```
import nltk
from nltk.corpus import brown
from nltk.tokenize import sent_tokenize, word_tokenize

# Download required NLTK data (only once)
nltk.download('brown')
nltk.download('punkt')
nltk.download('punkt_tab') # Download punkt_tab resource

# Load text from Brown corpus (news category)
text = brown.raw(categories='news')

# Convert into sentences
sentences = sent_tokenize(text)

# Take first sentence for demonstration
first_sentence = sentences[0]
print("First Sentence:\n", first_sentence)

# Tokenize into words
words = word_tokenize(first_sentence)
print("\nTokenized Words:\n", words)
```

```
First Sentence:

 The/at Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd Friday/nr an/at investigation/nn of/in Atlanta's/np$ recent

Tokenized Words:
 ['The/at', 'Fulton/np-tl', 'County/nn-tl', 'Grand/jj-tl', 'Jury/nn-tl', 'said/vbd', 'Friday/nr', 'an/at', 'investigation/nn', 'of/i
[nltk_data] Downloading package brown to /root/nltk_data...
[nltk_data]   Package brown is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
```

```
from nltk.corpus import stopwords
from nltk.probability import FreqDist

nltk.download('stopwords')

# Tokenize all words from news category
all_words = word_tokenize(text.lower())

# Frequency distribution
fdist = FreqDist(all_words)

# Remove English stopwords
stop_words = set(stopwords.words("english"))
filtered_words = [w for w in all_words if w.isalpha() and w not in stop_words]

# Top 20 words after removing stopwords
fdist_filtered = FreqDist(filtered_words)
print("Top 20 words without stopwords:\n", fdist_filtered.most_common(20))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Top 20 words without stopwords:
 [('u', 1)]
```

```
from nltk.stem import SnowballStemmer

stemmer = SnowballStemmer("english")

# Apply stemming
stemmed_words = [stemmer.stem(w) for w in filtered_words]

# Frequency of stemmed words
fdist_stem = FreqDist(stemmed_words)
print("\nTop 20 Stemmed Words:\n", fdist_stem.most_common(20))
```

```

Top 20 Stemmed Words:
 [('u', 1)]
```

```
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')
```

```
nltk.download('omw-1.4')

lemmatizer = WordNetLemmatizer()

# Apply lemmatization
lemmatized_words = [lemmatizer.lemmatize(w) for w in filtered_words]

# Frequency of lemmatized words
fdist_lemma = FreqDist(lemmatized_words)
print("\nTop 20 Lemmatized Words:\n", fdist_lemma.most_common(20))
```

```
Top 20 Lemmatized Words:
 [('u', 1)]
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# WordCloud before removing stopwords
wordcloud_before = WordCloud(width=800, height=400, background_color='white').generate(" ".join(all_words))

# WordCloud after removing stopwords
wordcloud_after = WordCloud(width=800, height=400, background_color='white').generate(" ".join(filtered_words))

# Plot both wordclouds
plt.figure(figsize=(12,6))

plt.subplot(1,2,1)
plt.title("Before Stopword Removal")
plt.imshow(wordcloud_before, interpolation='bilinear')
plt.axis("off")

plt.subplot(1,2,2)
plt.title("After Stopword Removal")
plt.imshow(wordcloud_after, interpolation='bilinear')
plt.axis("off")

plt.show()
```