Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2025-26

Class:	**BE-AI&DS**	**Semester:**	**VII**
Course Code:	**CSDOL7011**	**Course Name:**	**NLP Lab**

Name of Student:	BARI ANKIT VINOD
Roll No. :	61
Experiment No.:	**6**
Title of the Experiment:	**Performing Chunking and Named Entity Recognition using NLTK**
Date of Performance:	
Date of Submission:	

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1

CSDOL7011: Natural Language Processing Lab

Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty :

Signature :

Date :

Aim: To identify and extract syntactic phrases (chunks) and named entities from text using NLTK's chunking and NER functionalities.

Objective: To extract syntactic chunks and named entities using chunking and Named Entity Recognition techniques.

Tools Required:

1. Python (Jupyter Notebook or Google Colab)
2. nltk

Procedure:

1. Import required libraries:
    a. import nltk
    b. nltk.download('punkt')
    c. nltk.download('averaged_perceptron_tagger')
    d. nltk.download('maxent_ne_chunker')
    e. nltk.download('words')

2. Input or define a sentence:

Example: "Barack Obama was born in Hawaii and served as the 44th President of the United States."

3. Tokenize and POS-tag the sentence:
    a. tokens = nltk.word_tokenize(sentence)
    b. pos_tags = nltk.pos_tag(tokens)

4. Apply chunking:

Use regular expressions to define grammar rules.

    a. chunk_grammar = "NP: {<DT>?<JJ>*<NN>}"
    b. chunk_parser = nltk.RegexpParser(chunk_grammar)
    c. chunked = chunk_parser.parse(pos_tags)
    d. chunked.draw() # Optional: visualize the parse tree

5. Perform Named Entity Recognition:
    a. ner_tree = nltk.ne_chunk(pos_tags)
    b. ner_tree.draw() # Optional visualization

6. Extract named entities:

Traverse the NER tree and extract named entities like PERSON, ORGANIZATION, LOCATION.

Description of the Experiment:

In this experiment, students will implement chunking to group words into syntactic units (like noun phrases), and perform Named Entity Recognition (NER) to identify proper nouns such as names of people, places, or organizations. These are foundational steps in syntactic and information extraction tasks.

Detailed Description of the NLP Technique:

1. Chunking (Shallow Parsing):

Chunking segments and labels multi-token sequences, such as noun phrases (NP) or verb phrases (VP), without generating full parse trees.

Example:

a. Input: "The quick brown fox"
b. POS tags: [(The, DT), (quick, JJ), (brown, JJ), (fox, NN)]
c. Chunk: (NP The quick brown fox)

Uses regular expressions on POS tag sequences to define chunk patterns.

2. Named Entity Recognition (NER):

NER identifies and classifies named entities in text into predefined categories such as:

PERSON (e.g., "Barack Obama")

ORGANIZATION (e.g., "Google")

LOCATION (e.g., "India")

DATE, TIME, MONEY, etc.

NLTK's ne_chunk() uses a pre-trained Maximum Entropy classifier to identify named entities.

Importance of Chunking and NER:

a. Enhances understanding of text structure.
b. Crucial in tasks like question answering, information extraction, and document classification.

Conclusion:

The results show that NLTK's Named Entity Recognition (NER) successfully identifies and classifies entities such as names of people, organizations, and locations from text. This demonstrates NLTK's ability to extract meaningful information and structure unorganized textual data effectively.