Text and Speech Analysis-**CCS369**

**Observation (EXP-2)**

**1)Searching text:**

!pip install nltk  
import nltk  
from nltk.tokenize import word\_tokenize  
  
nltk.download('punkt')  
  
text = "Natural Language processing comes under deep learning too and it's important language."  
tokens = word\_tokenize(text)  
search\_word = "Language"  
matches = [word for word in tokens if word.lower() == search\_word.lower()]  
print(f"Occurrence of '{search\_word}':", len(matches))  
print(tokens)

**Output:**

Occurrence of 'Language': 2

['Natural', 'Language', 'processing', 'comes', 'under', 'deep', 'learning', 'too', 'and', 'it', "'s", 'important', 'language', '.']

**2)Counting Vocabulary:**

from nltk.probability import FreqDist  
from nltk.tokenize import word\_tokenize  
import string  
  
text = "Text and Speech analysis is an awesome subject. This Subject is powerful."  
tokens = word\_tokenize(text)  
filtered\_tokens = [word for word in tokens if word not in string.punctuation]  
  
fdist = FreqDist(filtered\_tokens)  
  
print("Vocabulary Count:", fdist.most\_common())

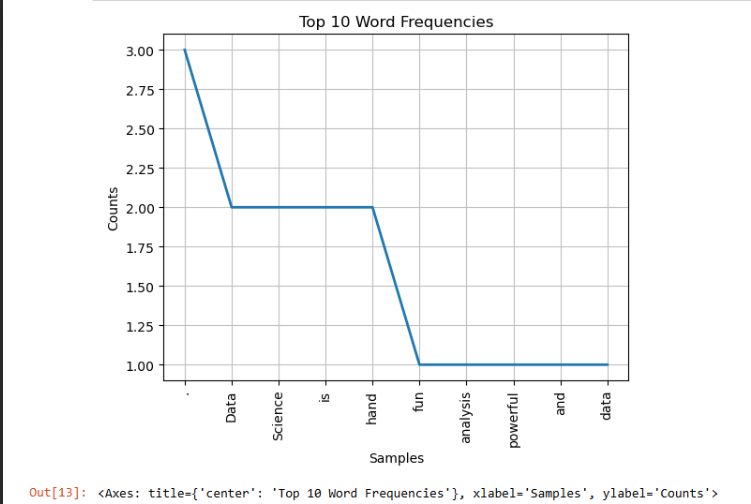
**Output:**

Vocabulary Count: [('is', 2), ('Text', 1), ('and', 1), ('Speech', 1), ('analysis', 1), ('an', 1), ('awesome', 1), ('subject', 1), ('This', 1), ('Subject', 1), ('powerful', 1)]

**3)Frequency Distribution(withplot):**

import matplotlib.pyplot as plt  
from nltk.tokenize import word\_tokenize  
from nltk.probability import FreqDist  
  
text = "Data Science is fun. Data analysis is powerful. Science and data go hand in hand."  
tokens = word\_tokenize(text)  
fdist = FreqDist(tokens)  
  
fdist.plot(10, title="Top 10 Word Frequencies")

**Output:**



**4) Collocations:**

import nltk  
from nltk.collocations import BigramCollocationFinder  
from nltk.metrics import BigramAssocMeasures  
from nltk.tokenize import word\_tokenize  
  
text = "Data Science is fun. Data analysis is powerful. Science and data go hand in hand."  
tokens = word\_tokenize(text)  
  
  
finder = BigramCollocationFinder.from\_words(tokens)  
collocations = finder.nbest(BigramAssocMeasures.likelihood\_ratio, 5)  
print("Top collocations:", collocations)  
  
**Output:**

Top collocations: [('and', 'data'), ('data', 'go'), ('Data', 'analysis'), ('Science', 'and'), ('analysis', 'is')]

**5)Bigrams:**

import nltk  
from nltk.tokenize import word\_tokenize  
text = "Data Science is fun. Data analysis is powerful. Science and data go hand in hand."  
bigrams=list(nltk.bigrams (tokens))  
print("Bigrams:",bigrams)  
  
**Output:**

Bigrams: [('Data', 'Science'), ('Science', 'is'), ('is', 'fun'), ('fun', '.'), ('.', 'Data'), ('Data', 'analysis'), ('analysis', 'is'), ('is', 'powerful'), ('powerful', '.'), ('.', 'Science'), ('Science', 'and'), ('and', 'data'), ('data', 'go'), ('go', 'hand'), ('hand', 'in'), ('in', 'hand'), ('hand', '.')]

|  |  |
| --- | --- |
|  |  |