Report CS

For our CS project, we chose the Simple Plagiarism Detector.

This program compares each sentence in a text file (test.txt) with every other sentence in the same file using the Brute Force and Rabin-Karp algorithms. There is only one class used which is PDetector.

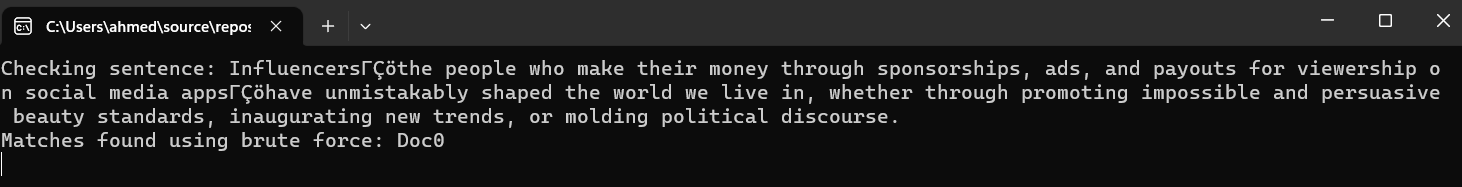
There are 6 member functions in the PDetector class:

* Constructor is void [ PDetector() ]
* Hamming\_distance(string s1, string s2) returns the Hamming distance between two strings when s1 and s2 are inputted. The distance between a pattern ( i.e sentence) and all of the substrings in the same text file is calculated using this function by the brute\_force() function.
* String pattern, text victor, int max\_distance, brute\_force (i.e sentence), a vector of strings (i.e text), and maximum distance are all inputs for this function. A vector of numbers representing the indices of the sentecnes that shows the maximum distance of the pattern is returned. The function calculates the Hamming Distance between each substring in the text vector. The index [i] of the matching sentence is added to the matches vector if the distance is less than or equal to the maximum distance.

A picture containing text, screenshot, software, font

Description automatically generated

* Rabin\_karp (string pattern, string text, vector): is a function that accepts a pattern and a vector of strings as inputs. The Rabin-Karp algorithm produces a vector of numbers that shows the indices of the phrases that matches the pattern. Hashing is used by the Rabin-Karp algorithm to compare the pattern in the text victor to all substrings that are the same length as the pattern. The function checks the two strings character by character to validate whether there is a match if the hash value of the pattern matches thehash value of substring.
* The read\_file() method takes a filename as input (in this case the filename is test.text) and returns the file’s content as a string. The classes ifstream and istreambuf\_iterator are used to read the file and check for plagiarism.
* The split\_sentences(string text) takes a string as input and outputs a vector of strings representing a sentence in the text file. When it detects punctuation marks, it divides the text into sentences.
* The run() function of the PDetector class is a void function and is called by the main function. Run() executes the brute\_force() and rabin\_karp() methods for each sentence after reading the content of the test file into a string and splittin it into sentences. The output of this displays the actual sentences that has been potentially been plagiarized.



However, we encountered an error that we did not know how to solve. The system brings a message that says Debug Assertion Failed.

A screenshot of a computer program

Description automatically generated with medium confidence

Pressing Abort/Retry/Ignore just exists the program. 