Computational Linguistics & NLP Lab 2

Implementing:

- **Tokenization**
- **Stemming**
- **Lemmatization**
- **Stop word removal**

Code:

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords
from nltk import pos_tag
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
```

Output:

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```
def preprocess_text(text):
    # Tokenization
    tokens = word_tokenize(text.lower())  # Convert to lowercase for
consistency

    # Removing stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token not in stop_words]

    # Stemming
    stemmer = PorterStemmer()
    stemmed_tokens = [stemmer.stem(token) for token in tokens]

    # Lemmatization
    lemmatizer = WordNetLemmatizer()
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in
tokens]

    return tokens, stemmed_tokens, lemmatized_tokens
```

```
input_text = "Tokenization is an important step in natural language
processing. It involves stemming and lemmatization."
tokens, stemmed_tokens, lemmatized_tokens = preprocess_text(input_text)
print("Original Tokens:", tokens)
print("Stemmed Tokens:", stemmed_tokens)
print("Lemmatized Tokens:", lemmatized_tokens)
```

Output:

```
Original Tokens: ['tokenization', 'important', 'step', 'natural', 'language', 'processing', '.', 'involves', 'stemming', 'lemmatization', '.']
Stemmed Tokens: ['token', 'import', 'step', 'natur', 'languag', 'process', '.', 'involv', 'stem', 'lemmat', '.']
Lemmatized Tokens: ['tokenization', 'important', 'step', 'natural', 'language', 'processing', '.', 'involves', 'stemming', 'lemmatization', '.']
```

```
def pos_tagging(text):
    tokens = word_tokenize(text)
    pos_tags = pos_tag(tokens)
    return pos_tags

tags = pos_tagging(input_text)

print("POS Tags:", tags)
```

Output:

```
POS Tags: [('POS', 'NNP'), ('tagging', 'VBG'), ('helps', 'VBZ'), ('identify', 'VB'), ('the', 'DT'), ('grammatical', 'JJ'), ('parts', 'NNS'), ('of', 'IN'), ('speech', 'NN'), ('in', 'IN
```