NLP Project Report
Round-1

Team Name
KAFKAESQUE

TEAM MEMBERS

1. SOHAM BAIJAL 19UCS098 2. SHIVAM SHUKLA 19UCS040 3. KESHAV MUNDRA 19UCS074

GitHub Project Code Link

https://github.com/sbaijal/NLP_Project_Round1.git

Data Description

Two books downloaded from http://www.gutenberg.org

Book 1 (T1) - **Crime and Punishment**
Book 2 (T2) - **War and Peace**

Data Preparation

The data is imported in raw form and prepared by removing the not required headers and chapter names and licence documents at the bottom of the book.

Data Preprocessing Steps

After obtaining only the book text we tokenize to remove '/n' , '/r' and store the words in a list.
We then apply two preprocessing steps to convert our raw data into a meaningful list of tokens to extract information.
Steps of Preprocessing

1) Converting all the tokens in the list to lowercase
2) Applying lemmatization

Note : Substantial changes were seen in output after applying the above two mentioned steps

Problem statement

- Import the text, let's call it as T1 and T2
- Analyze the frequency distribution of tokens in T1 and T2 separately
- Create a Word Cloud of T1 and T2 using the token that you have got

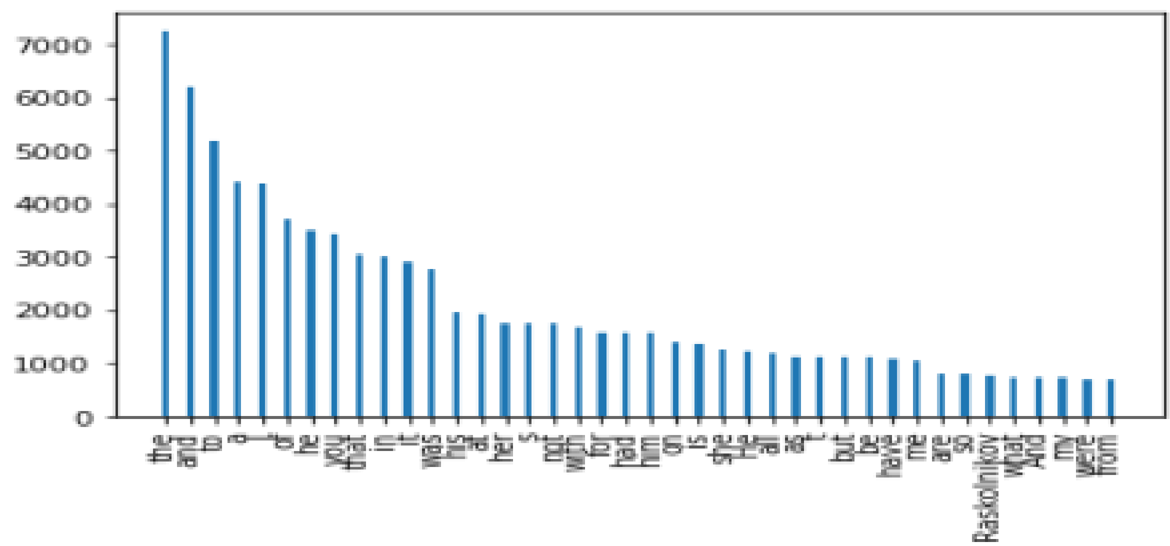
● Remove the stop words from T1 and T2 and then again create a word cloud - what's the difference it gives when you compare with word cloud before the removal of stop words?

● Evaluate the relationship between the word length and frequency for both T1 and T2 — what's your result?

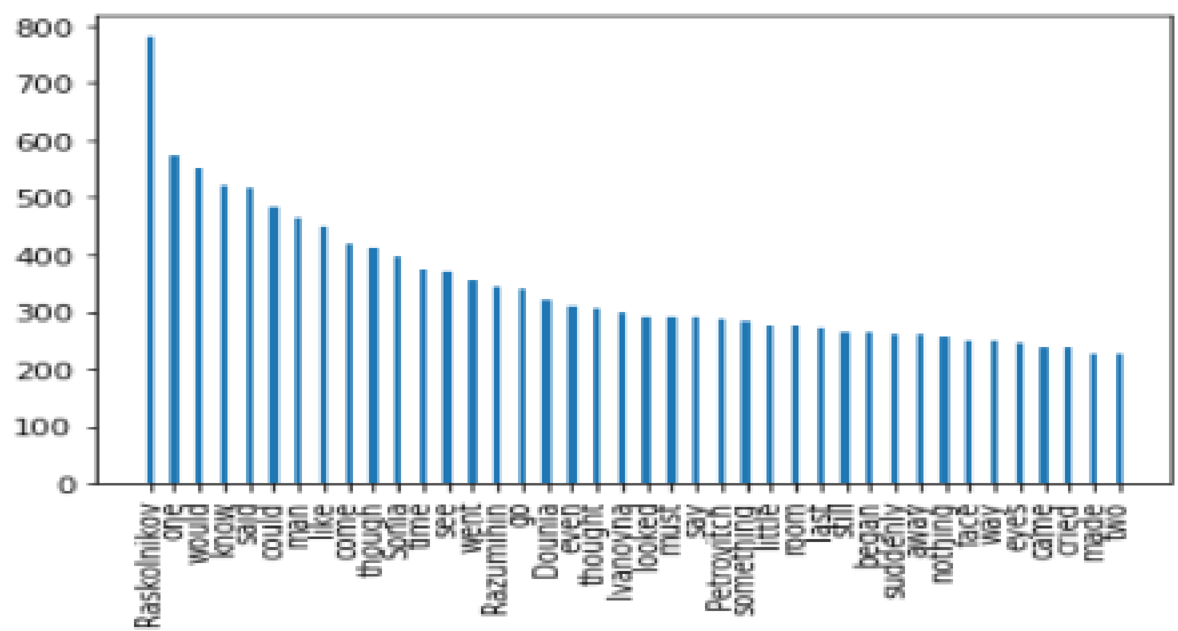
● Do PoS Tagging for both T1 and T2 using anyone of the four tag sets studied in the class and get the distribution of various tags

Plots

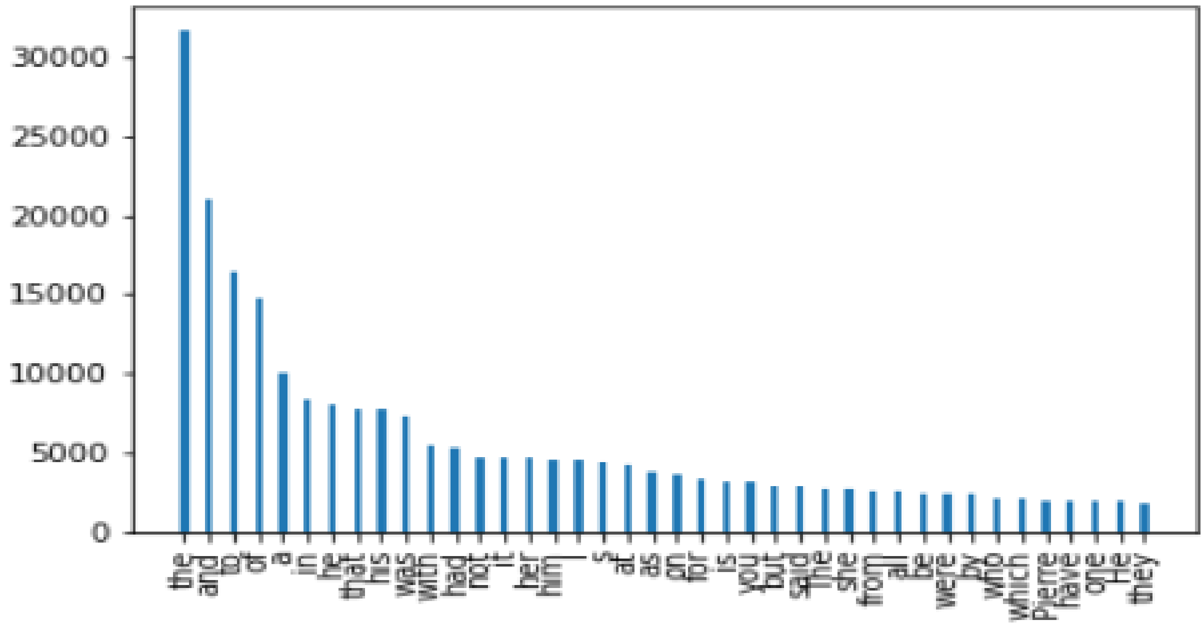
1) T1 Text frequency distribution of top 40 most frequent words including Stop Words



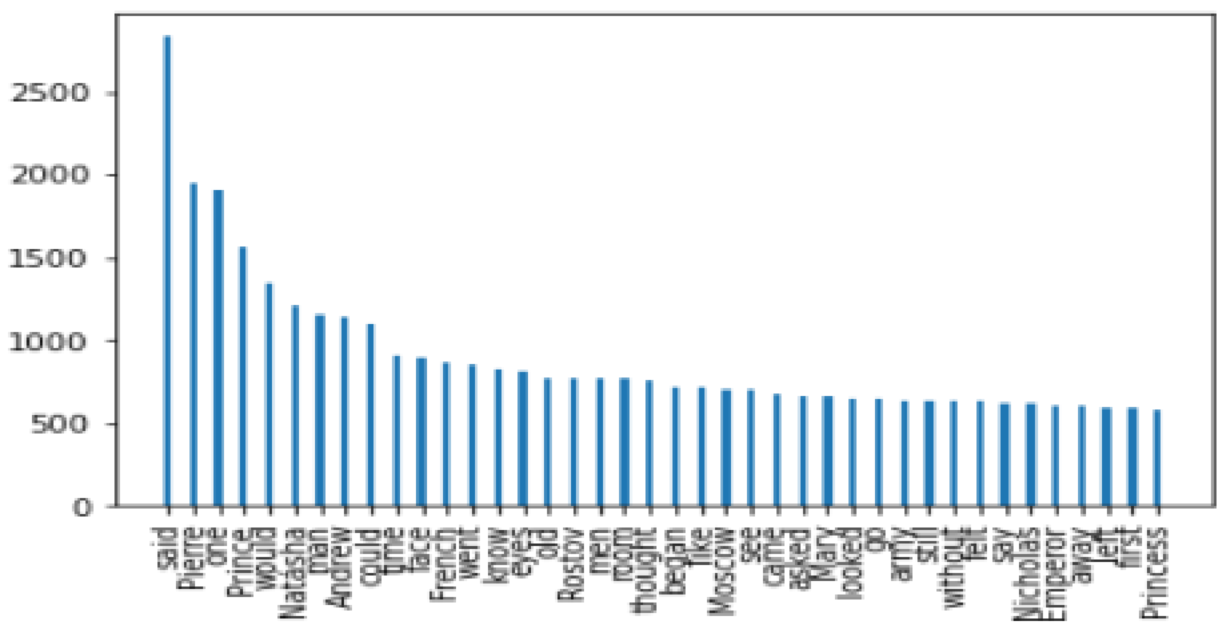
2) T1 Text frequency distribution of top 40 most frequent words excluding Stop Words



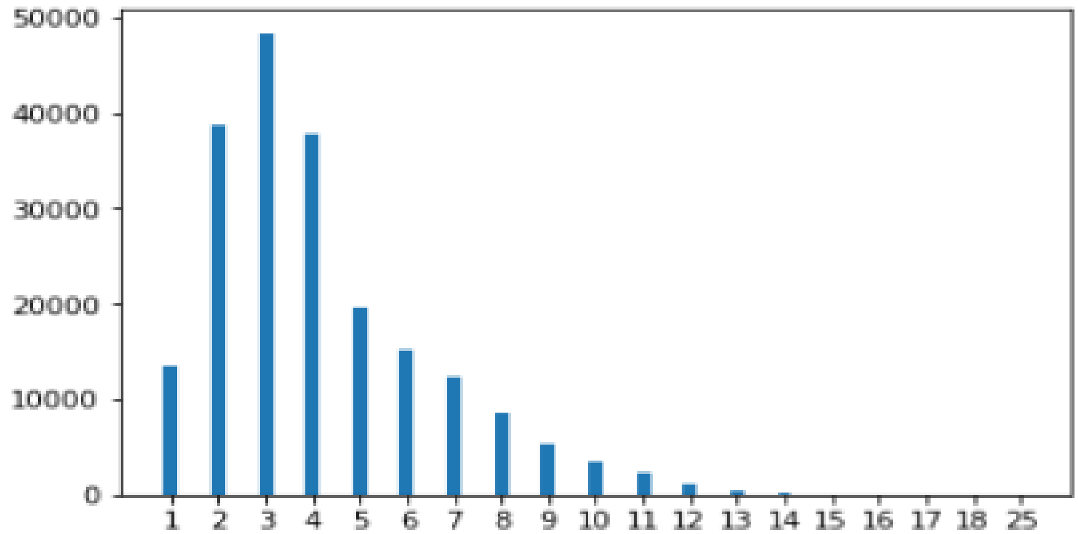
3) T2 Text frequency distribution of top 40 most frequent words including Stop Words



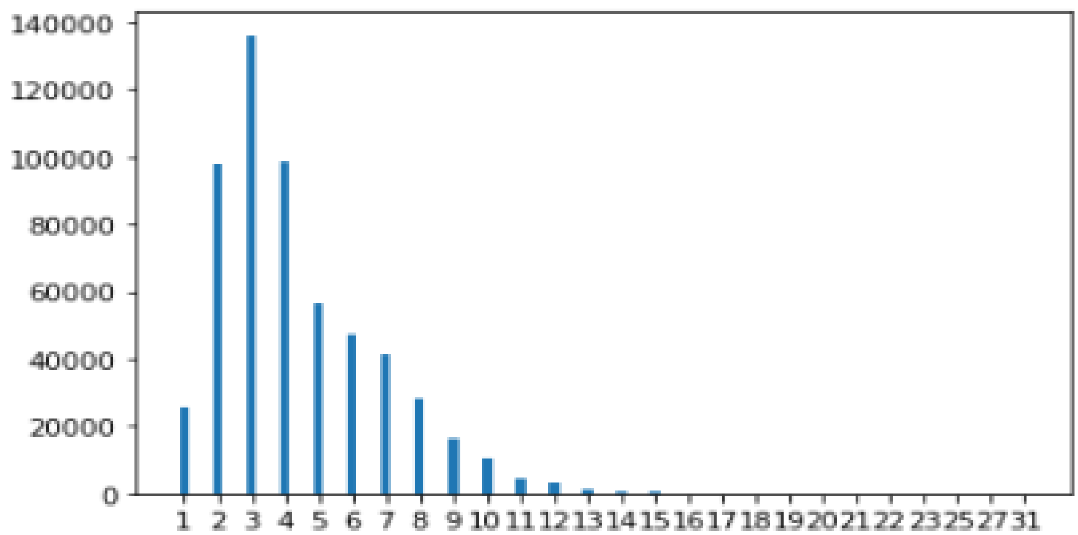
4) T2 Text frequency distribution of top 40 most frequent words excluding Stop Words



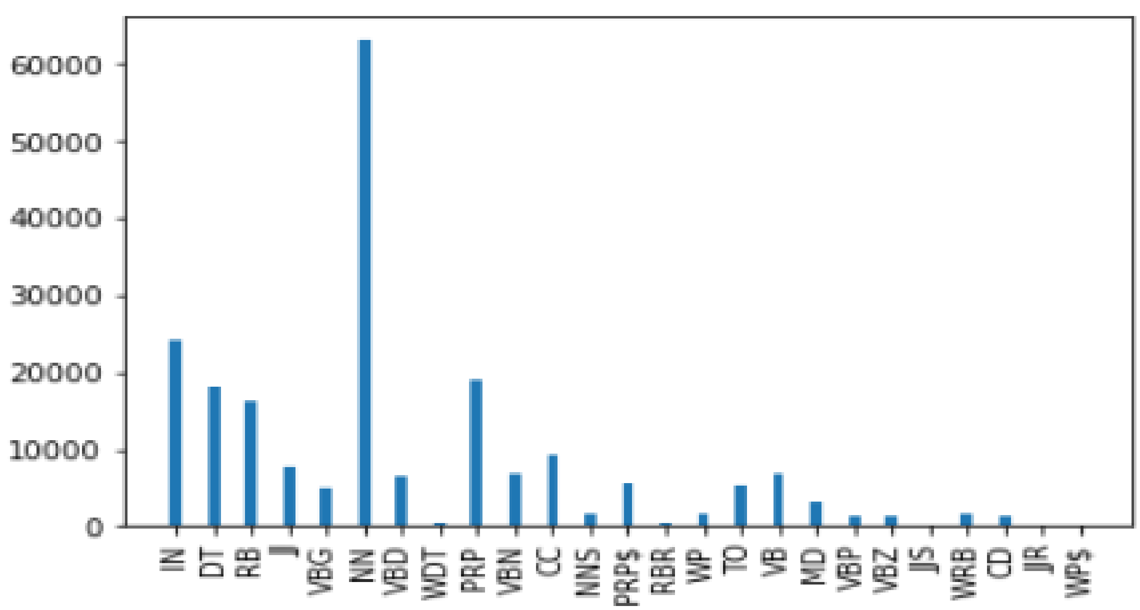
5) T1 Word Length Frequency Distribution Graph



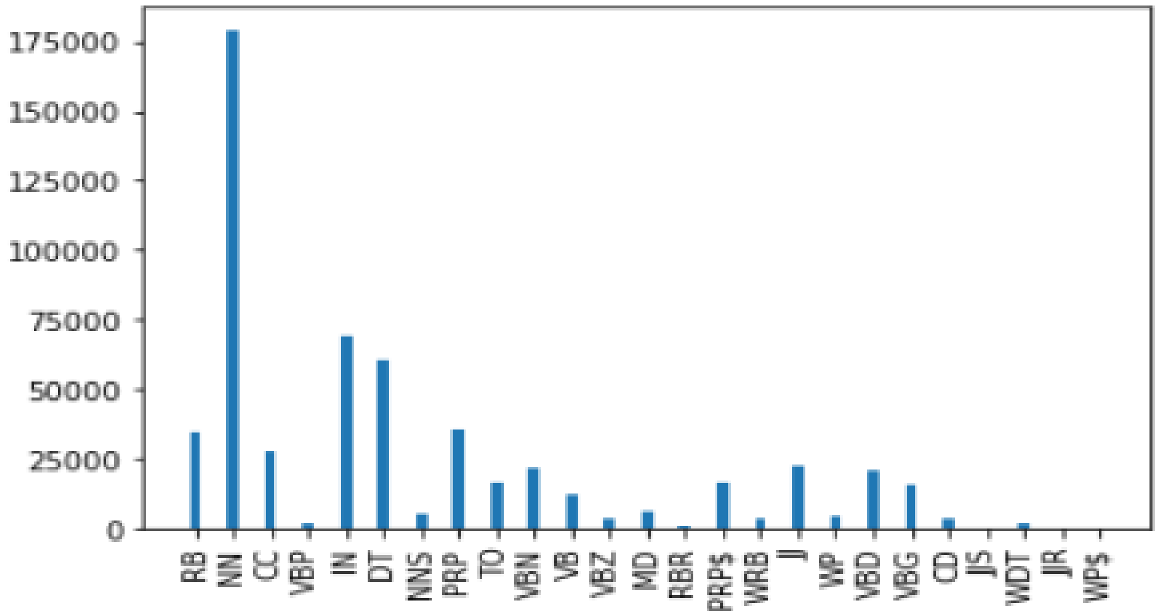
6) T2 Word Length Frequency Distribution Graph



7) T1 POS TAGGING frequency bar plot



8) T2 POS TAGGING frequency bar plot

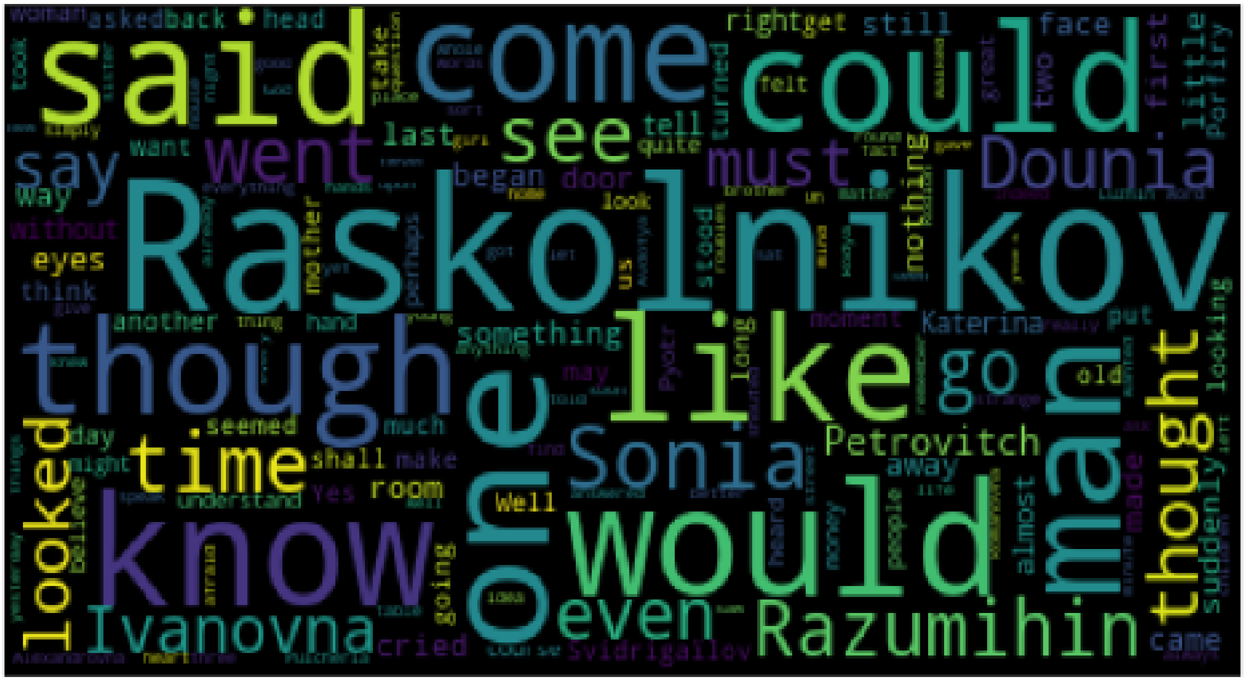


Figures

T1 Text Word Clouds

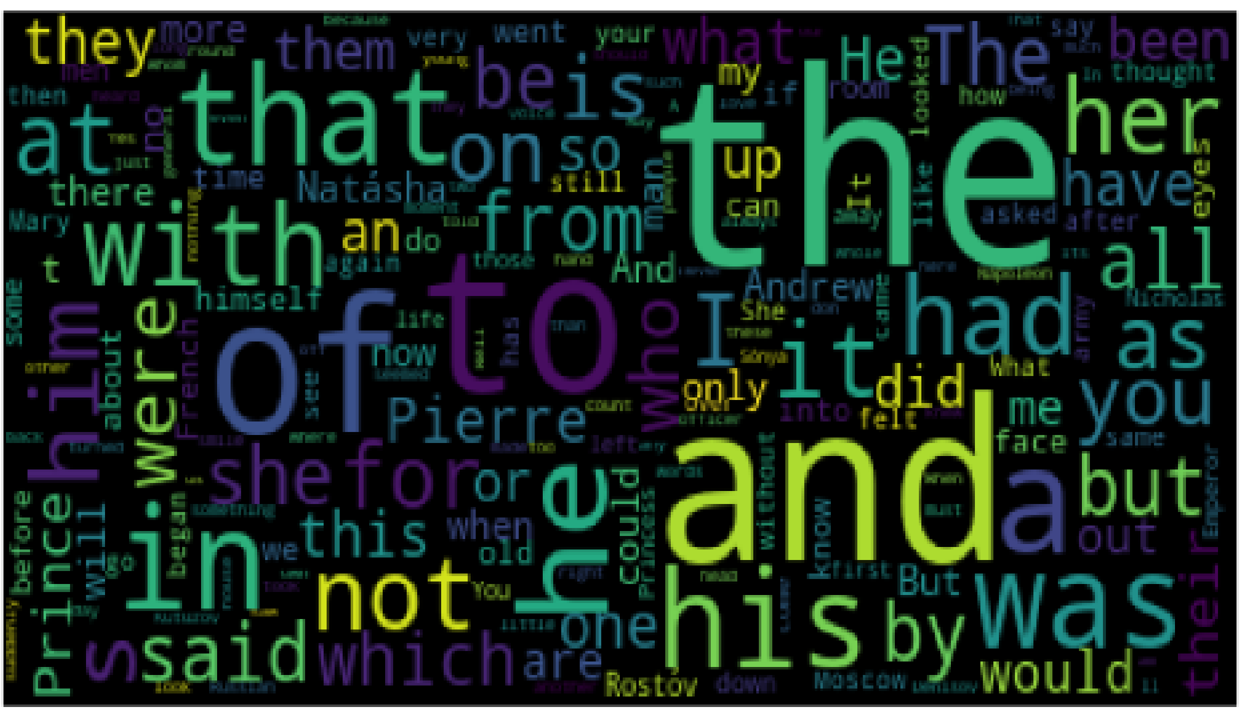
1) Word Cloud with Stop Words included

2) Word Cloud excluding Stop Words

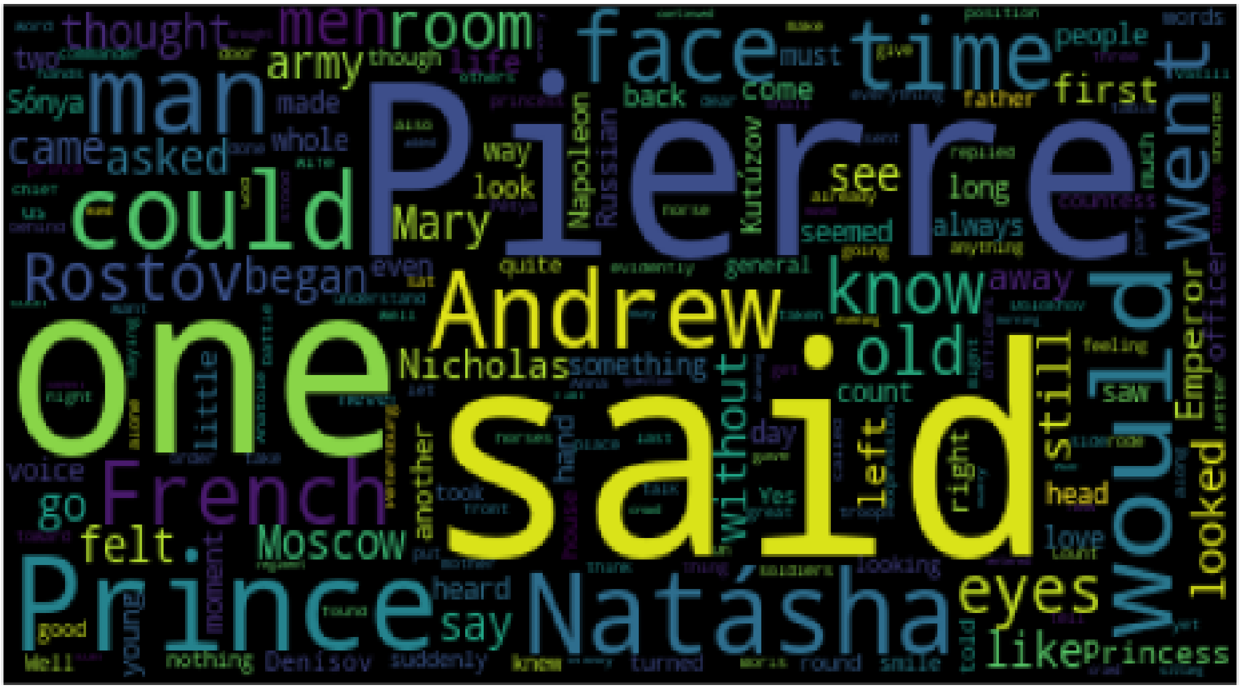


T2 Text Word Clouds

1) Word Cloud with Stop Words included



2) Word Cloud excluding Stop Words



Tables

1) T1 Pos Tagging Output Table

TAG	FREQUENCY
IN	24209
DT	18034
RB	16130
JJ	7602
VBG	5026
NN	63091
VBD	6540
WDT	349
PRP	18990
VBN	6765
CC	9178
NNS	1760
PRP$	5732
RBR	563
WP	1702
TO	5239
VB	6822
MD	3163
VBP	1390
VBZ	1468
JJS	279
WRB	1617
CD	1223
JJR	96
WP$	20

2) T2 Pos Tagging Output Table

TAG	FREQUENCY
RB	34684
NN	178612
CC	27777
VBP	1802
IN	69348
DT	60750
NNS	5806
PRP	35342
TO	16627
VBN	21999
VB	12631
VBZ	3441
MD	6278
RBR	1222
PRP$	16727
WRB	3638
JJ	23070
WP	4911
VBD	21345
VBG	15650
CD	3429
JJS	795
WDT	2085
JJR	354
WP$	111

Round-2 Steps

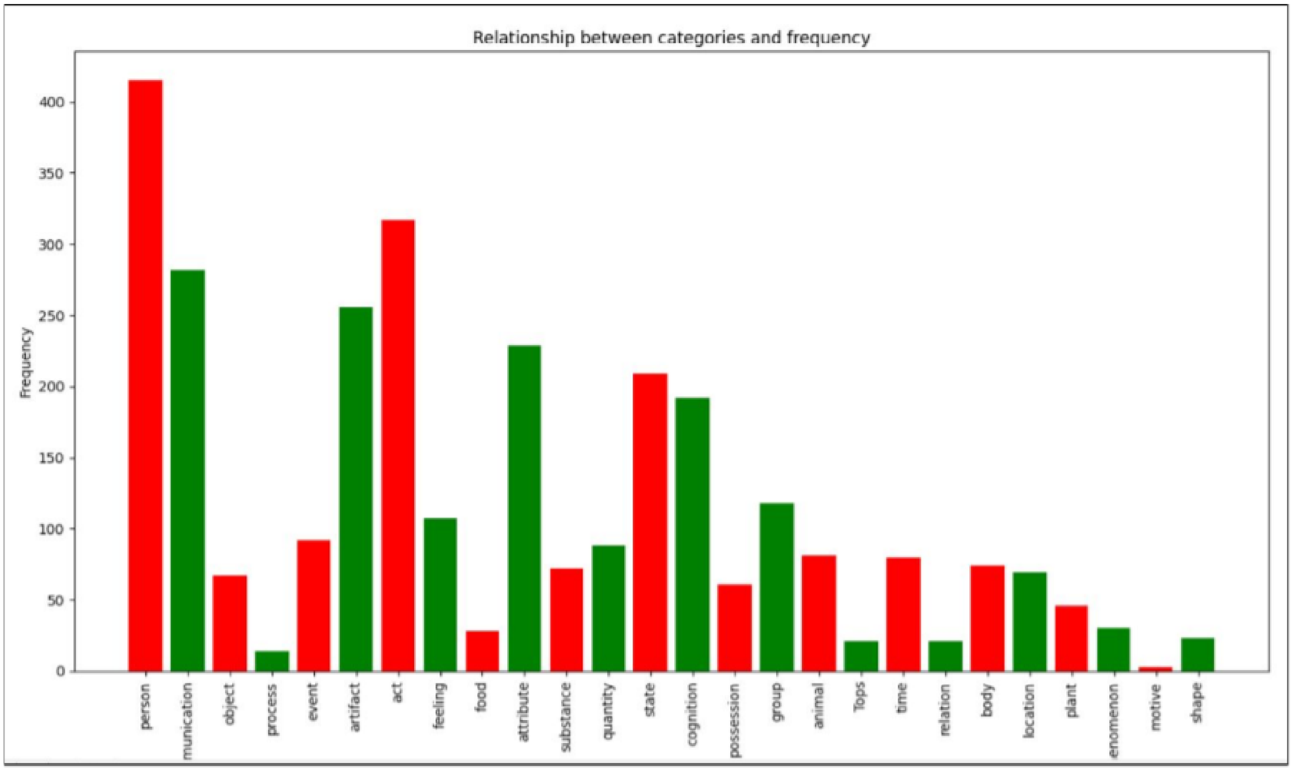
Problem Statement -1

Find the nouns and verbs in both the novels. Get the immediate categories (parent) that these words fall under in the WordNet. 2. Get the frequency of each category for each noun and verb in their corresponding hierarchies and plot a histogram for the same for each novel.

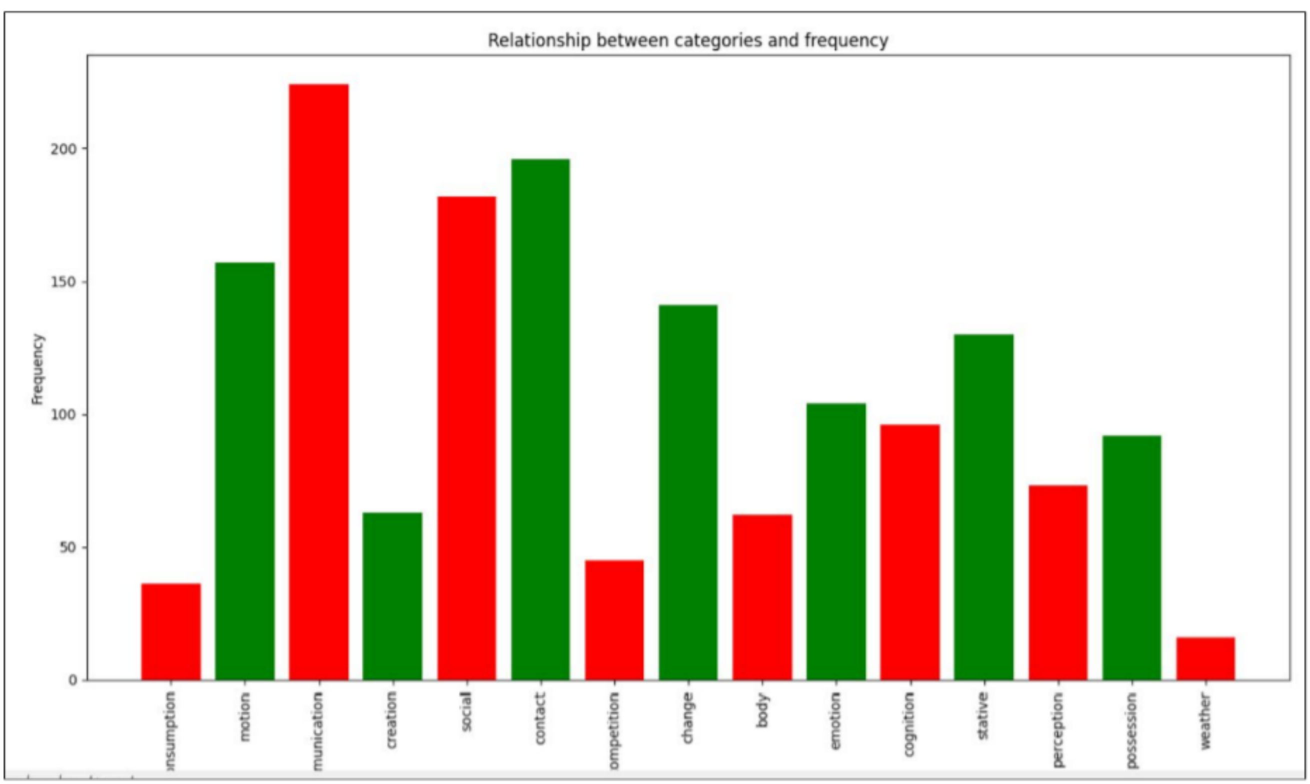
Solution

1. We first imported the books, preprocessed them, tokenized them, removed the stop words and then applied the PoS tagging to them (detailed explanation of these steps can be found above).
2. Using PoS Tags, we extracted the nouns and verbs.
3. For the extraction of nouns, we ran a for loop over the list of PoS tagged tuples which we got after PoS tagging. If the tag starts with 'N', then the corresponding word was appended in the nouns list. Similarly if the tag starts with 'V', the word belongs to the verbs list. Had the tag been directly compared with the string "NN", some of the words having tags such as "NNP" would have been left out from our nouns list. Similar is the case with verbs.
4. Now we have the nouns and verbs list. For each word in this list, its category was extracted from the word net. 19 WordNet :- It is the lexical database i.e. dictionary for different languages, specifically designed for natural language processing. For each word in the list, the list of its synsets is extracted using the function wordnet.synsets(word). Synsets :- It is a set of synonyms which share a common meaning. Moreover it is a simple interface that is present in NLTK to look up words in WordNet. Some of the words have only one Synset and some have several. Now using the 1st element of this list of synsets, the category was extracted using the function lexname().

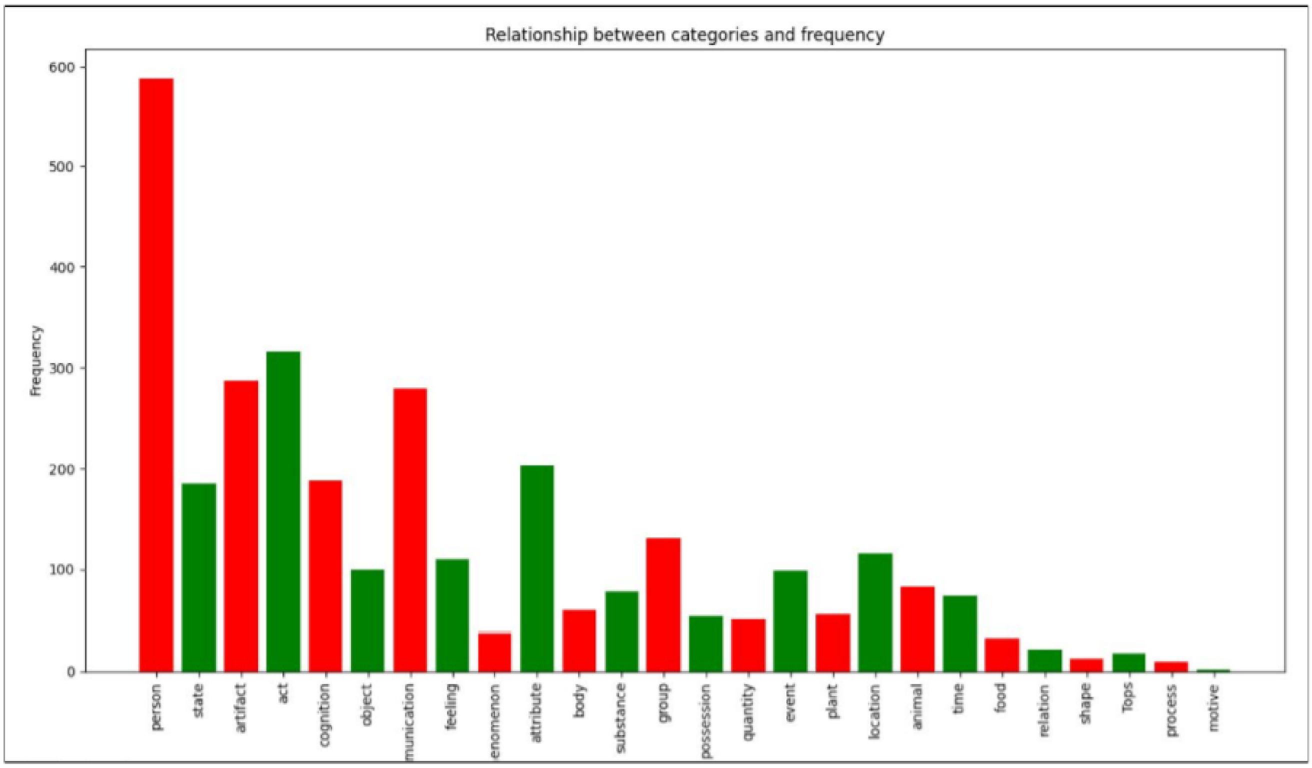
- Frequency plot of the noun categories of the nouns of book-1



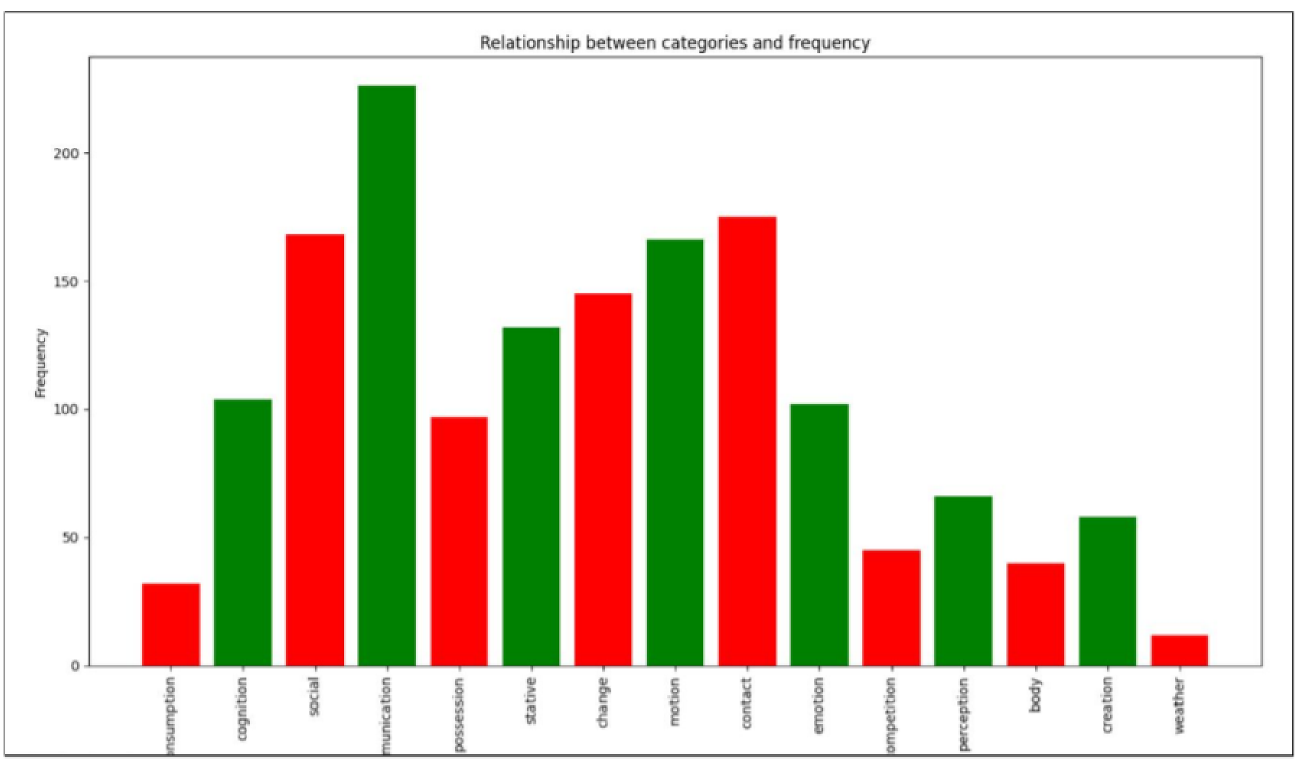
- Frequency plot of the verb categories of the verbs of book-1



- Frequency plot of the noun categories of the nouns of book-2



- Frequency plot of the verb categories of the verbs of book-2



Problem Statement - 2

Recognise all Persons, Location, Organisation (Types given in Fig 22.1) in book. For this you have to do two steps: (1) First recognise all the entity and then (2) recognise all entity types. Use performance measures to measure the performance of the method used - For evaluation you take a considerable amount of random passages from the Novel, do a manual labelling and then compare your result with it. Present the accuracy with F1 score here.

Solution

Steps :

1) First without doing any preprocessing, we perform tokenization for both the books. Both the books are POS tagged using nltk library.
2) Now, on this tagged data, we perform entity recognition using ne_chunk() function. All the used entities are recorded for both the books.

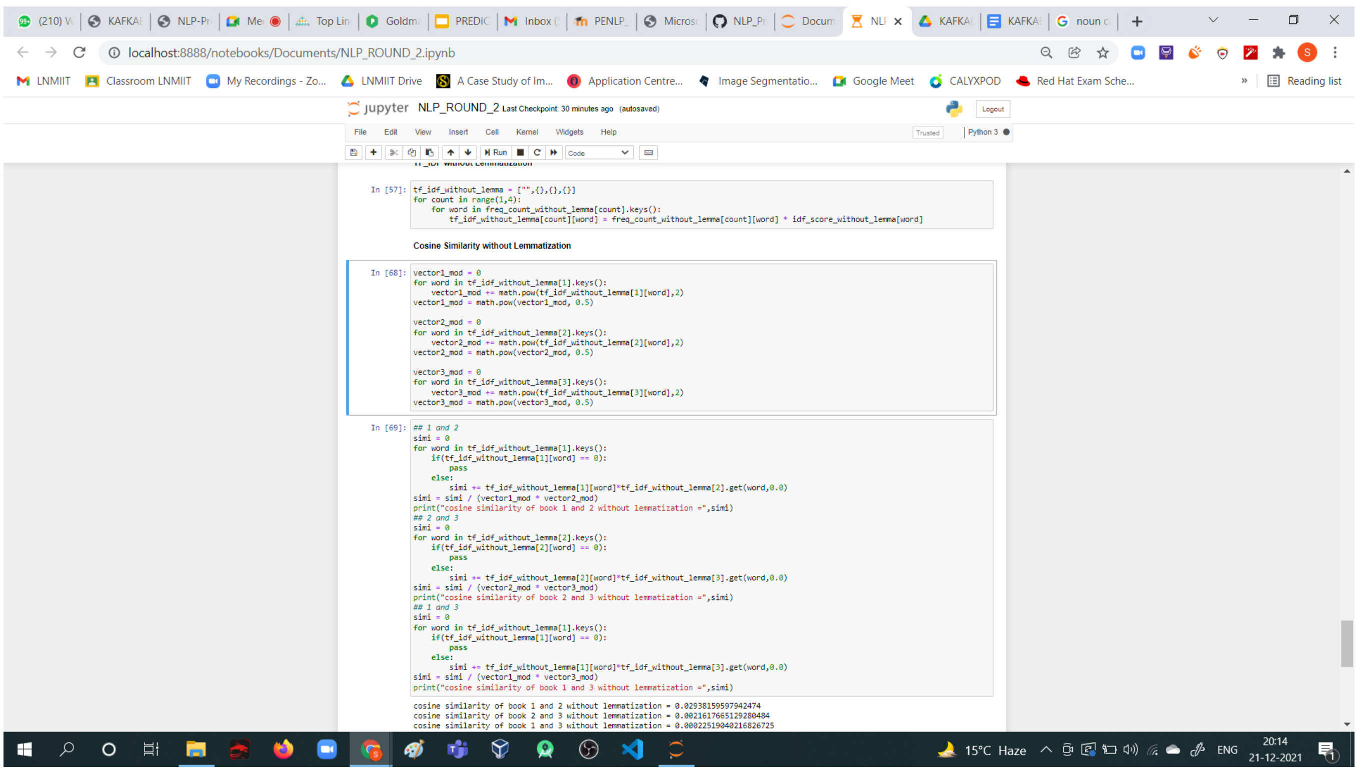
Entity Recognition :

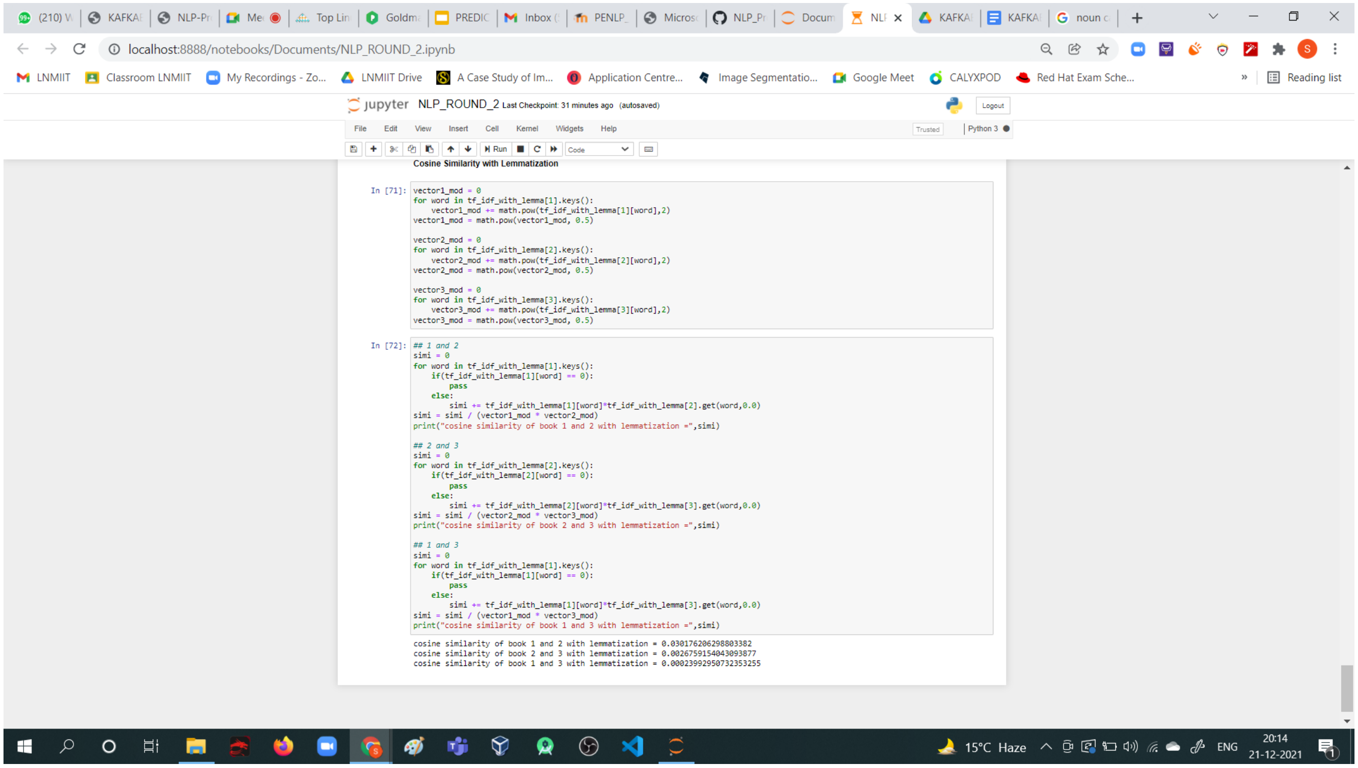
1) Sequence labelling is used to perform Named Entity Recognition (NER).
2) First we encode our training data with IOB tags. This is done manually by domain experts.
3) Set of features are associated with each token to be labelled.
4) When an adequate set of features are extracted from a training set, it is encoded in a form appropriate to train a machine learning based sequence classifier.
5) These extracted features are augmented with our earlier IOB scheme with more columns.
6) Now a sequence classifier can be trained.

Problem Statement-3

Create TF-IDF vectors for all books and find the cosine similarity between each of them and find which two books are more similar. Do lemmatization of the books and recreate the TF-IDF vectors for all the books and find the cosine similarity of each pair of books.

Solution





In-case of without lemmatization maximum similarity was found between **books 1 and 2.**

In-case of lemmatization maximum similarity was found between **books 1 and 2.**

Note: The order of cosine similarity does not change with or without lemmatization.

Output with your Inferences

1) When preprocessing steps such as lowercasing and lemmatization were applied the results obtained were more consistent than that observed before applying preprocessing.
2) Removal of stop words showed prominent character names and proper nouns, more consistent with the observations of a reader, going through the text corpus.
3) The frequency distribution bar plot showed very similar kurtosis and skewness(**right/positive skewness**) with the mode of the plot being 3 which signifies that even though they are different text corpus the core distribution of the words remains similar.
4) The frequency distribution of both text corpus is also congruent with the word frequency of the English Vocabulary in the sense that words with large length are less frequent that the words with smaller length. So the word frequency is inversely proportional to word length for length greater than 3.