**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

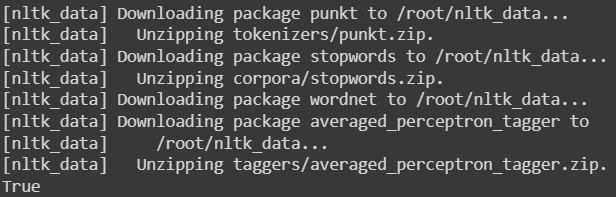
nltk.download('punkt')

nltk.download('stopwords')

nltk.download('wordnet')

nltk.download('averaged\_perceptron\_tagger')

**Output:**

****

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."

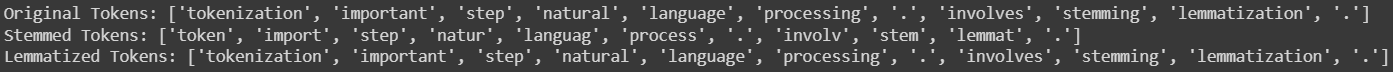
tokens, stemmed\_tokens, lemmatized\_tokens = preprocess\_text(input\_text)

print("Original Tokens:", tokens)

print("Stemmed Tokens:", stemmed\_tokens)

print("Lemmatized Tokens:", lemmatized\_tokens)

**Output:**



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:**

