Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

AY: 2025-26

Class:	BE	**Semester:**	VII
Course Code:	CSDOL7011	**Course Name:**	Natural Language Processing

Name of Student:	Konisha Jayesh Thakare
Roll No. :	71
Experiment No.:	6
Title of the Experiment:	Performing Chunking and Named Entity Recognition using NLTK
Date of Performance:	19.08.2025
Date of Submission:	26.08.2025

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

Checked by

Name of Faculty : Dr. Tatwadarshi P. Nagarhalli

Signature :

Date :

Aim: To perform chunking and Named Entity Recognition (NER) on a text corpus using NLTK, and to extract meaningful phrases and named entities from text.

Theory:

Chunking (or shallow parsing) is the process of grouping words in a sentence into meaningful phrases such as noun phrases (NP) or verb phrases (VP). Unlike full parsing, chunking does not generate a complete parse tree but identifies segments that form a syntactic unit. It is widely used in information extraction and text summarization. Named Entity Recognition (NER) identifies proper nouns and entities in text and classifies them into predefined categories such as Person, Organization, Location, Date, and Money. NER is crucial for applications like question answering, knowledge extraction, and document indexing. NLTK provides built-in tools for both chunking and NER using regular-expression-based chunkers and pre-trained named entity classifiers.

Procedure:

1. Prepare or select a sample text corpus.
2. Preprocess the text: tokenize sentences and words.
3. Apply POS tagging to the tokens.
4. Define a chunking grammar to identify noun or verb phrases.
5. Perform chunking using NLTK's RegexpParser.
6. Perform Named Entity Recognition using NLTK's ne_chunk.
7. Visualize and analyze the extracted phrases and named entities.

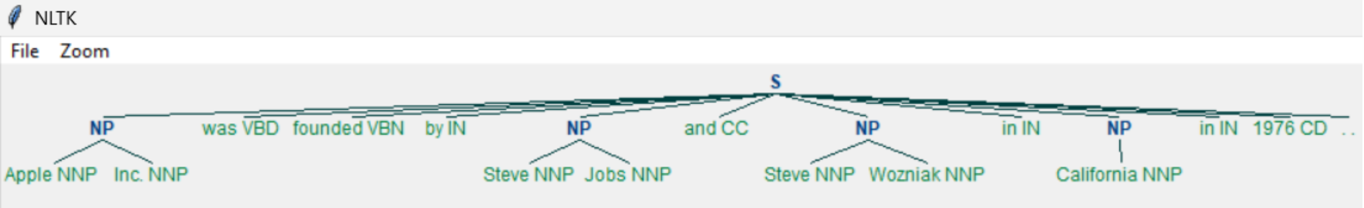
Algorithm:

1. Tokenize sentences and words.
2. Apply POS tagging.
3. Define chunking patterns (e.g., NP = optional determiner + adjectives + noun).
4. Use RegexpParser to extract chunks.
5. Use ne_chunk for NER.
6. Display and interpret the results.

Implementation:

```
import nltk
from nltk import word_tokenize, pos_tag, ne_chunk
from nltk.tree import Tree
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
sentence="Apple Inc. was founded by Steve Jobs and Steve Wozniak in California in 1976."
tokens=word_tokenize(sentence)
pos_tags=pos_tag(tokens)
print("POS Tags:\n",pos_tags)
grammar="NP: {<DT>?<JJ>*<NN.*>+}"
chunk_parser=nltk.RegexpParser(grammar)
chunk_tree=chunk_parser.parse(pos_tags)
print("\nChunked Tree:")
print(chunk_tree)
ner_tree=ne_chunk(pos_tags)
print("\nNamed Entities:")
print(ner_tree)
chunk_tree.draw()
ner_tree.draw()
```

Output:



Conclusion:

This experiment demonstrated chunking and Named Entity Recognition using NLTK. Chunking allowed the identification of meaningful phrases, such as noun phrases, while NER automatically detected entities like persons, organizations, and locations. These techniques are essential for extracting structured information from unstructured text, forming the foundation for applications such as information retrieval, question answering, and knowledge graph construction. The results show that combining POS tagging with chunking and NER provides a powerful method for syntactic and semantic analysis of text.