

Stop word elimination:

```
import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

nltk.download('punkt')

nltk.download('stopwords')

def remove_stopwords(text):

    stop_words = set(stopwords.words('english'))

    word_tokens = word_tokenize(text)

    filtered_text = [word for word in word_tokens if word.lower() not in
stop_words]

    return ' '.join(filtered_text)

text = "This is an example sentence demonstrating stop word removal."

filtered_text = remove_stopwords(text)

print(filtered_text)
```

Stemming:

```
import nltk

from nltk.stem import PorterStemmer

from nltk.tokenize import word_tokenize

# Input from the user

text = input("Enter a sentence for stemming: ")

# Tokenization

words = word_tokenize(text)

# Stemming

stemmer = PorterStemmer()

stemmed_words = [stemmer.stem(word) for word in words]

print("\nAfter Stemming:")

print(stemmed_words)
```

Lemmatization:

```
import nltk

from nltk.stem import WordNetLemmatizer

# Define a text string

text = "This is a sample text. It contains some words that we can use for
lemmatization."

# Tokenize the text into individual words

tokens = nltk.word_tokenize(text)

# Create a WordNetLemmatizer object

lemmatizer = WordNetLemmatizer()

# Lemmatize each word and print the result

for token in tokens:

    lemma = lemmatizer.lemmatize(token)

    print(token, "-->", lemma)
```

POS Tagging:

```
import nltk

from nltk.tokenize import word_tokenize

nltk.download('punkt')

nltk.download('averaged_perceptron_tagger')

text1 = "children are innocent."

tokens = word_tokenize(text1)

tagged_words = nltk.pos_tag(tokens)

print(tagged_words)

print()

text2 = "Visiting aunts can be a nuisance"

tokens = nltk.word_tokenize(text2)

tagged_words = nltk.pos_tag(tokens)

print(tagged_words)

print()
```

Lexical Analysis:

```
import nltk

from nltk.tokenize import word_tokenize, sent_tokenize

# Download required resources if not already present
nltk.download('punkt')

text = "This is a sample text for lexical analysis using NLTK."

# Sentence Tokenization
sentences = sent_tokenize(text)

print("Sentences:", sentences)

# Word Tokenization
words = word_tokenize(text)

print("Words:", words)

# Part-of-Speech Tagging
nltk.download('averaged_perceptron_tagger')

tagged_words = nltk.pos_tag(words)

print("Tagged Words:", tagged_words)
```