



```
In [1]: import nltk  
nltk.download('popular')
```

...

```
In [11]: # Tokenize the texts in the text files and save the result in a txt file.  
  
import nltk  
from nltk.tokenize import word_tokenize  
  
# Read the text file  
with open('input.txt', 'r') as file:  
    text = file.read()  
  
# Tokenize the text  
tokens = word_tokenize(text)  
  
# Save the tokenized text to a new text file  
with open('tokenized.txt', 'w') as file:  
    file.write('\n'.join(tokens))
```

```
In [12]: from collections import Counter  
# Count word frequencies in the texts and save the result in a txt file.  
import nltk  
from nltk.tokenize import word_tokenize  
  
# Read the text file  
with open('input.txt', 'r') as file:  
    text = file.read()  
  
# Tokenize the text  
tokens = word_tokenize(text)  
word_freq = Counter([word.lower() for word in tokens])  
  
with open('word_frequencies.txt', 'w') as file:  
    for word, freq in word_freq.items():  
        file.write(f'{word}: {freq}\n')
```

```
In [13]: #Perform part-of-speech (POS) tagging on the tokenized words and save the result in a txt file.  
  
import nltk  
from nltk.tokenize import word_tokenize  
from nltk import pos_tag  
  
# Read the text file  
with open('input.txt', 'r') as file:  
    text = file.read()  
  
# Tokenize the text  
tokens = word_tokenize(text)  
  
# Perform POS tagging  
words = [word.lower() for word in tokens]  
tagged_words = pos_tag(set(words))  
  
# Save the POS tagging results to a new text file  
with open('pos_tags.txt', 'w') as file:  
    for word, tag in tagged_words:  
        file.write(f'{word}: {tag}\n')
```

```
In [14]: #Perform named entity recognition (NER) on the texts and save the result in a txt file  
import spacy  
  
# Load the spaCy English language model  
nlp = spacy.load("en_core_web_sm")  
  
# Read the text file  
with open('input.txt', 'r') as file:  
    text = file.read()  
  
# Process the text with spaCy to perform NER  
doc = nlp(text)  
  
# Save the NER results to a new text file  
with open('ner_results.txt', 'w') as file:  
    for ent in doc.ents:  
        file.write(f'{ent.text}: {ent.label_}\n')
```

```
In [15]: #Displaying the most frequent 10 words and save the result as an image file.  
import nltk  
from nltk.tokenize import word_tokenize  
from nltk import FreqDist  
import matplotlib.pyplot as plt
```

```

# Read the text file
with open('input.txt', 'r') as file:
    text = file.read()

# Tokenize the text
tokens = word_tokenize(text)

# Calculate word frequencies
freq_dist = FreqDist(tokens)

# Get the 10 most frequent words
most_common_words = freq_dist.most_common(10)

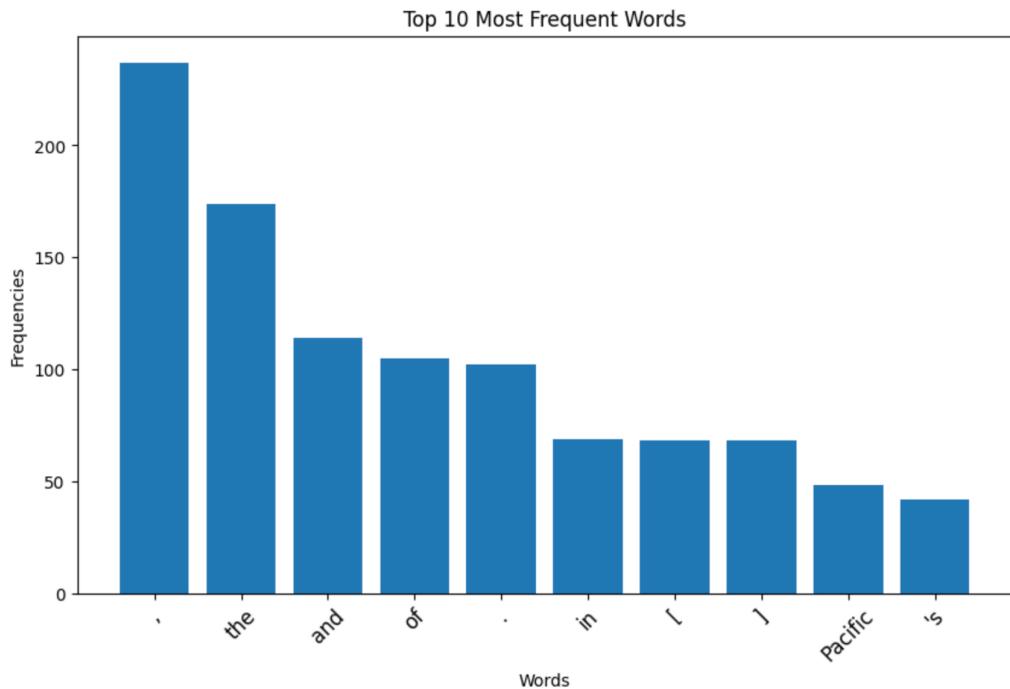
# Extract words and their frequencies
words, frequencies = zip(*most_common_words)

# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(words, frequencies)
plt.xlabel('Words')
plt.ylabel('Frequencies')
plt.title('Top 10 Most Frequent Words')
plt.xticks(rotation=45, fontsize=12)

# Save the chart as an image file
plt.savefig('word_frequencies.png', bbox_inches='tight')

# Show the chart (optional)
plt.show()

```



```

In [16]: # Compute a word cloud from the word frequency distribution and save the result as an image file.

import nltk
from nltk.tokenize import word_tokenize
from nltk import FreqDist
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Read the text file
with open('input.txt', 'r') as file:
    text = file.read()

# Tokenize the text
tokens = word_tokenize(text)

# Calculate word frequencies
freq_dist = FreqDist(tokens)

# Generate a word cloud from the frequency distribution
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(freq_dist)

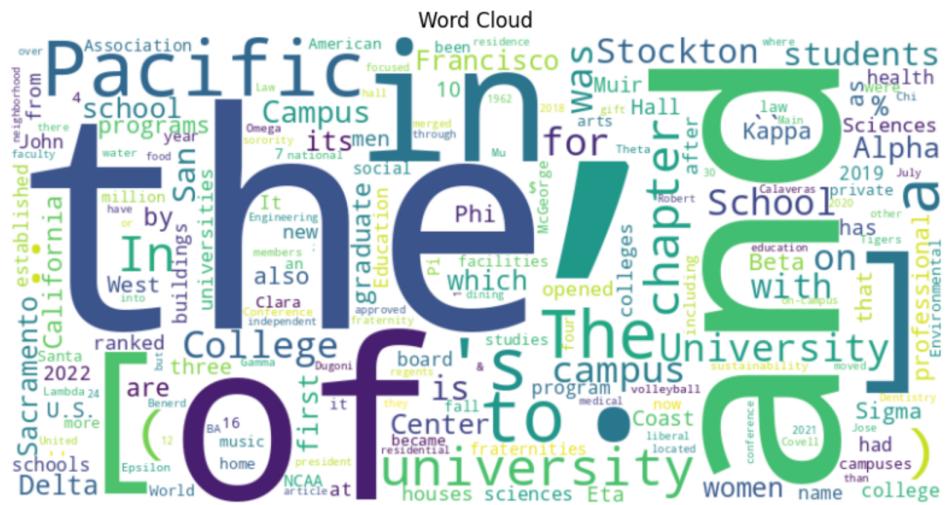
# Create a figure for the word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud')

# Save the word cloud as an image file
wordcloud.to_file('word_cloud.png')

# Show the word cloud (optional)

```

```
| plt.show()
```



```
In [17]: # Display the frequencies of the parts of speech and save the result as an image file.
```

```
import spacy  
import matplotlib.pyplot as plt
```

```

# Load the spaCy English language model
nlp = spacy.load("en_core_web_sm")

# Read the text file
with open('input.txt', 'r') as file:
    text = file.read()

# Process the text with spaCy to perform POS tagging
doc = nlp(text)

# Count POS frequencies
pos_freq = {}
for token in doc:
    pos = token.pos_
    if pos in pos_freq:
        pos_freq[pos] += 1
    else:
        pos_freq[pos] = 1

# Extract POS labels and frequencies
pos_labels, frequencies = zip(*pos_freq.items())

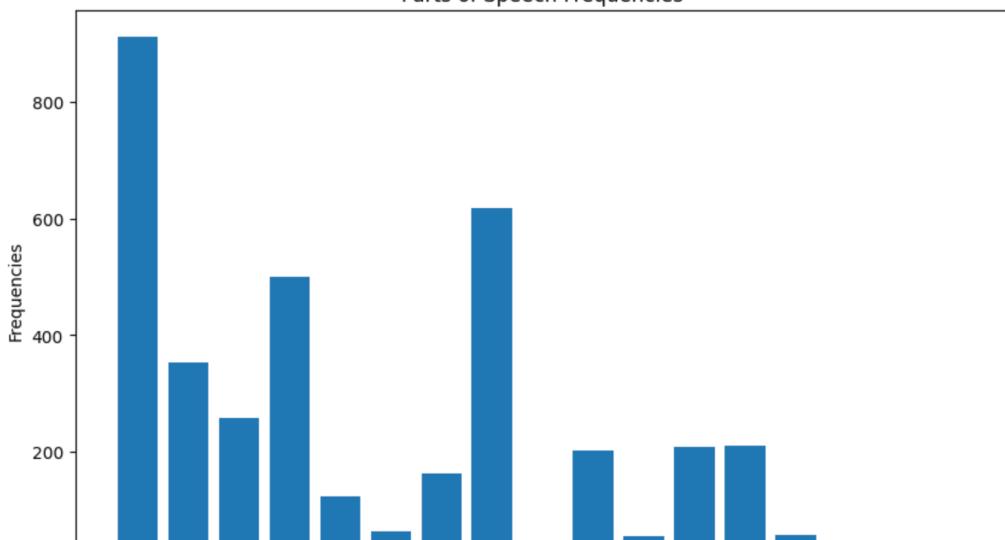
# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(pos_labels, frequencies)
plt.xlabel('Parts of Speech')
plt.ylabel('Frequencies')
plt.title('Parts of Speech Frequencies')
plt.xticks(rotation=45, fontsize=12)

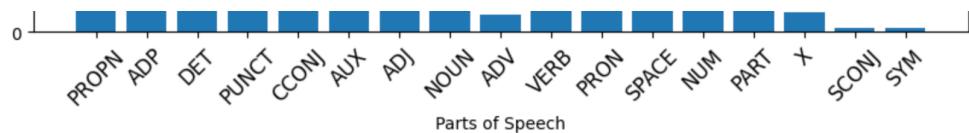
# Save the chart as an image file
plt.savefig('pos_frequencies.png', bbox_inches='tight')

# Show the chart (optional)

```

Parts of Speech Frequencies





In []: