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.:	5
Title of the Experiment:	Performing Part-of-Speech Tagging and Syntactic Analysis using NLTK
Date of Performance:	12.08.2025
Date of Submission:	19.08.2025

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 Part-of-Speech (POS) tagging and syntactic analysis on a text corpus using the NLTK library, and to understand the grammatical structure of sentences.

Theory:

Part-of-Speech (POS) tagging is the process of labeling each word in a text with its grammatical category, such as noun, verb, adjective, or adverb. POS tagging is fundamental in NLP because it helps understand the syntactic structure of a sentence and is used in applications like information extraction, parsing, and question answering. Syntactic analysis (parsing) identifies the grammatical structure of a sentence, representing relationships between words, typically as a parse tree. There are two main approaches:

- Constituency parsing: Breaks sentences into sub-phrases (noun phrase, verb phrase, etc.).
- Dependency parsing: Represents relationships between words as directed links (subject → verb, verb → object).

The NLTK library provides built-in tools for tokenization, POS tagging, and syntactic analysis, making it suitable for learning and experimentation in NLP labs.

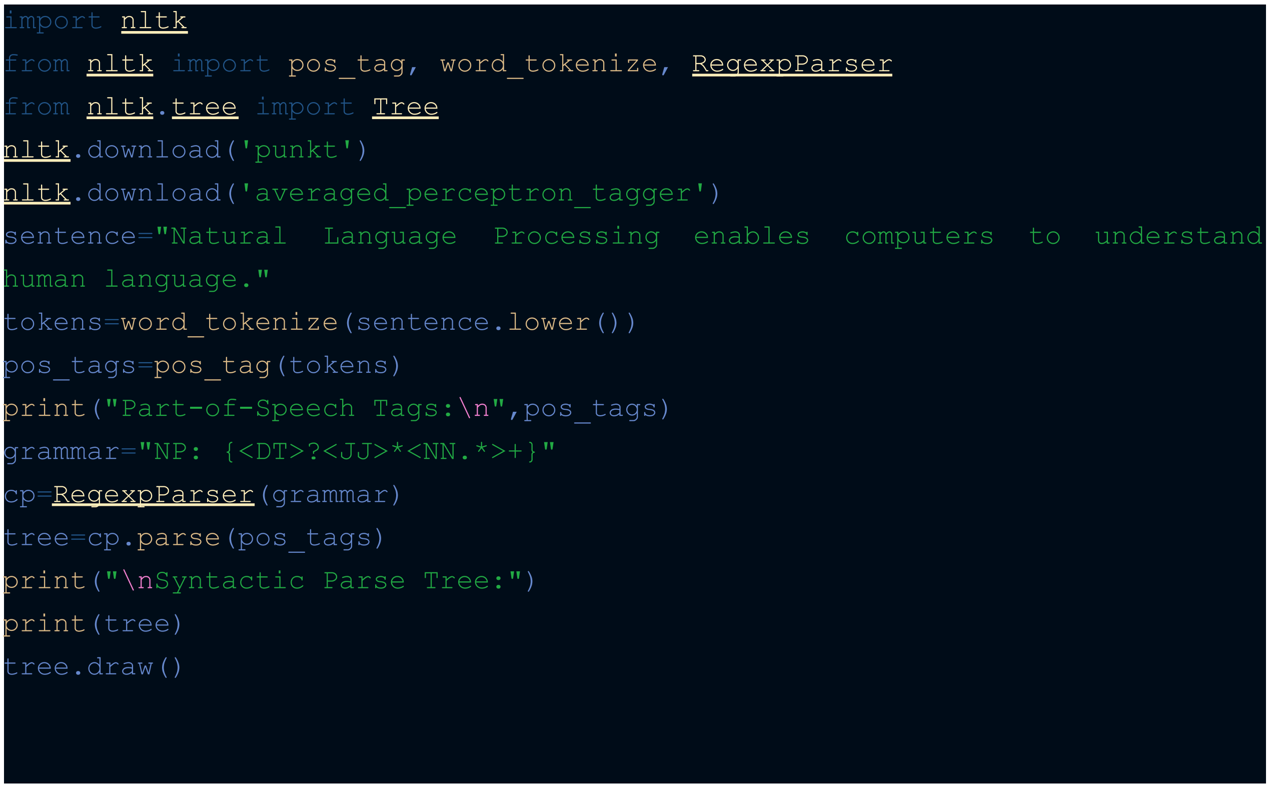
Procedure:

1. Prepare a sample text or corpus.
2. Preprocess the text: lowercase, tokenize sentences and words.
3. Apply POS tagging using NLTK.
4. Perform syntactic analysis to generate a parse tree.
5. Analyze the structure and relationships between words in sentences.

Algorithm:

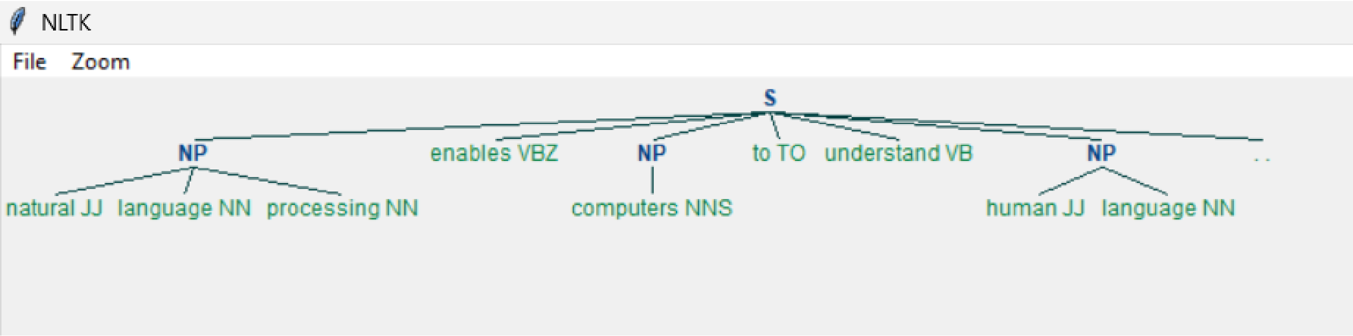
1. Tokenize sentences from the corpus.
2. Tokenize words within each sentence.
3. Apply NLTK's POS tagger to assign grammatical labels.
4. Use NLTK's parser or chunker for syntactic analysis.
5. Visualize the parse tree.
6. Interpret POS tags and tree structure for analysis.

Implementation:



```
import nltk
from nltk import pos_tag, word_tokenize, RegexpParser
from nltk.tree import Tree
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
sentence="Natural Language Processing enables computers to understand human language."
tokens=word_tokenize(sentence.lower())
pos_tags=pos_tag(tokens)
print("Part-of-Speech Tags:\n",pos_tags)
grammar="NP: {<DT>?<JJ>*<NN.*>+}"
cp=RegexpParser(grammar)
tree=cp.parse(pos_tags)
print("\nSyntactic Parse Tree:")
print(tree)
tree.draw()
```

Output:



Conclusion:

This experiment demonstrated the use of NLTK for Part-of-Speech tagging and syntactic analysis. POS tagging assigns grammatical labels to words, enabling machines to understand sentence structure, while syntactic parsing identifies hierarchical relationships between words. By generating parse trees, one can visualize noun phrases, verbs, and their dependencies, providing insights into the grammatical structure of sentences. Such analyses are foundational for many NLP applications, including information extraction, text summarization, and question-answering systems.