CASE STUDY: PLAGIARISM DETECTION SYSTEM

By Porwal Rushabh Rajesh

2022SP93034

2022-2YA

Background:

BITS Pilani is committed to maintaining academic integrity and ensuring that students submit original work. To solve the problem of plagiarism, the university decided to create a plagiarism detection system using the LCS method.

Scenario:

The university collects a set of essays submitted by students for a specific assignment. The goal is to compare these essays and identify case of copying where there are significant similarities between the textual content of two or more essays.

Justification:

The LCS algorithm is designed to find the Longest Common Subsequence between two sequences, making it suitable for comparing the textual content of essays. Plagiarised text contains copies or paraphrases, and the LCS algorithm can effectively identify similarities based on the order of words or phrases in the work.

Solution:

Essay Collection and Preprocessing:

The university collects essays submitted by students and prepares them for analysis. Each sentence is displayed as string. Essays are then processed to standardize the text by removing punctuation, spaces, and lowercase letters. This ensures that the comparison is only based on textual content and reduces false matches.

Implementing the LCS Algorithm:

A development team at the university implemented the LCS algorithm into a Plagiarism Detection System. They created a function that takes two essays as input and calculates the length of the LCS between them. This function uses Dynamic Programming to find LCS efficiently.

Essay Comparison and Threshold:

The Plagiarism Detection System compares each text with every other text in the collection using the implemented LCS algorithm. The length of LCS is calculated for each pair of Essays. The University sets a Plagiarism Threshold that represents the minimum length of the LCS for two copies to be considered a potential Plagiarism. For example, a threshold of 50% means that if the LCS length between two essays exceeds 50% of the longer essay's length, it is flagged as potentially plagiarized.

Identifying Potential Plagiarism Cases:

Based on the calculated LCS length, the Plagiarism Detection System identifies possible Plagiarism cases. Essay pairs exceeding a specified threshold are marked as probable Plagiarism cases. The system generates a report that lists these pairs, showing the number of entries and the percentage of overlap.

Manual Investigation and Actions:

The University manually investigates plagiarism cases detected by the system. A faculty member carefully reviews the assigned essay to check for similarities and ensure that it is really an Plagiarism Case. If necessary, they can use additional tools or techniques to gather more evidence. Appropriate actions, including warnings, disciplinary action or disciplinary action, are taken based on the findings of the investigation and university policy.

Results:

By implementing the Plagiarism Detection System based on the LCS method, BITS Pilani successfully detected plagiarism among submissions by students. This helps maintain academic integrity, promotes originality, and ensures fair evaluation of student effort.

The plagiarism detection system serves to raise awareness of the consequences of plagiarism and encourage students to submit original work. Create a culture of academic integrity where students understand the importance of generating their own ideas and citing external sources properly.

Overall, the LCS-based Plagiarism Detection System contributes to maintaining the reputation and credibility of BITS Pilani as an institution committed to academic excellence and integrity.

Code:

```
import os

def preprocessText(text):
    text = text.lower()
    text = ''.join(c for c in text if c.isalnum())
    return text

def calculateLCSlength( text1, text2 ):
    m = len(text1)
    n = len(text2)

    lcsMatrix = [ [0]*(n+1) for _ in range(m+1) ]

    for i in range(1, m + 1):
        for j in range(1, n + 1):
            if text1[i - 1] == text2[j - 1]:
                lcsMatrix[i][j] = lcsMatrix[i - 1][j - 1] + 1
            else:
                lcsMatrix[i][j] = max(lcsMatrix[i - 1][j], lcsMatrix[i][j -
1])

    return lcsMatrix[m][n]

def plagarismDetector(essays, threshold):
    numEssays = len(essays)
    plagarismCases = []
```

```
    for i in range(numEssays):
        for j in range(i + 1, numEssays):
            essay1 = preprocessText(essays[i])
            essay2 = preprocessText(essays[j])
            lcs_length = calculateLCSlength(essay1, essay2)
            thresholdLength = max(len(essay1), len(essay2))*(threshold/100)
            print("#LCS Length,Threshold Length, Essay lengths : ",lcs_length,
thresholdLength, [len(essay1), len(essay2)])
            if lcs_length >= thresholdLength:
                plagarismPercent = lcs_length/max(len(essay1),
len(essay2))*100
                # print("## Essays : ",[essay1,essay2])
                print("##Plagarism Case : ",i , j, plagarismPercent)
                plagarismCases.append((i, j, plagarismPercent))

    return plagarismCases

def getEssays(folderName):
    # print(folderName)
    essays = []
    text_files = [f for f in os.listdir(folderName) if f.endswith('.txt')]
    print("## Files : ",text_files)

    for file in text_files:
        essays.append(open(os.path.join(folderName, file),'r').read())

    return essays,text_files

def getReport(plagarismCases, essayNames):
    report = open('report.txt', 'w+')

    report.writelines("##### Plagarism Report #####")
    report.writelines(f"\nTotal Essays : {len(essayNames)}")
    report.writelines(f"\nTotal Plagarism Detected : {len(plagarismCases)}")
    report.writelines("\n---")

    print("##### PLAGIARISM REPORT #####")
    print("Total Essays : ",len(essayNames))
    print("Total Plagarism Detected : ",len(plagarismCases))
    print("---")
    for case in plagarismCases:
      essay1_index, essay2_index, plagarismPercent = case
      report.writelines(f"\nPotential plagiarism detected between
'{essayNames[essay1_index]}' and '{essayNames[essay2_index]}'.")
      report.writelines(f"\nPlagarism Rate(%): {round(plagarismPercent,2)}%")
      report.writelines("\n---")
```

```
      print(f"Potential plagiarism detected between
'{essayNames[essay1_index]}' and '{essayNames[essay2_index]}'.")
      print(f"Plagarism Rate(%): {round(plagarismPercent,2)}%")
      print("---")

folderName = input('Folder Location : ("./Submissions") ') or "./Submissions"
threshold = int(input('Threshold (%) : ("50")') or '50')
essays, essayNames = getEssays(folderName)
# print("TotalEssays : ",len(essays))
# print("Essays : ", essays)
plagarismCases = plagarismDetector(essays, threshold)
print("PlagarismCases : ", plagarismCases)
getReport(plagarismCases,essayNames)
```

Output:

```
Folder Location : ("./Submissions")
Threshold (%) : ("50")
## Files :  ['essay1.txt', 'essay2.txt', 'essay3.txt', 'essay4.txt', 'essay5.txt']
#LCS Length,Threshold Length, Essay lengths :  15 19.0 [19, 38]
#LCS Length,Threshold Length, Essay lengths :  12 18.5 [19, 37]
#LCS Length,Threshold Length, Essay lengths :  13 23.0 [19, 46]
#LCS Length,Threshold Length, Essay lengths :  12 26.5 [19, 53]
#LCS Length,Threshold Length, Essay lengths :  20 19.0 [38, 37]
##Plagarism Case :  1 2 52.63157894736842
#LCS Length,Threshold Length, Essay lengths :  28 23.0 [38, 46]
##Plagarism Case :  1 3 60.86956521739131
#LCS Length,Threshold Length, Essay lengths :  19 26.5 [38, 53]
#LCS Length,Threshold Length, Essay lengths :  20 23.0 [37, 46]
#LCS Length,Threshold Length, Essay lengths :  20 26.5 [37, 53]
#LCS Length,Threshold Length, Essay lengths :  21 26.5 [46, 53]
PlagarismCases :  [(1, 2, 52.63157894736842), (1, 3, 60.86956521739131)]
##### PLAGIARISM REPORT #####
Total Essays :  5
Total Plagarism Detected :  2
---
Potential plagiarism detected between 'essay2.txt' and 'essay3.txt'.
Plagarism Rate(%): 52.63%
---
Potential plagiarism detected between 'essay2.txt' and 'essay4.txt'.
Plagarism Rate(%): 60.87%
---
```