Lab 03: Introduction to NLTK (Natural Language Toolkit)

1. NLTK

NLTK (Natural Language Toolkit) is a powerful Python library used for working with human language data (text). It provides easy-to-use interfaces for over 50 corpora and lexical resources, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Importance of NLTK:

- Helps in understanding and processing natural languages using Python.
- Enables rapid prototyping of NLP applications.
- Widely used in academia, research, and commercial NLP solutions.
- Supports integration with machine learning libraries for advanced NLP tasks.

2. Lab Objectives

- Understand and install NLTK
- Tokenize text into words and sentences
- Perform stemming and lemmatization
- Remove stopwords
- Explore part-of-speech tagging
- Do basic Named Entity Recognition (NER)

3. Prerequisites

- Python 3 installed
- Jupyter Notebook / Google Colab / Python IDE
- nltk library installed (`pip install nltk`)

4. Lab Tasks

Task 1: Install and Import NLTK

This step initializes and downloads all necessary datasets and tokenizers required by NLTK for text processing.

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
nltk.download('punkt_tab')
nltk.download('maxent_ne_chunker_tab')
```

Task 2: Sentence and Word Tokenization

Tokenization is the process of breaking text into sentences or words. It's the first step in most NLP tasks.

```
from nltk.tokenize import sent_tokenize, word_tokenize

text = "Natural Language Processing is a field of Artificial Intelligence. It helps computers understand human language."

# Sentence Tokenization
sentences = sent_tokenize(text)
print("Sentences:", sentences)

# Word Tokenization
words = word_tokenize(text)
print("Words:", words)
```

Task 3: Remove Stopwords

Stopwords are common words like 'is', 'the', and 'and' that are usually removed to focus on meaningful words.

```
from nltk.corpus import stopwords

stop_words = set(stopwords.words("english"))
filtered_words = [w for w in words if w.lower() not in stop_words]
print("Filtered Words:", filtered_words)
```

Task 4: Stemming

Stemming is the process of reducing words to their root form. For example, 'running' becomes 'run'.

```
from nltk.stem import PorterStemmer

ps = PorterStemmer()
stemmed_words = [ps.stem(w) for w in filtered_words]
print("Stemmed Words:", stemmed_words)
```

Task 5: Lemmatization

Lemmatization also reduces words to their base form, but with dictionary knowledge. It's more accurate than stemming.

```
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(w) for w in filtered_words]
print("Lemmatized Words:", lemmatized_words)
```

Task 6: Part-of-Speech (POS) Tagging

POS tagging labels each word with its part of speech, such as noun, verb, adjective, etc.

```
pos_tags = nltk.pos_tag(filtered_words)
print("POS Tags:", pos_tags)
```

Task 7: Named Entity Recognition (NER)

NER locates and classifies named entities like persons, organizations, and locations in text.

```
from nltk import ne_chunk
from nltk import pos_tag # Import pos_tag

# Perform POS tagging on the filtered words
pos_tags = pos_tag(filtered_words)

ner_tree = ne_chunk(pos_tags)
print("Named Entities:")
print(ner_tree)
```

5. Activity: Analyze a Paragraph

Use any paragraph from a news article or Wikipedia and:

1. Tokenize it.
2. Remove stopwords.
3. Stem or lemmatize.
4. Do POS tagging.

5. Perform NER.

6. Submission

- Upload a .ipynb or .py file with your code.
  Add screenshots of the outputs.
  Submit to LMS on or before 20/06/2025