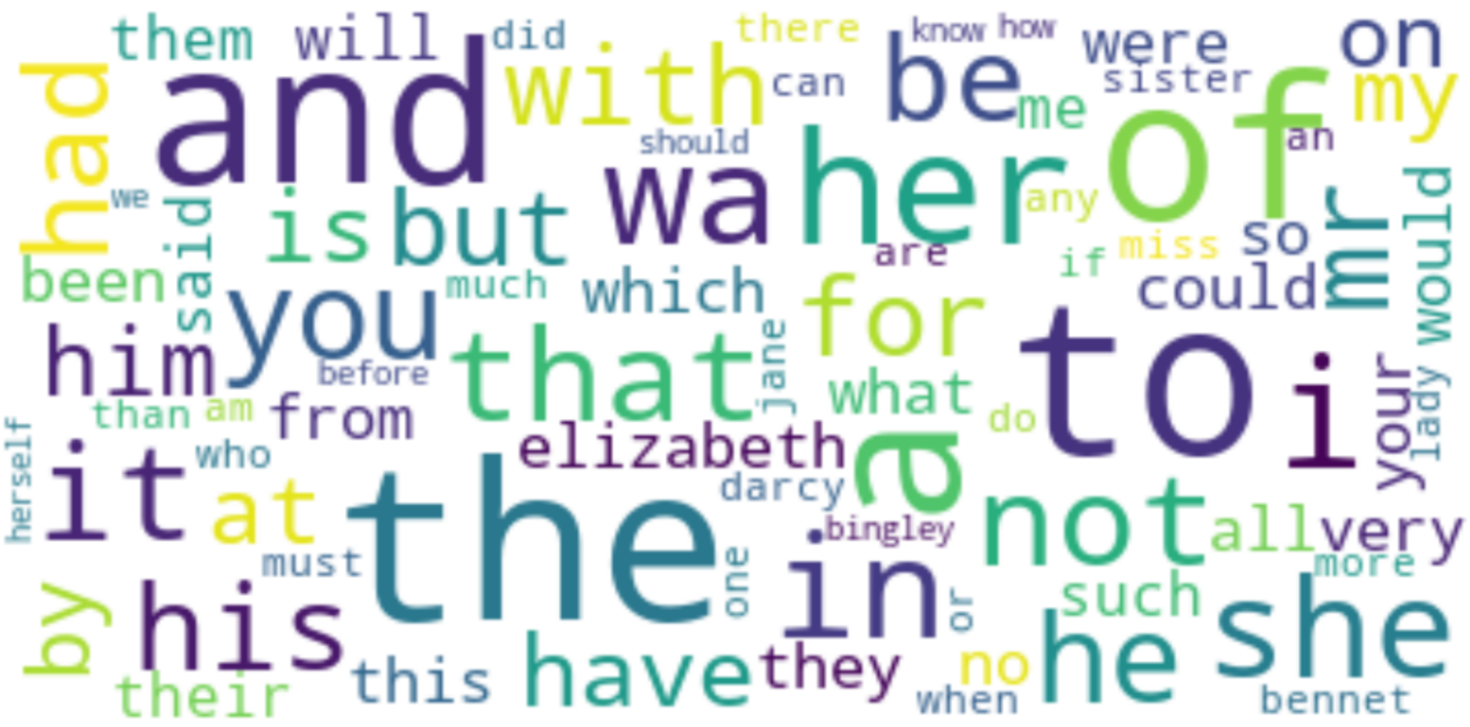

Frequency distribution of top 40 words in Book 2

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 :

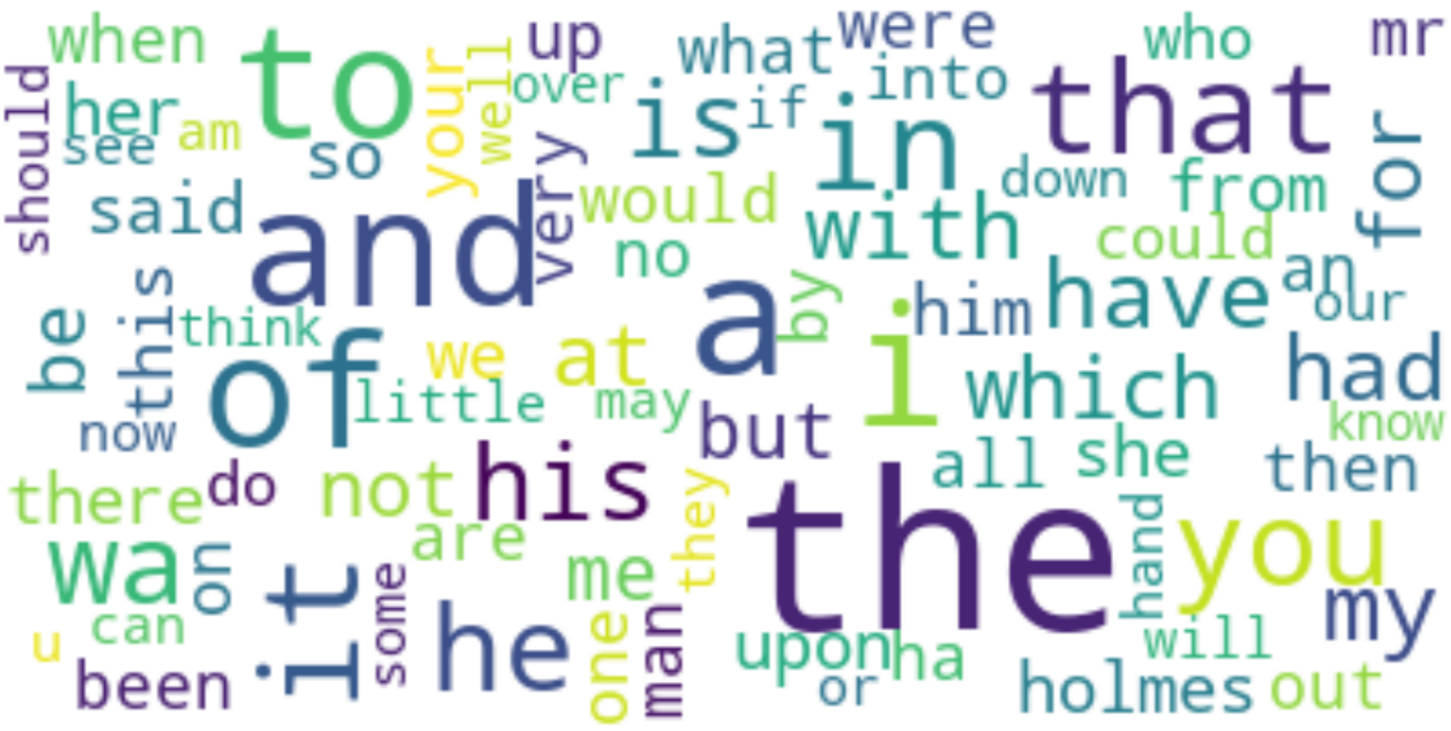
- For creating word cloud for both T1 and T2 we imported *Counter* from the module *Collections*. Then we imported *wordcloud* from the module *wordcloud*. Then we used functions *plt.figure() , plt.axis(), plt.show()* for the visualization of wordcloud. We made the word cloud for the top 80 most frequently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



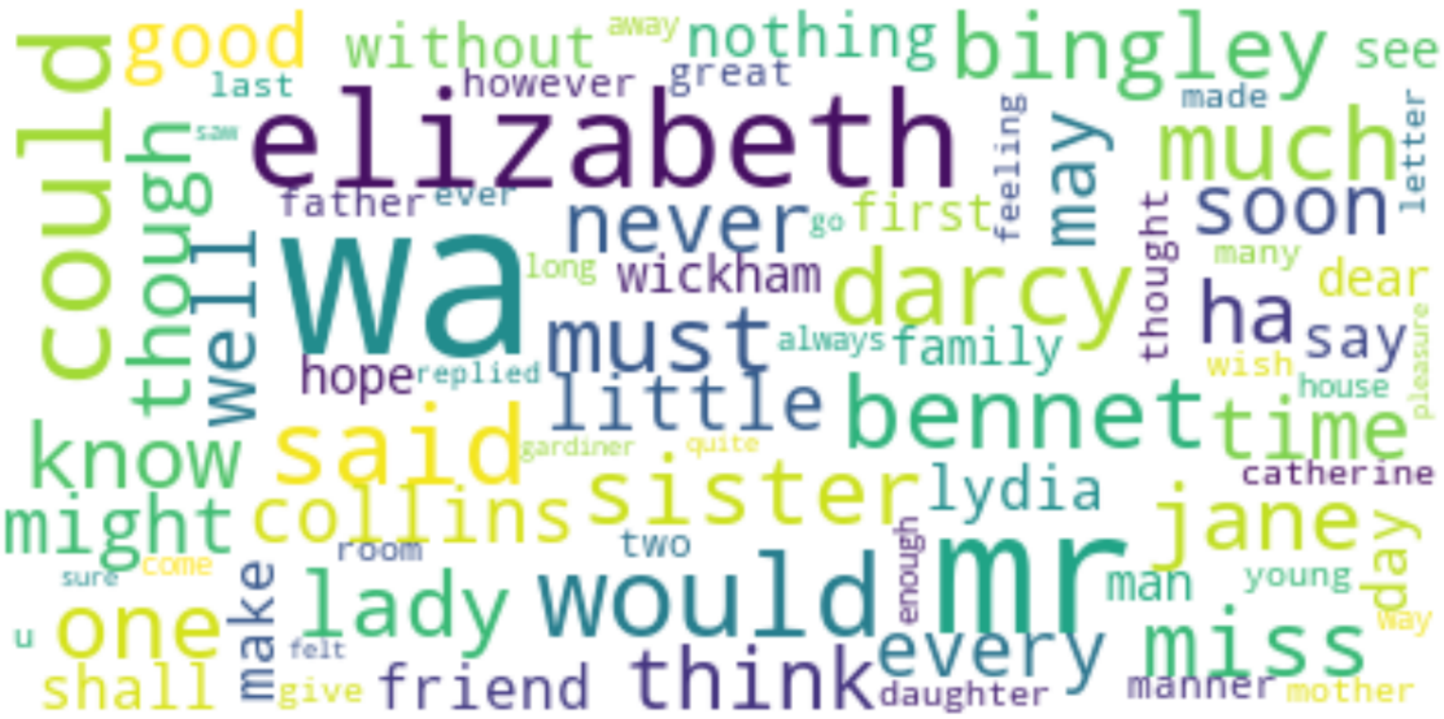
Word Cloud for Book 2 (with stop words)

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:**

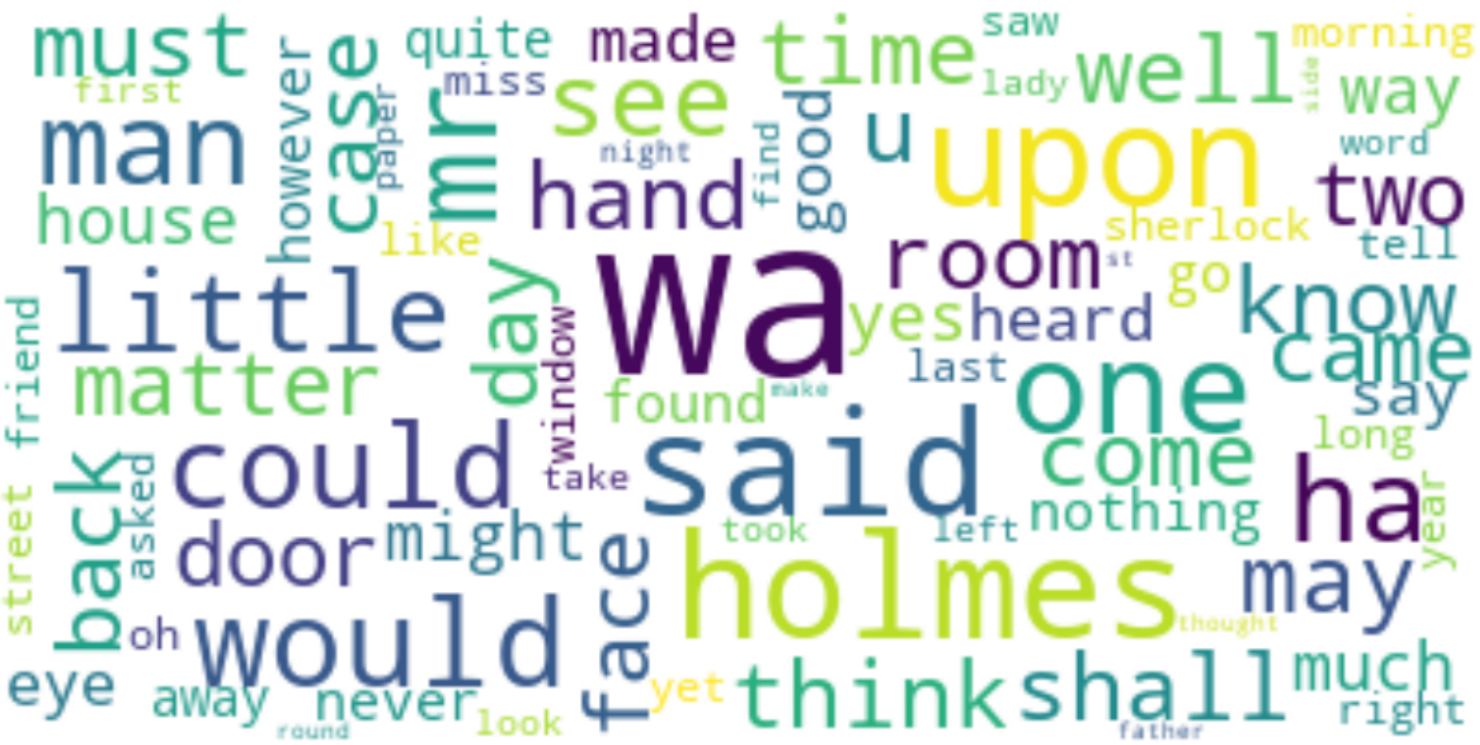
- We remove the stop words from the text. We used the *nltk.corpus stopwords* from English and removed them from the text. We updated the frequency distribution of words in the texts and repeated the above process to obtain a word cloud without stop words which is attached below :



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.



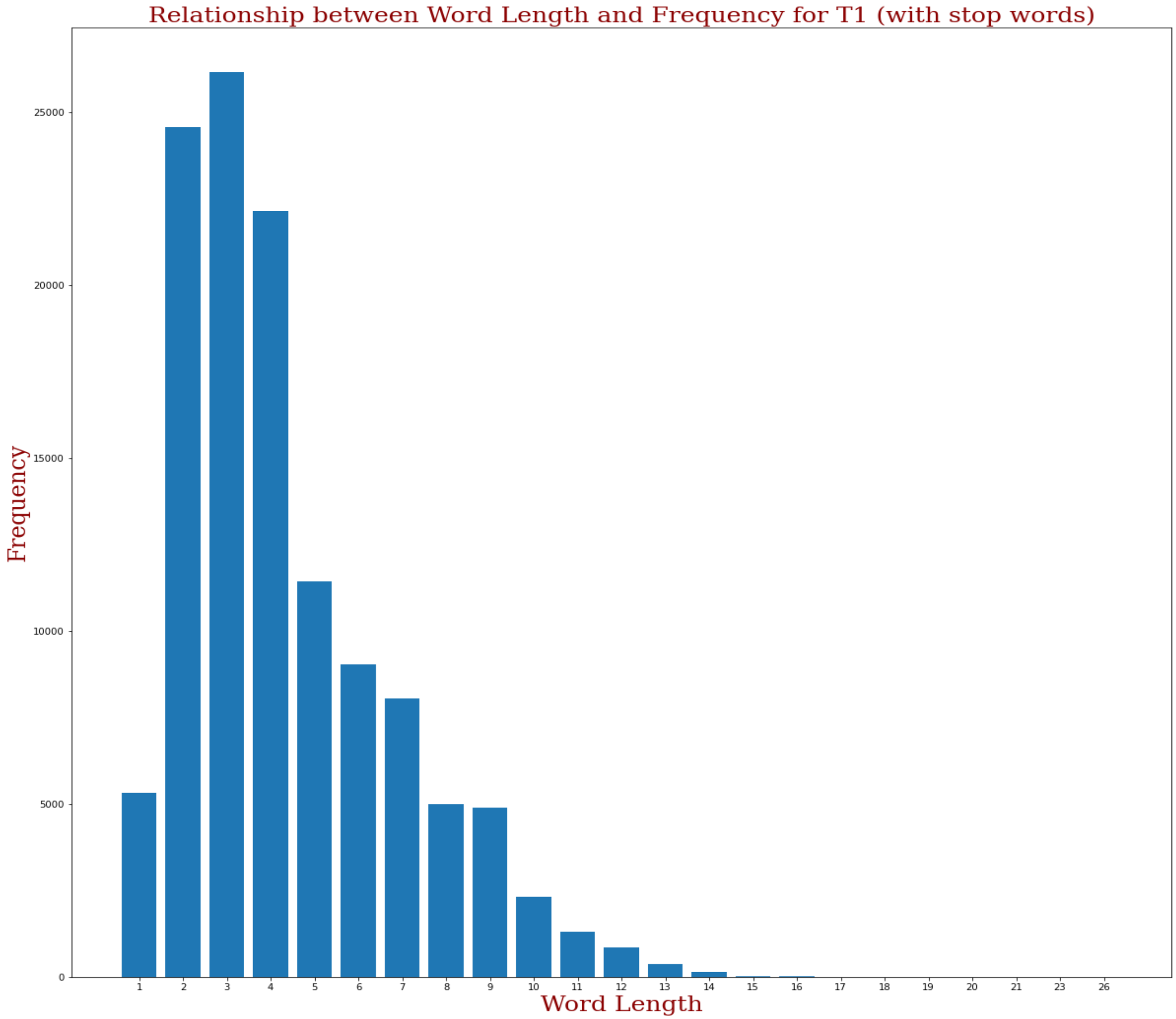
Word Cloud for Book 2 (without stop words)

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.

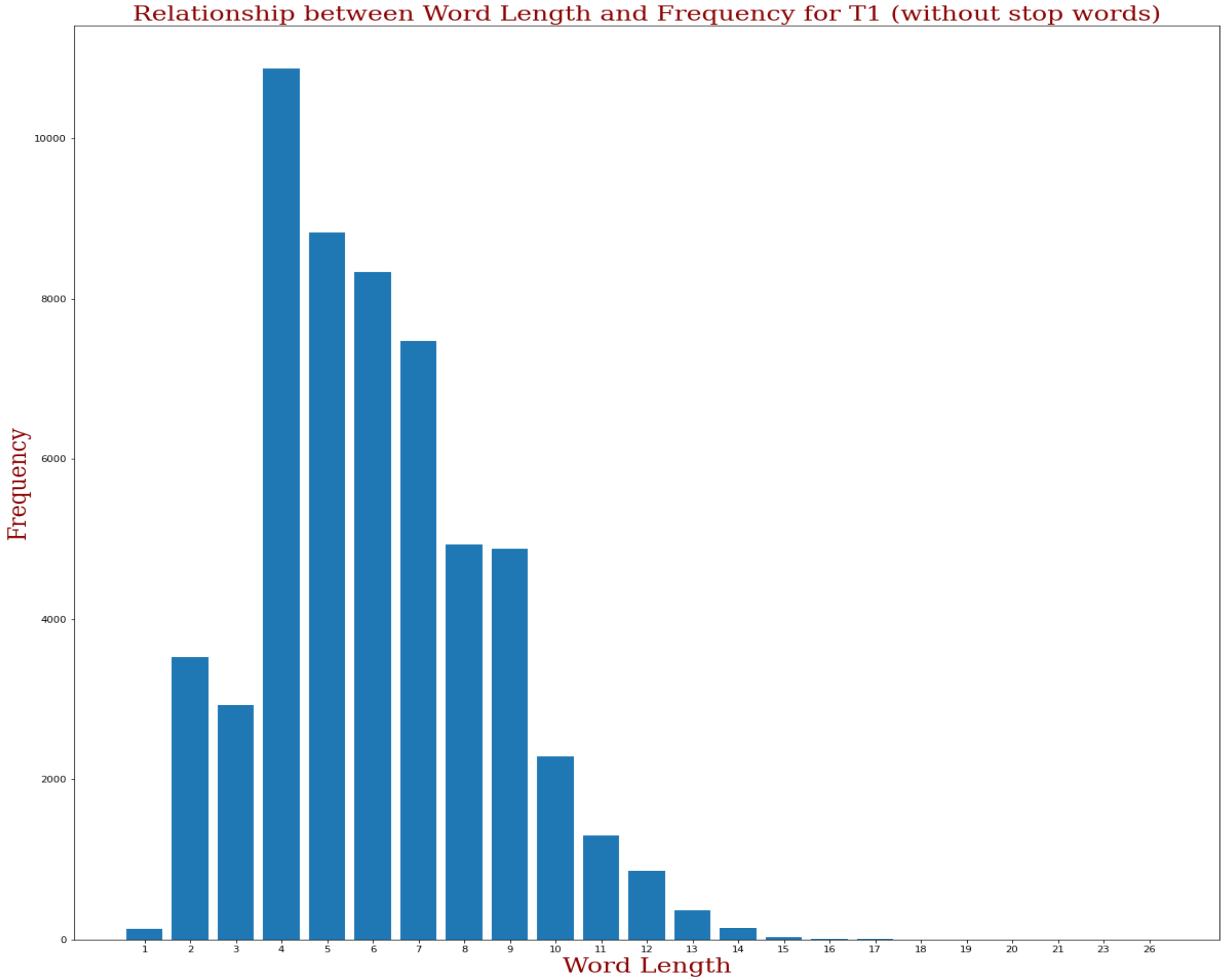
## Relationship between the word length and frequency :

- 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 :



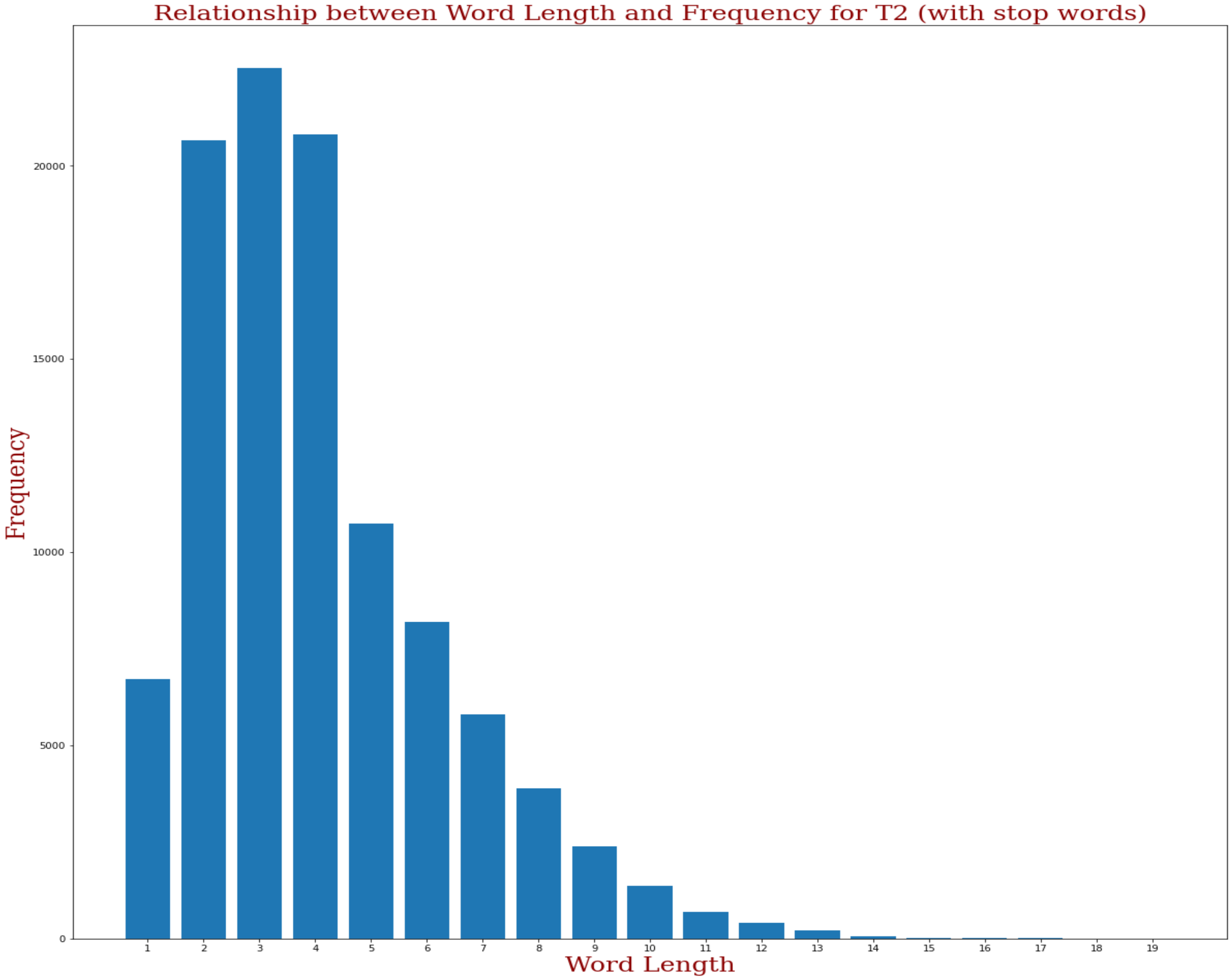
Inference :

- 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.



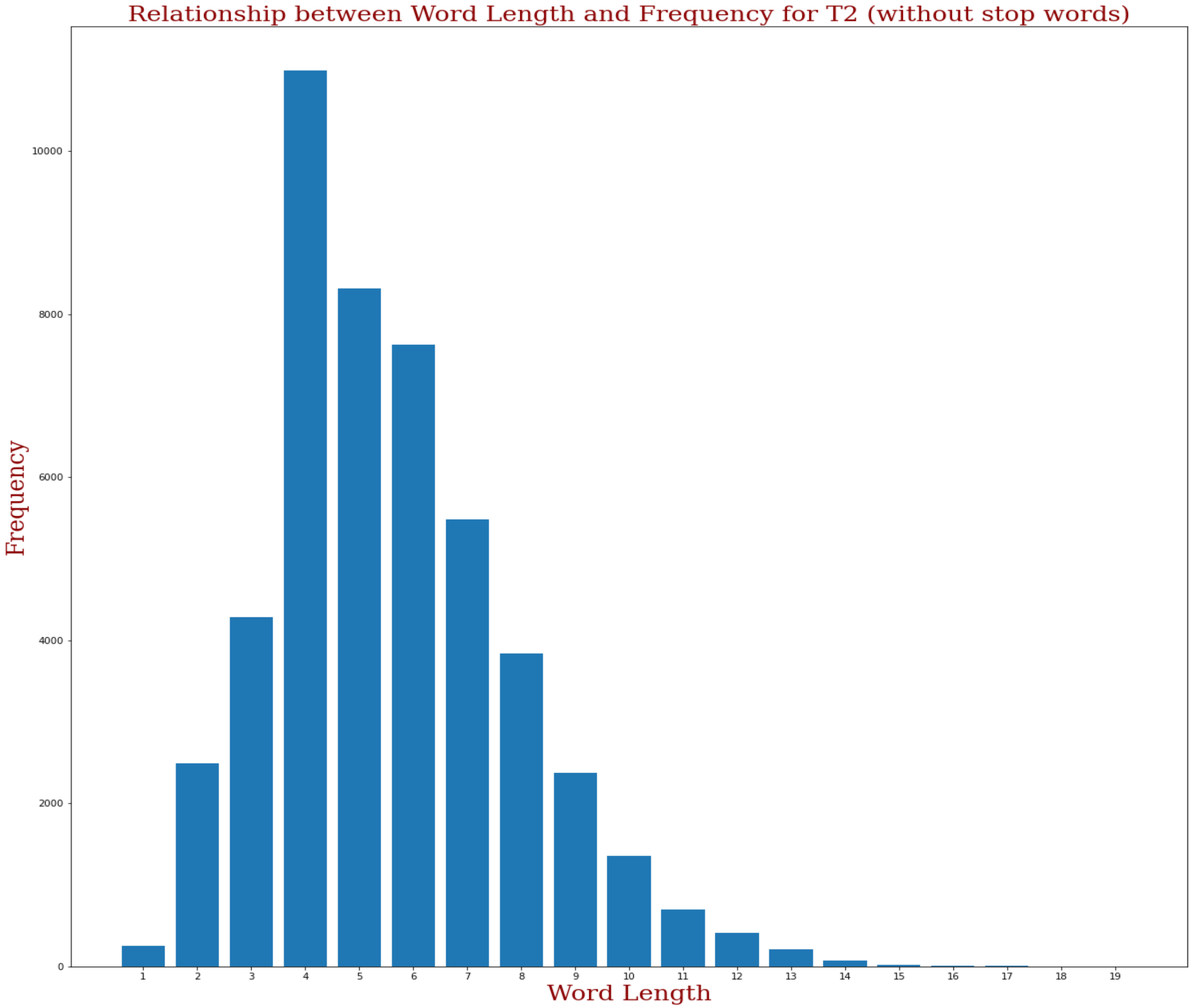
Inference :

- 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.

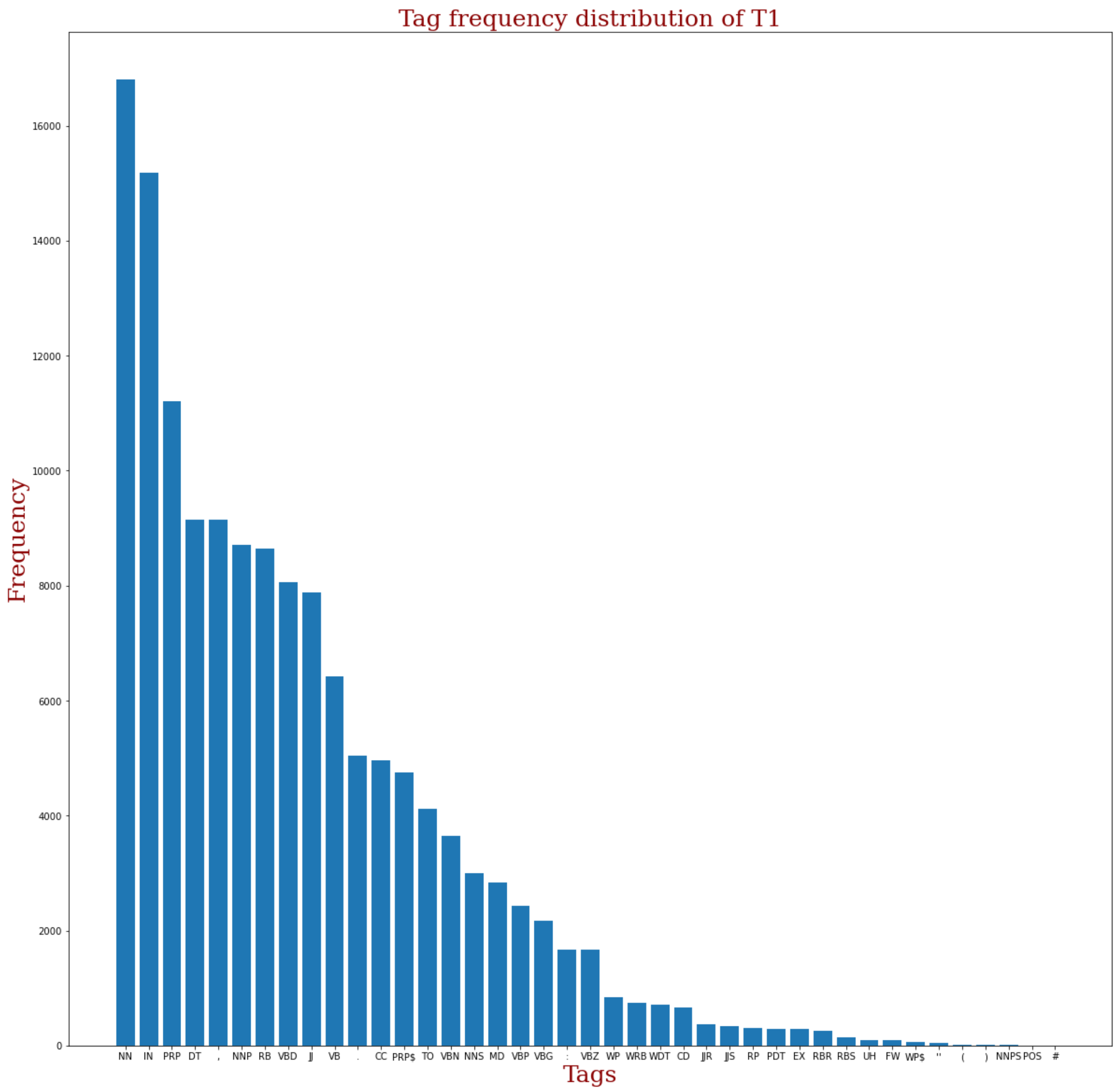


Inference :

- 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.

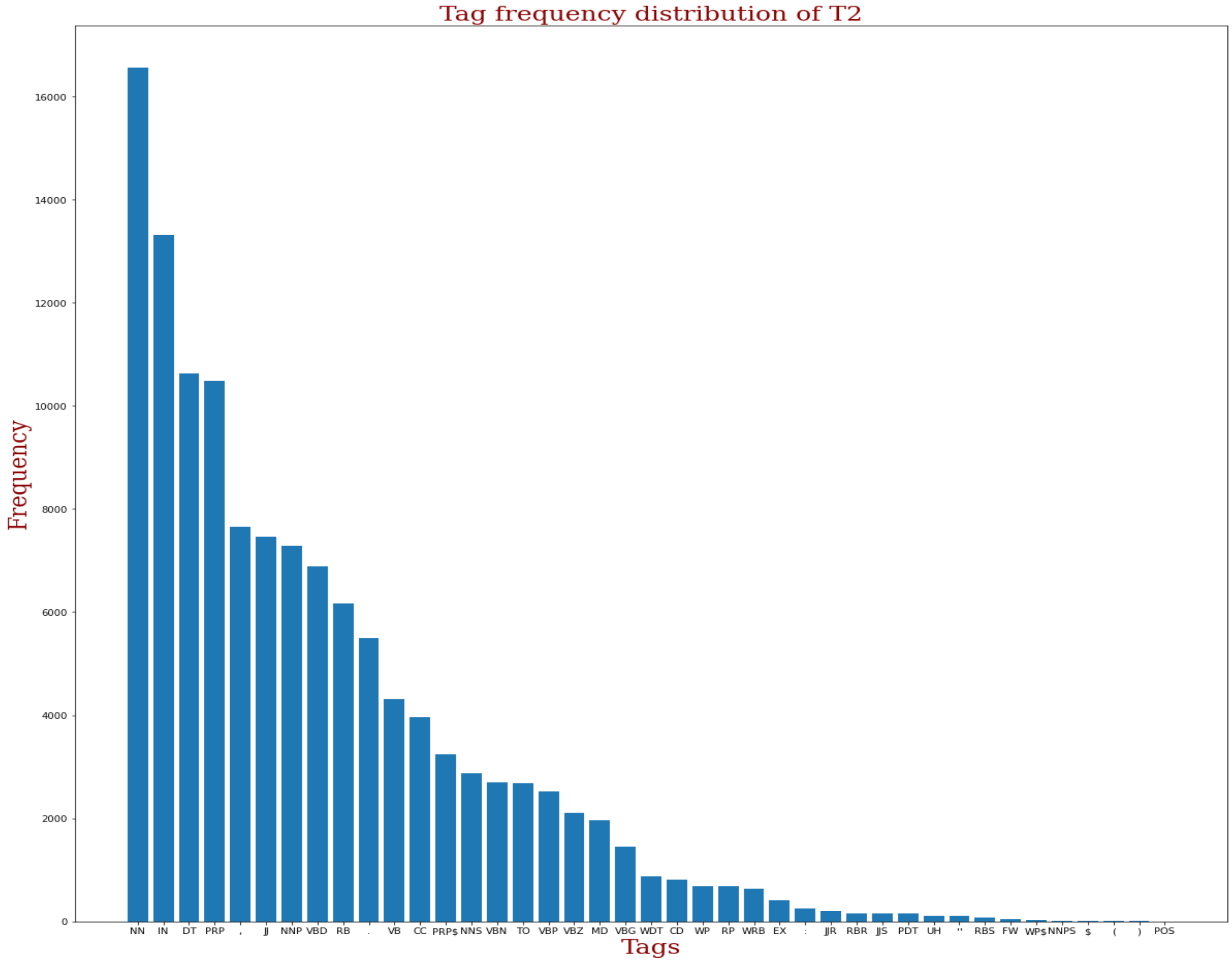
## PoS Tagging and distribution of tags :

- 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
16000
14000
12000
10000
8000
6000
4000
2000
0
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
#
Tags

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.

# 5. 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, etc. The Graphs, Bar Charts, Word clouds 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

- 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/