

**DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING**

|  |  |  |  |
| --- | --- | --- | --- |
| *Assignment* | **I** | *Academic Year/Semester* | **2024-25 /VI** |
| *Subject* | **CST 362 PROGRAMMING IN PYTHON** | *Branch* | **CSE A** |
| *Date of Announcement* | 04-03-2024 | *Date of Submission* | **10-03-2024** |
|  |  | *Max. Marks* | 15 |

**A case study on text analysis**

Imagine you are building a text analysis tool for a content creator who wants to optimize their writing. Your tool should perform the following tasks:

Calculate the average word length in the given text.

Identify and display the top three most frequent words in the text along with their counts.

Implement a feature to suggest replacing overused words (words repeated more than three times) with synonyms. Utilize an external API for synonym retrieval.

Your program should provide an interactive interface for the user to input the text and receive the analysis results.

**Requirements:**

Implement a function to calculate the average word length.

Implement a function to identify and display the top three most frequent words.

Utilize an external API for synonym retrieval and suggest replacements for overused words.

Gopikrishna V

S6 CSA

52

Program

import requests

from collections import Counter

class req\_functions():

    def averageWordLength(text):

        words=text.split()

        Tlen=sum(len(word) for word in words)

        if len(words)==0:

            return 0

        else:

            return Tlen/len(words)

    def modeT3(text):

        words=text.split()

        count=Counter(words)

        return count.most\_common(3)

    def synonymAPI(word):

        url=f"https://api.datamuse.com/words?rel\_syn={word}"

        response=requests.get(url)

        if response.status\_code==200:

            data=response.json()

            return [item['word'] for item in data]

        else:

            print("Failed Retrieval\n")

            return []

    def wordReplaceSuggest(textR,overusedW):

        suggestions={}

        for wordx in overusedW:

            synonyms=req\_functions.synonymAPI(wordx)

            if synonyms:

                suggestions[wordx]=synonyms[0]

            else:

                suggestions[wordx]="NO REPLACEMENTS"

        return suggestions

print('\nAVERAGE WORD LENGTH FINDER\n')

text=input("String >> ")

avgLen=req\_functions.averageWordLength(text)

print("Average Word Length = ", avgLen)

print('\nTOP THREE FREQUENT WORDS FINDER\n')

text=input("String >> ")

for word,count in req\_functions.modeT3(text):

    print(f"{word} :: {count}")

print('\nREPLACEMENT SUGGESTOR\n')

textS=input("String >> ")

wordS=input("Overused Words List >> ")

overusedList=wordS.split()

replaceW=req\_functions.wordReplaceSuggest(textS,overusedList)

for wordR,replaceW in replaceW.items():

    print(f"{wordR} >>> {replaceW}")

Output

