NLP Project Round-1 Report

(Team-Amigos)

Submitted by Members of Team Amigos:-

Aarzoo **19ucs075**

Yash Jindal **19ucs055**

Divyansh Goyal **19ucs230**

Shashank Bansal **19ucs049**

Link to Github code Repository

https://github.com/Devu-Goyal/NLP-Project-Round-1

Problem Description

Goto http://www.gutenberg.org

We have to download two considerably large books in text format Using Python do the following for both the books separately:

- Perform simple text pre-processing steps and tokenize the text T1 and T2 (Here T1 is textbook 1 and T2 is textbook 2)
- Analyzing the frequency distribution of tokens in T1 and T2 separately.
- Creating word clouds from the text before and after removing stopwords
- Evaluating relationship between word length and frequency
- Parts of Speech tagging for the words in the text

Python Libraries/Modules used:-

Matplotlib : for drawing plots

Python re library (regular expressions library): For regular expressions

NumPy :for parameters of axes while plotting graphs

Nltk :Used for tokenizing, removing stop words

Math :For calculating floor and ceil function values while plotting values

WordCloud :For creating word cloud

Collections :For getting the frequency mappings of the POS tags

Books chosen for applying the processing

T1: A Town Is Drowning

T2: Stained Glass Windows

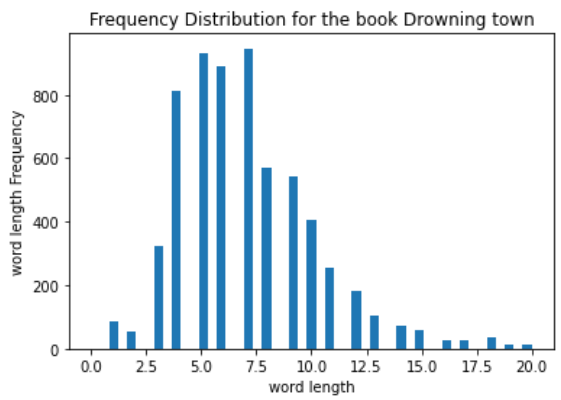
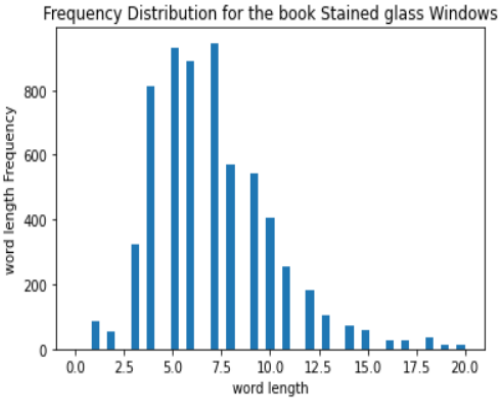
Inferences from raw data

Data Preprocessing and Preparation steps

We performed the following data preprocessing steps

1. Removing chapter number and chapter Headings
2. Removing all punctuation marks
3. Changing all text to lowercase
4. Removing numbers from text
5. Deleting white-spaces from text
6. Applying Lemmatization in the given text that provides us our final text ready to process.
7. Converting short forms like can't to actual representations
8. Tokenizing the text into a list of words
9. Removing chapter headings and unrelated data
10. Removing hyperlinks

Illustration : Word length – frequency plots

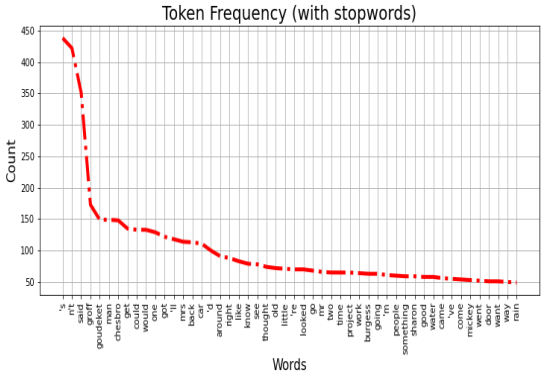
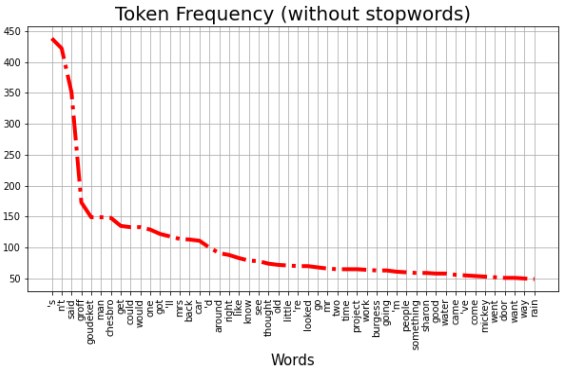
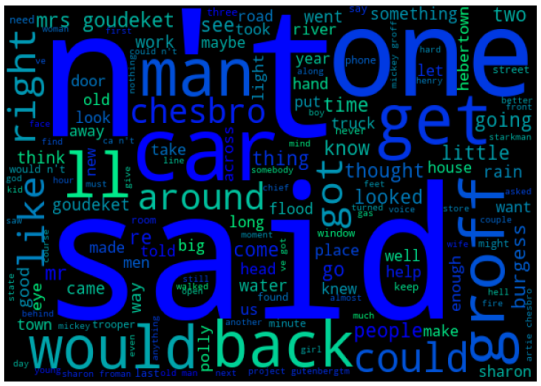
T1	T2
	

Inferences from word length- frequency plot

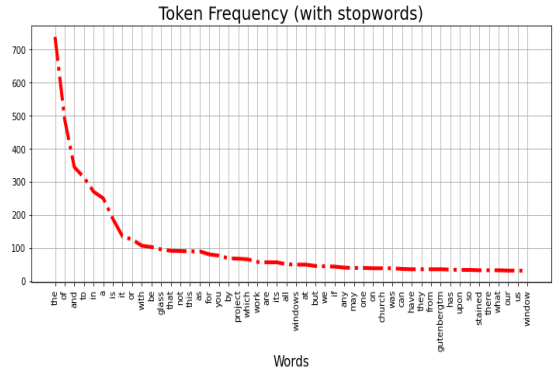
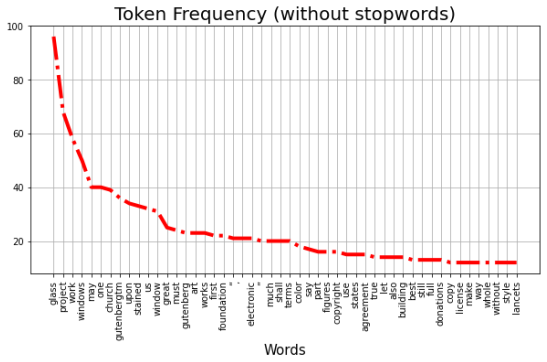
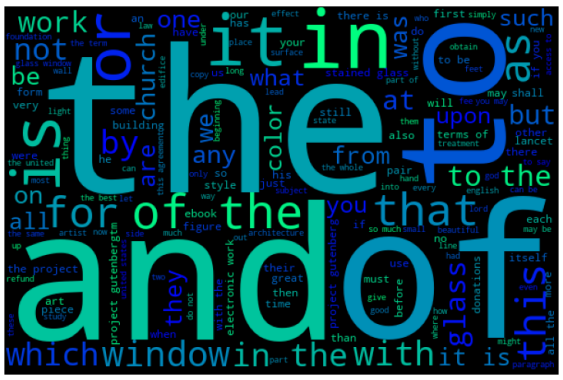
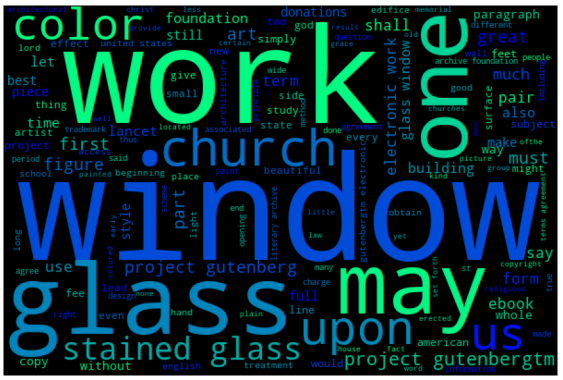
For both the books Words having length between 3 to 5 are the most frequently occurring words in these books . After that words with larger lengths (upto a certain length)are frequently followed by words of length 1 to2. Very long words appear very rarely .Overall implying that most of the words lie in the length range of 3 to 5.

Illustrations (Word Clouds and wordwise frequency plots)

T1

Before	After
	
	

T2

Before	After
	
	

Inference from word Clouds

The word clouds before and after removing stop-words are quite different due to the high frequency of many of these stopwords. One of the reasons may be that stop-words can be used in a variety of contexts whereas nouns and verbs are more restricted to the situations to which they relate to.

Sample results from pos_tagging from T1

```
[('novel', 'JJ'),
 ('takes', 'VBZ'),
 ('right', 'JJ'),
 ('heart', 'NN'),
 ('new', 'JJ'),
 ('flood', 'NN'),
 ('countrythe', 'NN'),
 ('northeast', 'NN'),
 ('united', 'JJ'),
 ('states', 'NNS'),
 ('generally', 'RB'),
 ('free', 'JJ'),
 ('hurricane sandy', 'NN'),
 ('attendant', 'NN'),
 ('floods', 'NNS'),
 ('disaster', 'NN'),
 ('struck', 'VBD'),
 ('onceterribleand', 'RB'),
 ('grim', 'JJ'),
 ('although', 'IN'),
 ('novel', 'JJ'),
 ('give', 'JJ'),
 ('accurate', 'NN'),
 ('brilliantly vivid', 'NN'),
 ('picture', 'NN'),
 ("'s", 'POS'),
 ('like', 'IN'),
 ('live', 'JJ'),
 ('flood', 'NN'),
 ('even', 'RB'),
 ('more importantly', 'RB'),
 ('show', 'VBP'),
 ('people', 'NNS'),
 ('like', 'IN'),
 ('fought', 'JJ'),
 ('the catastrophe', 'NN'),
 ('survived', 'VBD'),
```

('still', 'RB'),
('fighting', 'VBG'),
('the persons', 'NNS'),
('starkman', 'JJ'),
('burgess', 'NN'),
('groff', 'NN'),
('dynamic', 'JJ'),
('young', 'JJ'),
('executive sharon', 'NN'),
('shrewd', 'JJ'),
('opportunist', 'NN'),
('mrs', 'NN'),
('goudeket', 'NN')]

Sample results from pos_tagging from T2

[('project', 'NN'),
('gutenberg', 'NN'),
('ebook', 'NN'),
('stained', 'VBD'),
('glass', 'NN'),
('windows', 'NNS'),
('william frederic', 'VBP'),
('faber this', 'JJ'),
('ebook', 'NN'),
('use', 'NN'),
('anyone', 'NN'),
('anywhere', 'RB'),
('united', 'JJ'),
('states', 'NNS'),
('andmost', 'VBD'),
('parts', 'NNS'),
('world', 'NN'),
('cost', 'NN'),
('almost', 'RB'),
('restrictions whatsoever', 'NN'),
('may', 'MD'),
('copy', 'VB'),

('give', 'VB'),
('away', 'RP'),
('reuse', 'NN'),
('terms of', 'NN'),
('project', 'NN'),
('gutenberg', 'NN'),
('license', 'NN'),
('included', 'VBD'),
('ebook', 'JJ'),
('online', 'NN'),
('at www gutenberg org', 'NN'),
('located', 'VBN'),
('united', 'JJ'),
('states', 'NNS'),
('you will', 'RB'),
('check', 'VBP'),
('laws', 'NNS'),
('country', 'NN'),
('located', 'VBD'),
('before using', 'VBG'),
('ebook the', 'NN'),
('first', 'JJ'),
('edition', 'NN'),
('report', 'NN'),
('stained', 'VBD'),
('glass', 'NN'),
('windows', 'NNS'),
('gracechurch', 'VBP')]

Inferences POS_tagging

We are able to apply part of speech tagging to the words in these books using the penn treebank tagset. The pos_tag(words) function uses the penn treebank as the default tagset as per official documentation.

Conclusions

We completed the duties of word pre-processing which includes getting the text into simpler form by removing useless text , word tokenization , Word Cloud generation, POS tagging along with finding relation among the words and corresponding frequencies, and derived many conclusions about the books in Round 1 of our project while learning in the process.