Word Distribution Analysis

Shah Jainish, Palnitkar Saurav, Yenimireddy Lokesh Reddy
Engineering Data Science
University of Houston
jshah12@uh.edu, smpalnitkar@uh.edu, lyenimireddy@uh.edu

Abstract

Word distribution analysis is a fundamental task in natural language processing, and in this project, we explore the word distributions in a corpus of text files using Python libraries such as NLTK, OS, chardet, and re. We preprocess the text by converting it to lowercase, removing punctuations, non-alphabetic characters, and URLs. We then tokenize the text into words and remove stop words to compute the frequency distribution of the words in the corpus. To further analyze the corpus, we implement the concept of n-grams to compute the frequency distribution of two or more consecutive words in the text. We use NLTK's ngrams() function to generate a list of n-grams and FreqDist() to compute the frequency distribution of the n-grams. Finally, we print the 30 most common n-grams in the corpus. This project demonstrates how to use Python libraries to analyze word distributions and n-grams in a corpus of text files, which can be useful for more advanced natural language processing tasks, such as sentiment analysis, topic modeling, and text classification.

1. Introduction

Natural Language Processing (NLP) is a field of computer science and linguistics that focuses on the interaction between computers and human languages. Word distribution analysis is a fundamental task in NLP that involves counting the frequency of words in a text or collection of texts. This analysis is used for many NLP applications, such as topic modeling, sentiment analysis, and text classification.

This project demonstrates how to use Python libraries such as NLTK, OS, chardet, and re to compute the word distributions of a corpus of text files and implement n-grams analysis. We traverse the directory tree of the corpus using the OS library, read each file's content using chardet to detect character encoding, and clean the text using the re library. After cleaning, we use the NLTK library to tokenize the text, remove non-alphabetic words and stop words, and compute the frequency distribution of words using the FreqDist object. We also implement n-grams analysis to calculate the frequency of word pairs in the text, which provides insights into the corpus's language structure and patterns of word use.

The proposed method is straightforward to implement and can be extended or modified to suit specific requirements. The insights gained from this project can help in building better NLP models and improving the accuracy of NLP applications. By analyzing word distributions, we can gain a deeper understanding of the language used in the corpus and use this knowledge for various NLP tasks.

2. Methodology

Word distribution analysis is a statistical approach used to understand how frequently specific words occur within a text or corpus of texts. This analysis can provide valuable insights into the usage patterns and

relationships between different words within a given context.
Following is the flowchart for a clear understanding of the methodology used: -

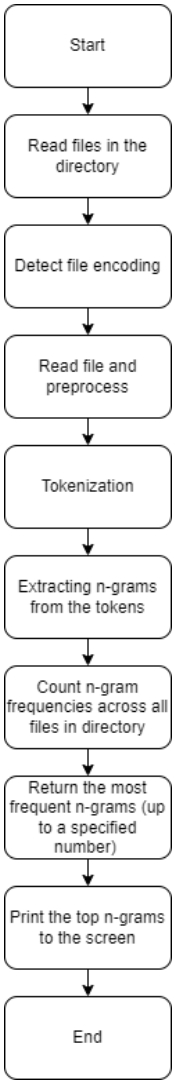

Figure 1: Flowchart of methodology

1. Reading files from a directory is a common task in data processing. It involves accessing and retrieving files stored in a specific folder on the computer. This can be done using file I/O functions provided by programming languages or libraries.

2. Detecting file encoding is important because it ensures that the text is read and processed correctly. Different file encodings are used for different languages, and not all encodings are compatible with all programs. The chardet.detect() function is a Python library that detects the encoding of a file based on its content.

3. Preprocessing involves cleaning and transforming the text data to make it suitable for further analysis. This can include converting text to lowercase, removing punctuation and non-alphabetic characters, and removing URLs. Lowercasing the text helps in treating the same words with different capitalizations as the same, while removing punctuation and non-alphabetic characters helps in simplifying the text and reducing noise. Removing URLs can be useful in text analysis if the URLs do not contain useful information.

4. Tokenization involves breaking up the text into smaller chunks, or tokens, such as words or phrases. This is a crucial step in natural language processing and allows for further analysis of the text data. Tokenization can be done using various libraries, such as NLTK or spaCy, depending on the language and specific requirements of the analysis.

5. N-grams are sequences of n consecutive tokens from the text. Extracting n-grams from the tokens allows for analysis of the frequency of specific sequences of words or phrases. This can be useful in tasks such as language modeling or sentiment analysis.

6. Counting the frequency of n-grams across all files in a directory involves iterating through all the files, preprocessing the text, tokenizing it, and extracting the n-grams. The n-gram frequencies are then counted and stored in a data structure such as a dictionary or a pandas data frame.

7. Returning the most frequent n-grams involves selecting the n-grams with the highest frequency counts. This can be done using sorting or filtering techniques, depending on the specific requirements of the analysis. The most frequent n-grams can provide insights into the language and topics discussed in the text data.

3. Experimental Results

3.1 Word Frequency Distribution

Word Frequency Distribution refers to the way in which words are counted and analyzed based on their frequency of occurrence within a given text or corpus. To compute word frequency distributions, we have followed the methodology shown in Fig 1. Below is the word cloud for 30 most common words in both the corpus before the stop words are removed.

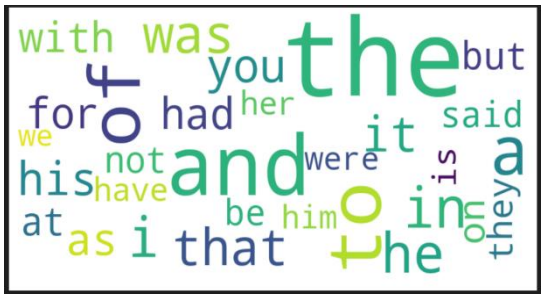

Figure 2: Word Cloud

Fig 2 shows that the word “the” occurs the most frequently, followed by the words “and”, “to” and “of”. It is difficult to determine the topic from these words because these words are essential for the structure and meaning of sentences, but they don’t carry much specific meaning on their own. Therefore, a more detailed analysis of the corpus is needed to fully understand its characteristics and meaning.

3.2 Analysis of n-grams

Analysis of n-grams can provide a more nuanced understanding of language use, by considering the context in which words or phrases appear. Here, we have computed the frequency of most common Bigrams(n=2) for both the documents.

Table 1: Most frequent Bigrams in corpus.

Bigram	Frequency
'of', 'the'	70000
'in', 'the'	43000
'to', 'the'	33000
‘and’ 'the'	23000
'it', 'was'	22000
'on', 'the'	20000
'to', 'be'	19500

From Table 1, we can see that many of the most common bigrams involve prepositions like "of", "in", "to", "on", and "at", indicating that these words are frequently used to connect other words or phrases. This information can be useful for understanding the context and patterns of language use in the text corpus being analyzed. For example, if the text corpus is a collection of news articles, the high frequency of certain Bi-grams like "in the" and "of the" may indicate that the articles are primarily focused on a particular topic or subject.

3.3 Word Frequency Distribution and Bi-gram analysis after removal of stop words

After removing the stop words from the corpus, we performed the word frequency distribution for Unigrams and Bigrams for individual corpus to get the insights from them. Below are the insights for each corpus.

3.3.1 Insight from Unigrams from Corpus1

The most frequent word in the corpus is "said," which suggests that it may involve a lot of dialogue. Words like "one," "two," and "men" suggest that there may be a focus on individuals and their interactions with others. The frequent use of "would," "could," and "don’t" suggest that there may be a focus on possibilities and limitations. The use of words like "little," "good," and "great" suggests that the corpus may involve descriptions of characters, settings, or events. Words like "time," "way," and "back" suggest that there may be a focus on journeys or time periods. The use of "know," "think," and "say" suggests that there may be a focus on characters sharing their thoughts and opinions. The frequent use of "made" suggests that there may be a focus on actions taken by characters.

3.3.2 Insight from Bigrams from Corpus1

Based on the most common bigram words in the list, it seems like the content could be related to children's literature, folktales, or fables. The words "Brer Rabbit" and "Brer Fox" are both characters from African American folklore, while "Uncle Remus" is a

collection of animal tales from the southern United States. The use of phrases such as "let us", "said Captain", "said Doctor", and "said Uncle" suggest that the content might be dialogue-heavy, potentially in a storytelling or conversational format. Additionally, the use of phrases like "next morning", "next day", and "one day" indicate that the content could be structured around a series of events or episodes. Moreover, the content also talks about the "project gutenberg". Overall, the content appears to be aimed towards a young audience, possibly with an educational or moralistic purpose.

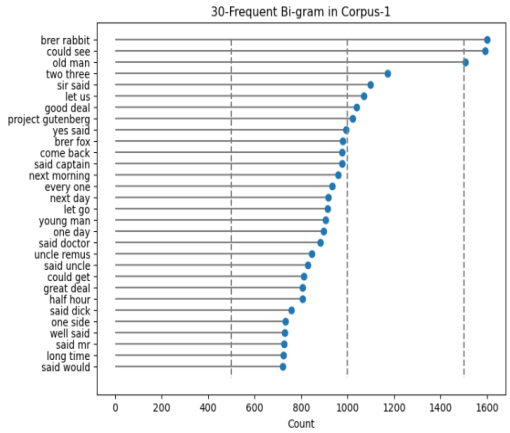

Figure 3: Lollipop Chart

3.3.3 Insight from Unigrams from Corpus2

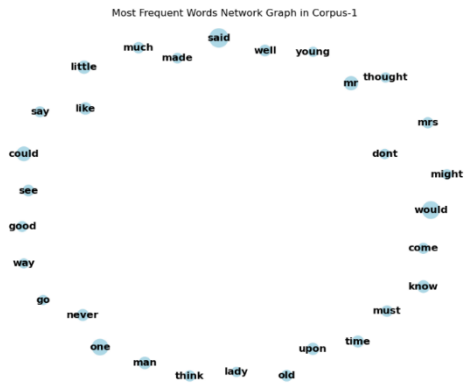

Figure 4: Network Graph

Fig 4 shows A network graph identifies words that are closely related based on how frequently they appear together and provides information about corpus structure. It seems that the corpus may contain a lot of dialogue, as the most frequent word is "said." Corpus may also involve speculation and possibilities, as words like "would," "could," and "might" are also highly frequent. Other common words like "one," "man," and "woman" suggest that the corpus may involve characters and their interactions. The use of words like "little" and "old" suggests that the content may involve descriptions of characters or settings. Additionally, the frequent use of "know," "think," and "said" suggests that the corpus may involve characters sharing their thoughts and opinions.

3.3.4 Insight from Bigrams from Corpus2

Based on the most common bigram words in the list, the phrase “don’t know” appears the most frequently in the corpus, suggesting uncertainty or confusion in the text. The combination of “young man” and “young lady” are frequent, suggesting that the corpus may contain works that depict youthful characters in romantic or social situations. The phrases "said mr" and "said mrs" appear frequently, suggesting that the corpus may contain works that heavily feature dialogue. The phrases "said mr" and "said mrs" appear frequently, suggesting that the corpus may contain works that heavily feature dialogue. Overall, the content an uncertainty, youthful characters in romantic or social situations, add-ons to books or magazines, heavy dialogue, multi-volume works, visual scenes or imagery, travel, or journeys, leaving and returning, and characters affirming or agreeing.

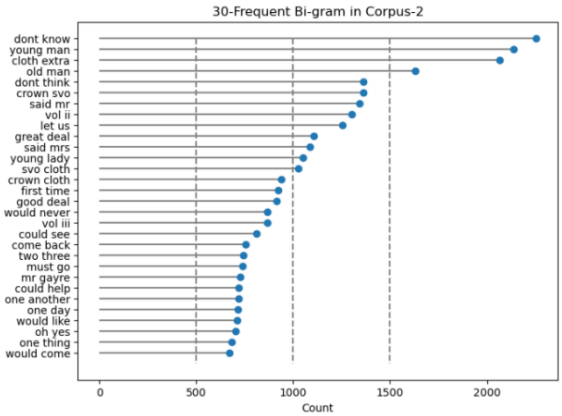

Figure 5: Lollipop Chart

Fig 5 depicts the top 30 bi-gram words in a text document. Each word is represented by the y-

axis, and the number of times each word appears in the document is represented by the x-axis. The vertical lines, or 'lollipops,' represent the word count. The most common words in English language text are 'dont know,' 'young man,' 'cloth extra,' 'old man,' and 'dont thing,' which are all common bi-gram words. This indicates that the text document is written in a standard style with few uncommon or specialized words.

4. GitHub link for project code

Repository Link:
CIS-6397-Textmining-Spring-2023/miniproject1-miniproject1_group-13: miniproject1-miniproject1_group-13 created by GitHub Classroom

5. Conclusion

This project focuses on demonstrating the use of Python libraries such as NLTK, OS, chardet, and re to perform word distribution analysis and n-gram analysis on a corpus of text files. The project explains the importance of word distribution analysis in NLP and its applications, such as topic modeling. The project explains the steps involved in cleaning the text, tokenizing, removing non-alphabetic words and stop words, and computing the frequency distribution of words. The implementation of n-grams analysis helps in identifying patterns of word use, language structure and it suggests that the material seems to target a youthful readership, potentially with the intention of imparting knowledge or values, youthful characters in romantic or social situation and the overall content might be taken from the books or magazines. Additional investigation could be conducted by utilizing more advanced n-gram models or alternative techniques to acquire deeper comprehension of the text's composition and meaning.

6. Appendix

6.1 List of stop words used in this project

{'while', 'then', 'now', 've', 'theirs', 'them', "haven't", "should've", 'was', 'yourself', 's', "wasn't", 'he', 'through', 'when', 'ourselves', 'into', 'until', 'had', 'down', 'ain', 'same', 'just', 'were', 'all', 'herself', 'i', 'm', 'at', 'not', 'where', 'are', 'his', 'won', 'were', 'nor', "you've", "you're", 'here', "needn't", "hadn't", 'own', 'which', 'only', 'we', 'than', "it's", 'in', 're', 'once', 'a', 'was', 'do', 'some', "shan't", 'does', 'him', 'did', 'with', 'from', 'so', 'an', 'because', 'other', 'for', 'me', 'they', 'about', 'had', 'such', 'out', 'am', 'the', 'their', 'up', 'this', 'haven', "won't", "didn't", 'would', 'doing', 'to', 'is', 'can', 'between', 'by', 'again', 'should', 'yourselves', 'more', 'myself', 'what', 'has', 'ma', 'before', "weren't", 'does', 'below', 'itself', 'been', 'and', 'why', 'll', 'over', 'after', 'whom', 'o', 'too', 'shouldn't', 'you', 'd', 'our', 'being', 'himself', 'any', 'if', 'or', "you'd", "isn't", 'but', 'its', "you'll", 'couldn't', 'it', "shouldn't", "don't", 'mustn', 'might', 'having', 'as', 'off', 'your', 'didn', 'needn', 'against', 'how', 'will', 'themselves', 'shan', "mightn't", 'during', 'hasn', 'be', 'no', 'each', 'on', 'she', 'who', 'are', 'of', 't', 'that', 'don', 'y', "aren't", 'most', 'these', 'both', 'above', "wouldn't", "that'll", 'under', 'very', "couldn't", 'have', 'isn', 'her', 'my', "she's", "doesn't", 'further', 'those', "mustn't", "hasn't", 'yours', 'hers', 'ours', 'there', 'few'}

7. Author Credit Statement

Jainish Shah: - Software, Methodology, Conceptualization, Project Administration, Writing – review & editing.

Saurav Palnitkar: - Data Curation, Writing – original draft, Visualization, Validation

Yenimireddy Lokesh Reddy: - Formal Analysis, Software, Writing – review & editing, Visualization, Resources