Name: Vinayak V Thayil

Roll No:AM.EN.U4CSE21161

```
#1 Write a paragraph about any chatbot and save as chatbot .text file.
chatbot_paragraph = "A chatbot is a computer program designed to simulate human conversation.\
It can interact with users through text or voace, providing information, answering questions, and even assisting with tasks.\
Chatbots are used in various applications, from customer support to virtual assistants, and they rely on natural language processing to unders
They have become increasingly popular in recent years, enhancing user experiences and automating routine interactions."
with open("chatbot.txt", "w") as file:
    file.write(chatbot_paragraph)
```

```
#2 Read and display this file with a Python program.
with open("chatbot.txt", "r") as file:
    chatbot_text = file.read()
print(chatbot_text)
```

```
A chatbot is a computer program designed to simulate human conversation.It can interact with users through text or voace, providing infc
```

```
#3 Extract sentences and words from this file.
import nltk
file_path = "chatbot.txt"
with open(file_path, 'r', encoding='utf-8') as file:
    text = file.read()
sentences = sent_tokenize(text)
words = word_tokenize(text)
print(sentences)
print(words)
```

```
['A chatbot is a computer program designed to simulate human conversation.It can interact with users through text or voace, providing ir
['A', 'chatbot', 'is', 'a', 'computer', 'program', 'designed', 'to', 'simulate', 'human', 'conversation.It', 'can', 'interact', 'with',
```

```
#4 Count the number of stop words and nonstop-words.
import nltk
stop_words = set(stopwords.words('english'))
non_stop_words = [word for word in words if word.lower() not in stop_words]
num_stop_words = len(words) - len(non_stop_words)
print("Number of Stopwords: ", num_stop_words)
print("Number of Non-Stopwords: ", len(non_stop_words))
```

```
Number of Stopwords:  23
Number of Non-Stopwords:  51
```

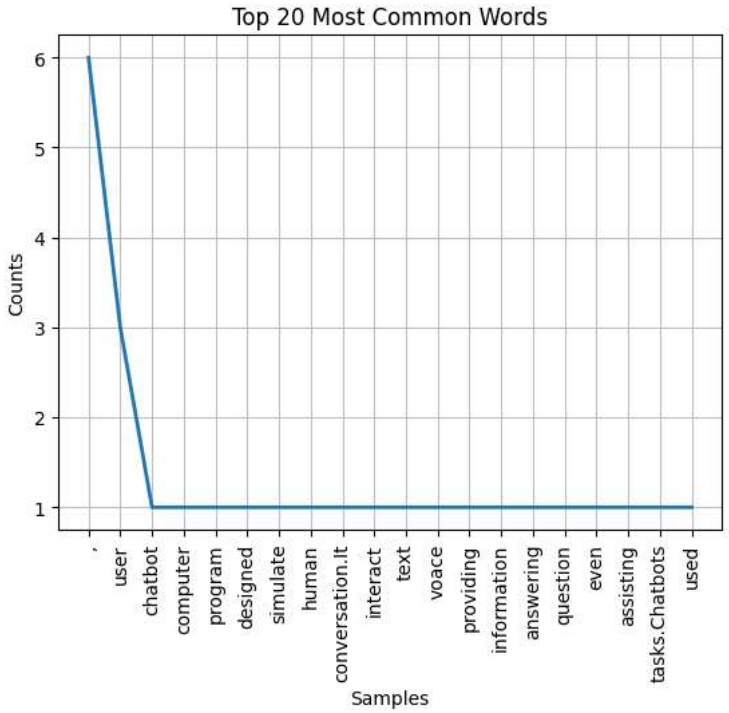
```
#5 If there are any stopwords, remove them.
filtered_text = ' '.join(non_stop_words)
print(filtered_text)
```

```
chatbot computer program designed simulate human conversation.It interact users text voace , providing information , answering questions
```

```
#6 Identify the stems and lemma for these words
import nltk
nltk.download('wordnet')
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()
stems = [stemmer.stem(word) for word in non_stop_words]
lemma = [lemmatizer.lemmatize(word) for word in non_stop_words]
print(stems)
print(lemma)
```

```
['chatbot', 'comput', 'program', 'design', 'simul', 'human', 'conversation.it', 'interact', 'user', 'text', 'voac', ',', 'provid', 'infc
['chatbot', 'computer', 'program', 'designed', 'simulate', 'human', 'conversation.It', 'interact', 'user', 'text', 'voace', ',', 'provic
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
#7 Find the frequency of words and plot the frequency distribution.
import matplotlib.pyplot as plt
from nltk import FreqDist
freq_dist = FreqDist(lemma)
freq_dist.plot(20, title="Top 20 Most Common Words")
print(freq_dist)
```




```
<FreqDist with 44 samples and 51 outcomes>
```

```
#8 Check for any words that are not lemmatised
non_lemmatized_words = [word for word, lemma in zip(non_stop_words, lemma) if word != lemma]
print(non_lemmatized_words)
```

```
['users', 'questions', 'applications', 'assistants', 'years', 'experiences', 'interactions']
```

```
#9 If there are any spelling mistakes, correct the identified ones
!pip install python-Levenshtein
!pip install pyspellchecker
import Levenshtein as distance
from spellchecker import SpellChecker
spell = SpellChecker()
def closest_match(word, word_list):
    return min(word_list, key=lambda x: distance.distance(word, x))
misspelled = [word for word in words if word.lower() not in spell.word_frequency.dictionary]
corrected_words = [closest_match(word, spell.word_frequency.dictionary) if word in misspelled else word for word in words]
print("Original Words:")
print(words)
print("\nCorrected Words:")
print(corrected_words)
```

```
Requirement already satisfied: python-Levenshtein in /usr/local/lib/python3.10/dist-packages (0.23.0)
Requirement already satisfied: Levenshtein==0.23.0 in /usr/local/lib/python3.10/dist-packages (from python-Levenshtein) (0.23.0)
Requirement already satisfied: rapidfuzz<4.0.0,>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from Levenshtein==0.23.0->python-Leve
Requirement already satisfied: pyspellchecker in /usr/local/lib/python3.10/dist-packages (0.7.2)
Original Words:
['A', 'chatbot', 'is', 'a', 'computer', 'program', 'designed', 'to', 'simulate', 'human', 'conversation.It', 'can', 'interact', 'with',

Corrected Words:
['A', 'cabot', 'is', 'a', 'computer', 'program', 'designed', 'to', 'simulate', 'human', 'conversation', 'can', 'interact', 'with', 'user
```

```
#10 Translate any ten of these words into your native language.
!pip install mtranslate
from mtranslate import translate
words_to_translate = non_lemmatized_words[:10]
for word in words_to_translate:
    translation = translate(word, 'ml')
    print(f"{word} - {translation}")
```

```
Requirement already satisfied: mtranslate in /usr/local/lib/python3.10/dist-packages (1.8)
users - ഉപയോക്താക്കൾ
questions - ചോദ്യങ്ങൾ
applications - അപേക്ഷകൾ
assistants - സഹായികൾ
years - വർഷങ്ങൾ
experiences - അനുഭവങ്ങൾ
interactions - ഇടപെടലുകൾ
```