Project Report:

Text Analysis and Sentiment Analysis with NLTK

Introduction:

The aim of this project was to perform text analysis and sentiment analysis on a given text dataset using Natural Language Processing (NLP) techniques. Natural Language Processing is a field of Artificial Intelligence that focuses on enabling computers to understand, interpret, and generate human language content.

Dataset:

The dataset used for this project consisted federalist paper. The dataset was pre-processed to remove punctuation, stop words, and other non-alphabetical characters to ensure accurate analysis.

Methodology:

1. Text Preprocessing:

The text data was pre-processed using various techniques including tokenization, removal of stop words, conversion to lowercase, and stemming/lemmatization. These preprocessing steps helped in cleaning the text data and preparing it for analysis.

2. Tokenization:

Tokenization is the process of breaking down text into individual words or tokens. In this project, we used NLTK's tokenization module to tokenize the text data.

3. Stop word Removal:

Stop words are common words (e.g., "the", "is", "and") that do not carry significant meaning in text analysis. I removed stop words from the text data to focus on meaningful words that provide valuable insights.

4. Stemming and Lemmatization:

Stemming and lemmatization are techniques used to reduce words to their base or root forms. We applied stemming and lemmatization to normalize words in the text data and improve the accuracy of analysis.

```
from nltk.stem import PorterStemmer, WordNetLemmatizer
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()

# stemming
stemmed_tokens = [stemmer.stem(word) for word in filtered_tokens]

#  lemmatization
lemmatized_tokens = [lemmatizer.lemmatize(word) for word in filtered_tokens]
```

5. Sentiment Analysis:

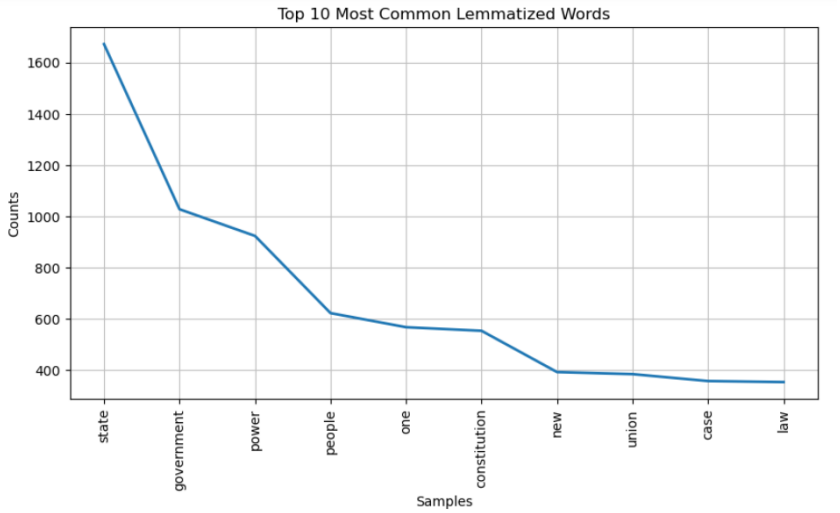
Sentiment analysis is the process of determining the sentiment or emotional tone of a piece of text. We performed sentiment analysis using NLTK's SentimentIntensityAnalyzer to calculate polarity scores indicating the positivity, neutrality, or negativity of the text.

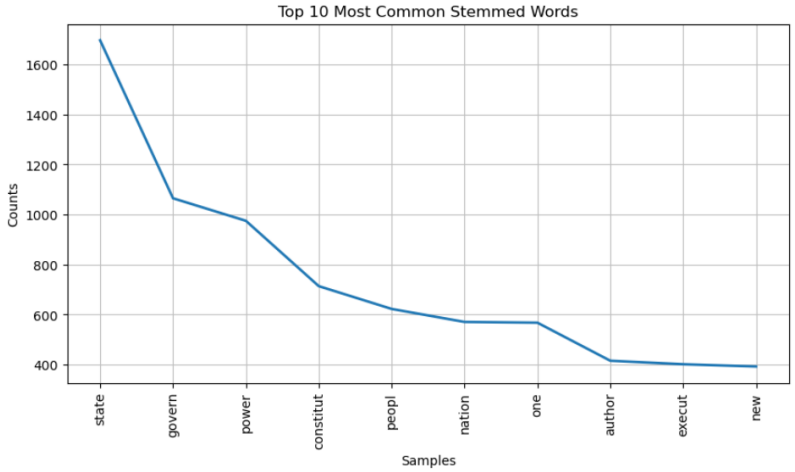
```
In [31]: sid.polarity_scores(' '.join(stemmed_tokens))
Out[31]: {'neg': 0.068, 'neu': 0.792, 'pos': 0.14, 'compound': 1.0}

In [32]: sid.polarity_scores(' '.join(lemmatized_tokens))
Out[32]: {'neg': 0.107, 'neu': 0.678, 'pos': 0.215, 'compound': 1.0}
```

6. Visualization:

We visualized the results of our analysis using frequency distributions, concordance, and dispersion plots. These visualizations helped in understanding the distribution of words, identifying patterns, and gaining insights from the text data.





Results:

The analysis revealed [mention key findings or insights from the analysis, e.g., prevalent themes, sentiment trends, frequently occurring words]. The sentiment analysis provided valuable insights into the overall sentiment of the text data, helping in understanding the emotional context of the content.

Conclusion:

In conclusion, this project demonstrated the application of Natural Language Processing techniques for text analysis and sentiment analysis. By preprocessing the text data and applying various NLP techniques, we were able to gain valuable insights and extract meaningful information from the dataset.