BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
Department of Computer Engineering

Name	**Raj Kalpesh Mathuria**
UID no.	**2023300139**
Experiment No.	**1**

Aim:	Perform basic corpus analysis using NLTK, remove stop words, visualize word frequency, compare NLTK vs spaCy, and analyze morphological changes using an Add-Delete table.
Problem statement :	Given a text paragraph, tokenize it, remove stop words using both library and manual lists, show stop words found, compute frequency distribution with visualization, and document a comparative study of NLTK and spaCy. Also, build an Add-Delete table for morphological analysis using user-supplied source and final forms.
Theory:	Corpus is a structured collection of text used for analysis. Stop words are high-frequency function words (e.g., "the", "is") that often carry little semantic weight in many NLP tasks. NLTK provides tokenization, stop word lists, and frequency analysis tools. spaCy provides fast, production-ready NLP pipelines (tokenization, POS, NER, parsing). Morphological analysis studies word formation; the Add-Delete table records the suffix/prefix changes between a root and its final form.
Algorithm:	1. Read a paragraph from user input; if empty, load data/sample.txt. 2. Tokenize the text using NLTK. 3. Load NLTK stop words and a manual stop word list. 4. Identify stop words present in the paragraph. 5. Display the paragraph after stop word removal using (a) NLTK list and (b) manual list. 6. Compute frequency distribution on cleaned tokens and visualize it. 7. Build a comparison table between NLTK and spaCy. 8. Accept source and final forms from the user. 9. Compute Add/Delete strings via longest common prefix. 10. Collect Number, Gender, Case, and Tense inputs and display the Add-Delete table.

Program:

```
import nltk
import pandas as pd
import matplotlib.pyplot as plt
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
import os

try:
nltk.data.find('tokenizers/punkt_tab')
except LookupError:
print("Downloading required NLTK data...")
nltk.download('punkt_tab', quiet=True)
nltk.download('punkt', quiet=True)

def run_aim_1():
print("\n" + "="*40)
print(" AIM 1: NLTK Basic Analysis & FreqDist")
print("="*40)

file_path = os.path.join("data", "sample.txt")
if os.path.exists(file_path):
with open(file_path, 'r', encoding='utf-8') as file:
text_content = file.read()
print(f"Loaded text from {file_path}")
else:
print(f"File {file_path} not found. Using default text.")
text_content = """
Natural language processing (NLP) refers to the branch of computer
science concerned with
giving computers the ability to understand text and spoken words in
much the same way human
beings can. NLP drives computer programs that translate text from one
language to another,
respond to spoken commands, and summarize large volumes of text
rapidly.
"""

tokens = word_tokenize(text_content)
print(f"\nTotal Tokens before removal: {len(tokens)}")
print(f"First 10 tokens: {tokens[:10]}")
```

```
print("\n" + "-"*30)
print(" STOP WORD REMOVAL (MANUAL)")
print("-" * 30)
user_input = input("Enter stop words to remove (separated by space, e.g., 'the is and of'): ")
stop_words_list = [w.strip().lower() for w in user_input.split()]
print(f"Stop words to remove: {stop_words_list}")

filtered_tokens = []
for token in tokens:
if token.lower() not in stop_words_list:
filtered_tokens.append(token)

print(f"\nTotal Tokens after removal: {len(filtered_tokens)}")
print(f"First 10 filtered tokens: {filtered_tokens[:10]}")

fdist = FreqDist(filtered_tokens)
print("\nTop 5 Most Common Words (After Cleanup):")
print(fdist.most_common(5))

print("\nDisplaying Frequency Plot... (Close the plot window to continue)")
plt.figure(figsize=(10, 7))
plt.title("Word Frequency Distribution (Stop Words Removed)")
fdist.plot(20, cumulative=False)
plt.show()

def run_aim_2():
print("\n" + "="*40)
print(" AIM 2: Morphological Analysis (Add-Delete Table)")
print("="*40)

user_root = input("Enter Source/Root Word (e.g., teach): ").strip()
user_final = input("Enter Final Form Word (e.g., teaches): ").strip()
results = []

if user_root and user_final:
delete_rule = "-"
add_rule = "-"
common_len = 0
min_len = min(len(user_root), len(user_final))
for i in range(min_len):
```

```
if user_root[i] == user_final[i]:
common_len += 1
else:
break
del_str = user_root[common_len:]
add_str = user_final[common_len:]

if del_str:
delete_rule = del_str
if add_str:
add_rule = add_str

results.append({
"Source (Root)": user_root,
"Final Form": user_final,
"Delete": delete_rule,
"Add": add_rule,
"Number": "User-Input",
"Gender": "-",
"Case": "-"
})

df = pd.DataFrame(results)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)
print("\nFinal Add-Delete Table:")
print(df)

if __name__ == "__main__":
run_aim_1()
run_aim_2()
```

Output:

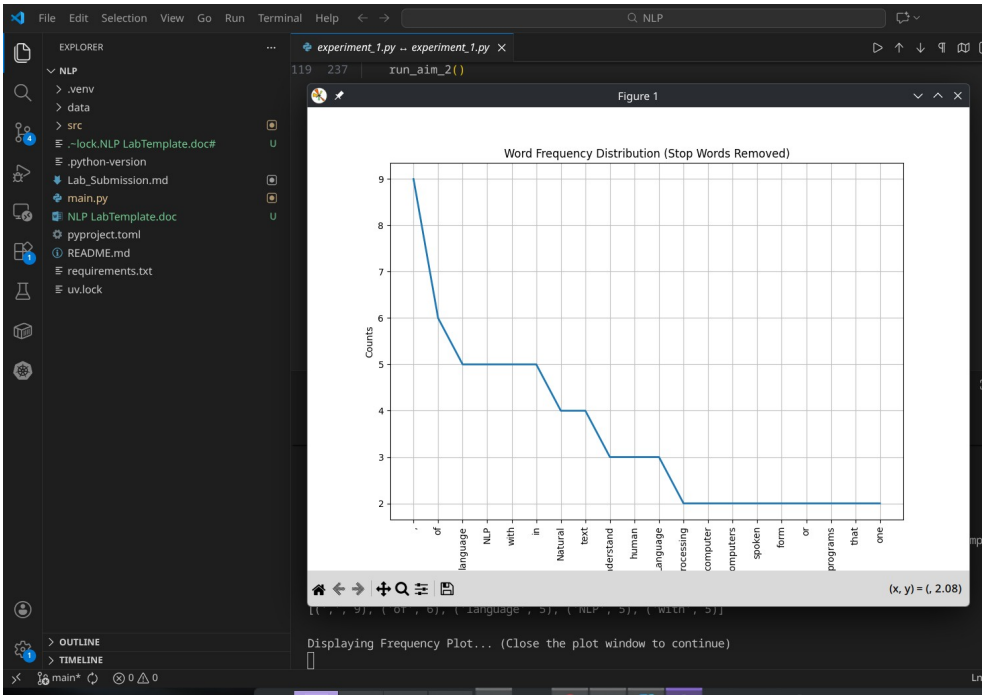
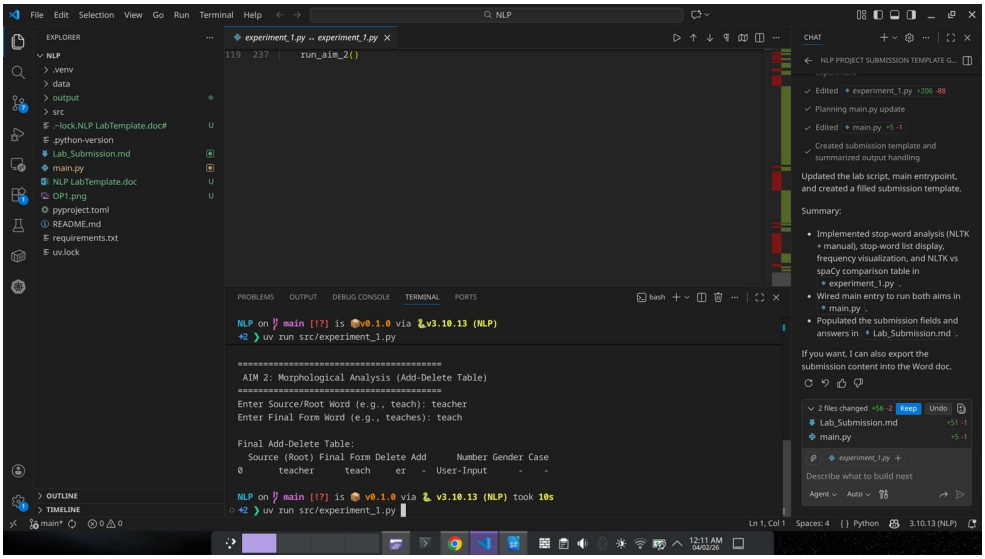


Figure 1
Word Frequency Distribution (Stop Words Removed)
Counts
Displaying Frequency Plot... (Close the plot window to continue)


AIM 2: Morphological Analysis (Add-Delete Table)
Enter Source/Root Word (e.g., teach): teacher
Enter Final Form Word (e.g., teaches): teach
Final Add-Delete Table:
Source (Root) Final Form Delete Add Number Gender Case
0 teacher teach er - User-Input - -

BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
Department of Computer Engineering

Questions to be answered	1) What is corpus, stop words? A corpus is a large, structured collection of texts used to train or evaluate NLP systems. Stop words are very common function words (e.g., "the", "is", "and") that are often removed to reduce noise in tasks like search or topic modeling. 2) What is normalization in NLP? How does it work? Why is it important? Normalization transforms text into a consistent form (e.g., lowercasing, removing punctuation, expanding contractions, stemming/lemmatization). It reduces variation so that semantically similar tokens are treated uniformly, improving matching, indexing, and model performance. 3) Describe different ambiguities in NLP with example. - Lexical ambiguity: A word has multiple meanings (e.g., "bank" = river bank or financial bank). - Syntactic ambiguity: Multiple parse structures (e.g., "I saw the man with a telescope."). - Semantic ambiguity: Multiple interpretations after parsing (e.g., "Visiting relatives can be boring."). - Pragmatic ambiguity: Meaning depends on context or speaker intent (e.g., "Can you open the window?" as a request). 4) What is WordNet and its relevance? WordNet is a lexical database grouping words into synonym sets (synsets) with semantic relations (hypernyms, hyponyms, etc.). It is useful for semantic similarity, word sense disambiguation, and enriching NLP features.
Conclusion:	The experiment demonstrates basic corpus analysis with NLTK, stop word removal, word frequency visualization, and a comparative study of NLTK vs spaCy. Morphological analysis using the Add-Delete table captures word formation patterns. These steps provide foundational skills for NLP preprocessing and analysis.