



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

PLAGIRISM CHECKER

A Report for the Evaluation 3 of Project 2

Submitted by :-

KRISHNA KUMAR

(1613101340 / 16SCSE101603)

in partial fulfillment for the award of the degree
of

Bachelor of Technology

IN

Computer Science and Engineering

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Dr.P.sasikumar , M.Tech., Ph.D.,

Professor

APRIL / MAY- 2020

Table of Content

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1.	Abstract	1
2.	Introduction	2
3.	Existing system	5
4.	Proposed system	8
5.	Implementation or architecture diagrams	12
6.	Output / Result / Screenshot	14
7.	Conclusion/Future Enhancement	16

Abstract

The number of students following programming courses is steadily increasing at the same time as access to computers and networks is readily available. There is a significant minority of students who for a variety of reasons take advantage of the available technology and illicitly copy other students programming assignments and attempt to disguise their deception. Software that can help tutors to detect plagiarism is therefore of immense assistance in detecting and so helping to prevent such abuse. We design new and ancient algorithm for a basis to such software. Our algorithm is simple to implement, and provides very ancient means to detect plagiarized programs.

Introduction

Description

Assessment of programming courses typically involves students writing programs, either individually or in teams, which are then marked against criteria such as correctness and style. Unfortunately, it is very easy for students to exchange copies of code they have written. A student who has produced working code may be tempted to allow a colleague to copy and edit their program. This is discouraged, and is likely to be regarded as a serious disciplinary open. However, it is easy for a lecturer to fail to detect plagiarism, especially when class sizes are measured in hundreds of students. We have developed algorithm for detecting instances of possible plagiarism.

Purpose

One of the reasons to use a plagiarism checker is that this software highlights the content that is exact. In other words, you can see for yourself what sentences or words are verbatim what the original author wrote.

Plagiarism detection software also gives percentages of similarity. Many universities use plagiarism software like Turnitin to check papers for plagiarism. When students and instructors use this software to check papers, a similarity percentage is given. A university will have a standard percentage rate that is acceptable to them. Students must remain at that percentage rate or lower for their papers to be satisfactory in regard to similarity.

Plagiarism checker software offers you proof that you have not plagiarized. Printing out or saving an electronic copy of your plagiarism checker report can be proof to your instructor or university that your content is original. Some instructors will ask for a copy of this report, others will not. Keeping a copy for your records in either case can offer you protection

Motivation and Scope

Plagiarism has become very common in educational institutions. Students copy without any hesitation other students' assignments, both text and source code, to complete their work in time or to complete their work in a better way. Many students seldom care to put their time and effort into doing the assignments on their own when it is far simpler and effortless to copy from someone else. However, it is necessary to differentiate the original work from plagiarized work.

There is an alarming rise in plagiarism due to the widespread use of internet. Internet is an enormously huge repository of information which can be accessed easily from almost anywhere. This has made it very difficult to control plagiarism. Since the

task of manually detecting plagiarism in a large document database is very tedious and time-consuming, efforts are continuously being made to automate the process.

There exist many different plagiarism detection techniques and numerous tools based on these techniques. There are two main categories of techniques for source code plagiarism detection: attribute-counting-based and structure-based comparison. Attribute-counting-based techniques consider the number of occurrences of different attributes in a file following certain criteria and different similarity measures are used to obtain the similarity between files. Structure-based techniques derive information on program structure and obtain similarity scores based on this information. Section 1.4 gives a brief overview of the various plagiarism detection techniques.

Attribute-counting algorithms are simple to implement and execute faster. Structure-based methods, on the other hand, are more reliable since they gather details of program structure for comparison of programs. However, structure-based methods are computationally expensive. Hence, the aim of this research is to develop a new strategy which combines the advantages of both the categories.

Functional requirements:-

Enable teachers to detect plagiarism and cheating in student submitted assignments. The system reads the submitted assignments and enters them to the algorithm to find the degree of similarity between them. Viewing visually aided cheating (similarity) reports. Teachers can display cheating (plagiarism) report, which contains all submitted assignments and the percentage of similarity of each assignment with others. System is capable of displaying file content comparisons that have similarities. The system can automatically send alerts to students detected cheating or plagiarism action, in the form of SMS alerts.

Non-Functional requirements:-

Compatibility. System should be compatible and can be integrated with Moodle because it will be added as new feature to Moodle. Easy to use. Teachers will interact with the system to generate plagiarism report through a userfriendly graphical user interface. Furthermore, the generated reports will contain both textual and visual (bars, charts, etc.) representation for the results.

Development requirements:-

- Hardware resources:- Personal Computer (PC), for server application. - LAN or WAN, used to connect a computer server with client computer. - Modem for SMS Gateway connection.
- Software resources:- Plagiarism detection system to integrate into the virtual classroom platform. Moodle is a Web-based applications using PHP programming language, and database applications using MySQL,so it is a system built using the PHP programming and MySQL database. As for the SMS Gateway can be integrated into Moodle, used Ozeki NG.

Existing system :-

Existing system use RAVIN KARP algorithm for pattern searching that is slow and very time consuming compare to the algorithm we are using in our project that is KMP.

Knuth–Morris–Pratt string-searching algorithm (or KMP algorithm) searches for occurrences of a "word" W within a main "text **string**" S by employing the observation that when a mismatch occurs, the word itself embodies sufficient information to determine where the next match could begin, thus bypassing re-examination of previously matched characters. and sentiment analysis use cases. Naive Bayes Algorithm can be built using Gaussian, Multinomial and Bernoulli distribution. This algorithm is scalable and easy to implement for the large data set.

Naive:-

Naive Bayes Algorithm is one of the popular classification machine learning algorithms that helps to classify the data based upon the conditional probability values computation. It implements the Bayes theorem for the computation and used class levels represented as feature values or vectors of predictors for classification problems. This algorithm is a good fit for real-time prediction, multi-class prediction, recommendation system, text classification, and sentiment analysis use cases. Naive Bayes Algorithm can be built using Gaussian, Multinomial and Bernoulli distribution. This algorithm is scalable and easy to implement for the large data set.

```
public class Naive {
    public int timeCount=0;
    public int search(String pat, String txt) {
        int patternLen = pat.length();
        int txtLen = txt.length();
        int maxcount=0;
        for (int i = 0; i <= txtLen - patternLen; i++)
        {
            int j;
            /* For current index i, check for pattern match */
            for (j = 0; j < patternLen; j++) {
                timeCount++;
                if (txt.charAt(i + j) != pat.charAt(j))
                {
                    //System.out.println("j is"+ j);
                    maxcount= Math.max(maxcount,j);
                    break;
                }
            }
            if (j == patternLen) // if pat[0...M-1] = txt[i, i+1, ...i+M-1]
            {
                //System.out.println("Pattern found at index \n" + (i+1));
                i = i + patternLen;
                maxcount=patternLen;
            }
        }
    }
}
```

RABIN KARP:-

```
public class RabinKarp {
    public String pat;
    public long patHash;
    public int patlen;
    public long randomprime;

    public int ibase;
    public long RM;
    public int timeCount=0;

    private long hash(String key, int M) {
        long h = 0;
        for (int j = 0; j < M; j++) {
            h = (ibase * h + key.charAt(j)) % randomprime;
            timeCount++;
        }
        return h;
    }

    private boolean check(String txt,String pattern, int i) {
        String p = pattern;
        for (int j = 0; j < patlen; j++)
            { timeCount++;
              if (p.charAt(j) != txt.charAt(i + j))
                  return false;
            }
        return true;
    }
}
```


Proposed model:-

LCS (LONGEST COMMON SUBSEQUENCE)

LCS Problem Statement: Given two sequences, find the length of longest subsequence present in both of them. A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. For example, “abc”, “abg”, “bdf”, “aeg”, “acefg”, .. etc are subsequences of “abcdefg”.

In order to find out the complexity of brute force approach, we need to first know the number of possible different subsequences of a string with length n , i.e., find the number of subsequences with lengths ranging from 1, 2, .. $n-1$. Recall from theory of permutation and combination that number of combinations with 1 element are ${}^n C_1$. Number of combinations with 2 elements are ${}^n C_2$ and so forth and so on. We know that ${}^n C_0 + {}^n C_1 + {}^n C_2 + \dots + {}^n C_n = 2^n$. So a string of length n has $2^n - 1$ different possible subsequences since we do not consider the subsequence with length 0. This implies that the time complexity of the brute force approach will be $O(n * 2^n)$. Note that it takes $O(n)$ time to check if a subsequence is common to both the strings. This time complexity can be improved using dynamic programming.

```
public class LCSS {
    private String strOne;
    private String strTwo;
    private int len1;
    private int len2;
    private int lcssLen;
    private int[][] lcsTable; // the LCSS table to be populated while comparing strings using dynamic programming
    private char[] comStr; // the variable to store the longest common subsequence between strOne and strTwo
    public int timeCount=0;
    /**
     * Constructor for the LCSS initialized two strings and the lcsTable
     *
     * @param st1
     *         the first string
     * @param st2
     *         the second string
     */
    public LCSS(String st1, String st2){
        strOne = st1;
        strTwo = st2;
        len1 = strOne.length();
        len2 = strTwo.length();
        lcsTable = new int[len1 + 1][len2 + 1];

        for(int i = 0; i < len1; i++) // lcsTable gets initialized to 0 for all entries
            for(int j = 0; j < len2; j++){
                lcsTable[i][j] = 0;
            }
    }
}
```

LCSS Naive String Search KMP Boyer-Moore Rabin-Karp

Line 1 of the input file has plagiarised 100.0% from line 1 of the source file ADA.txt
Line 4 of the input file has plagiarised 62.5% from line 1 of the source file ADA.txt
Line 2 of the input file has plagiarised 100.0% from line 2 of the source file ADA.txt
Line 4 of the input file has plagiarised 64.70588235294117% from line 2 of the source file ADA.txt
Line 5 of the input file has plagiarised 64.70588235294117% from line 2 of the source file ADA.txt
Line 2 of the input file has plagiarised 76.92307692307693% from line 3 of the source file ADA.txt
Line 3 of the input file has plagiarised 100.0% from line 3 of the source file ADA.txt
Line 4 of the input file has plagiarised 84.61538461538461% from line 3 of the source file ADA.txt
Line 5 of the input file has plagiarised 84.61538461538461% from line 3 of the source file ADA.txt
Line 6 of the input file has plagiarised 92.3076923076923% from line 3 of the source file ADA.txt
Line 7 of the input file has plagiarised 76.92307692307693% from line 3 of the source file ADA.txt
Line 8 of the input file has plagiarised 76.92307692307693% from line 3 of the source file ADA.txt
Line 9 of the input file has plagiarised 76.92307692307693% from line 3 of the source file ADA.txt
Line 3 of the input file has plagiarised 70.58823529411765% from line 4 of the source file ADA.txt
Line 4 of the input file has plagiarised 72.2222222222221% from line 4 of the source file ADA.txt
Line 5 of the input file has plagiarised 77.7777777777779% from line 4 of the source file ADA.txt
Line 6 of the input file has plagiarised 72.2222222222221% from line 4 of the source file ADA.txt
Line 7 of the input file has plagiarised 72.2222222222221% from line 4 of the source file ADA.txt
Line 8 of the input file has plagiarised 61.11111111111114% from line 4 of the source file ADA.txt
Line 2 of the input file has plagiarised 63.1578947368421% from line 5 of the source file ADA.txt
Line 3 of the input file has plagiarised 70.58823529411765% from line 5 of the source file ADA.txt
Line 4 of the input file has plagiarised 63.1578947368421% from line 5 of the source file ADA.txt
Line 5 of the input file has plagiarised 84.21052631578947% from line 5 of the source file ADA.txt
Line 6 of the input file has plagiarised 68.42105263157895% from line 5 of the source file ADA.txt
Line 7 of the input file has plagiarised 63.1578947368421% from line 5 of the source file ADA.txt
Line 4 of the input file has plagiarised 92.0% from line 6 of the source file ADA.txt
Line 3 of the input file has plagiarised 64.70588235294117% from line 7 of the source file ADA.txt

KMP (KNUTH-MORRIS-PRAT):-

Knuth–Morris–Pratt string-searching algorithm (or KMP algorithm) searches for occurrences of a "word" W within a main "text string" S by employing the observation that when a mismatch occurs, the word itself embodies sufficient information to determine where the next match could begin, thus bypassing re-examination of previously matched characters. and sentiment analysis use cases. Naive Bayes Algorithm can be built using Gaussian, Multinomial and Bernoulli distribution. This algorithm is scalable and easy to implement for the large data set.

```
public class KMP {
    /**
     * Pre processes the pattern array based on proper prefixes and proper
     * suffixes at every position of the array
     *
     * @param ptrn
     *         word that is to be searched in the search string
     * @return partial match table which indicates
     */
    public int searchTimeCount= 0;
    public int preprocessingTimeCount= 0, timeCount=0;

    public int[] preProcessPattern(char[] ptrn) {
        int i = 0, j = -1;
        int ptrnLen = ptrn.length;
        int[] b = new int[ptrnLen + 1];

        b[i] = j;
        while (i < ptrnLen) {
            while (j >= 0 && ptrn[i] != ptrn[j]) {
                j = b[j];
                preprocessingTimeCount++;
            }
            i++;
            j++;
            b[i] = j;
        }
    }
}
```

LCSS Naive String Search KMP Boyer-Moore Rabin-Karp

Line 1 of the input file has plagiarised 100.0% from line 1 of the source file ADA.txt
Line 3 of the input file has plagiarised 84.61538461538461% from line 3 of the source file ADA.txt
Line 5 of the input file has plagiarised 100.0% from line 7 of the source file ADA.txt
Line 8 of the input file has plagiarised 63.1578947368421% from line 10 of the source file ADA.txt

Screenshots :-

The screenshot displays the Plagiarism Checker web application interface. The top navigation bar includes the logo, 'About', 'Contact', 'Help', and 'Language' options. The left sidebar contains navigation buttons for 'Online Plagiarism', 'Side by Side Comparison', 'Bulk Search', and 'Options'. The main workspace is divided into 'Source' and 'Target' text input areas, each with 'Import', 'Clear Highlighting', and 'Clear' buttons. Below the input areas, settings for 'Matching Limit' (set to 3), 'Case Sensitive', and 'Clean Invalid Spaces' are visible. A progress bar shows '70% Unique' in green and '30%' in red. A 'Remarks' section states 'Medium plagiarism detected, you better improve the document.' and a 'Scan Now' button is present. The footer includes social media links and a license notice: 'License is Registered to Andrew Smith.'

Plagiarism Checker

About Contact Help Language

Start Keywords

Online Plagiarism

Paste your Content Here

Highlight Color

Source

Target

Testing Plagiarism ABC College for Women is one of the most prestigious institutions of London with a full time enrollment of about 8000 students. The glorious academic values of this oldest premier post-graduate female institution have been shaped by its institutional history, which is spread over a span of 64 years. In January 2002, the University made all strong decisions for the improvement in Higher Education. Established in May 1962 as an Intermediate residential college and affiliated with the University of the Oxford, it was housed in a building on XYZ Road, with strength of 90 students and then the progress flourished with full shot. And College started programs like Electronics, Environmental Science, Fine Arts, Economics and Mass Communication. Various national industries and linkages with foreign Colleges helped a lot...

Since the establishment of ABC College for Women and in early January 2002, the University has tried its level best for improvement in Higher Education. Government did various national industries and linkages with foreign universities MoU with various national industries and linkages with foreign universities have been established in the field of Pharmacy, Electronics, Environmental Science, Fine Arts, Economics and Mass Communication. This is how they made the glorious academic values of this oldest premier post-graduate female institution very nicely

Import Clear Highlighting Clear

Import Clear Highlighting Clear

3 Matching Limit Case Sensitive Clean Invalid Spaces

70% Unique 30%


Remarks: Medium plagiarism detected, you better improve the document.

Scan Now



Connect with us

License is Registered to Andrew Smith.


Start
Search List
Results



 Online Plagiarism

Comparison Results

 Print
  Save

×



 Side by Side Comparison



Source

Tumblr Charmstotreasure 2.docx

shopping online for which time and money are of due significance. Numbers of online stores are out there selling **different types of** jewelry in various price ranges. Before buying any type of jewelry whether for your loved one, wedding anniversary, engagement rings or simply for your daily use, you must consider the following things. **There are number of** online stores and **you have to find the one** that has well reputed among the customers. You need to put some extra effort **in order to** eliminate the stores that are less desirable. **Make sure that you** are well aware of your budget before purchasing any jewelry. The best way to stick to budget is to narrow down your choices and find whether you need precious jewelry **made of gold or** silver or simply go **for women charms** and bracelets. The next thing **you have to** do is to search for the desired jewelry item and go for price comparison. By comparing the prices of different stores you will find an enormous difference in the prices of different stores. **If you are** going for a ring, bracelet or necklace,



Target

Tumblr Charmstotreasure.docx

Different Types of Charm Bracelets **to Go for** Nowadays the trend of wearing charm bracelets is increasing. **There are number of** charm bracelets out there varying in color, size, design and material. These types of bracelets allows number of charms to slide on to it. Either **made of gold or** silver, links and charms can be clipped onto it. The fresh style of bracelet, however, offers extra safe form of fastener. Not only are the charms more firmly attached to the bracelets these days, but the bracelets themselves have numerous different kinds of fastener. Lobster clasps are the most commonly used. These types of bracelets are very easy to use because the clasp can be opened quickly. For this reason, number of manufacturers also sells safety chains with bracelets. It provides extra security to the bracelet in order to remain fastened. Wide variety of materials is also available in these types of charm bracelets, making it easy for women to choose according to their budget. **There are number of** unbranded varieties available in the

×

Duplicate Found !

Source Document is 20% Duplicate with Target Document.

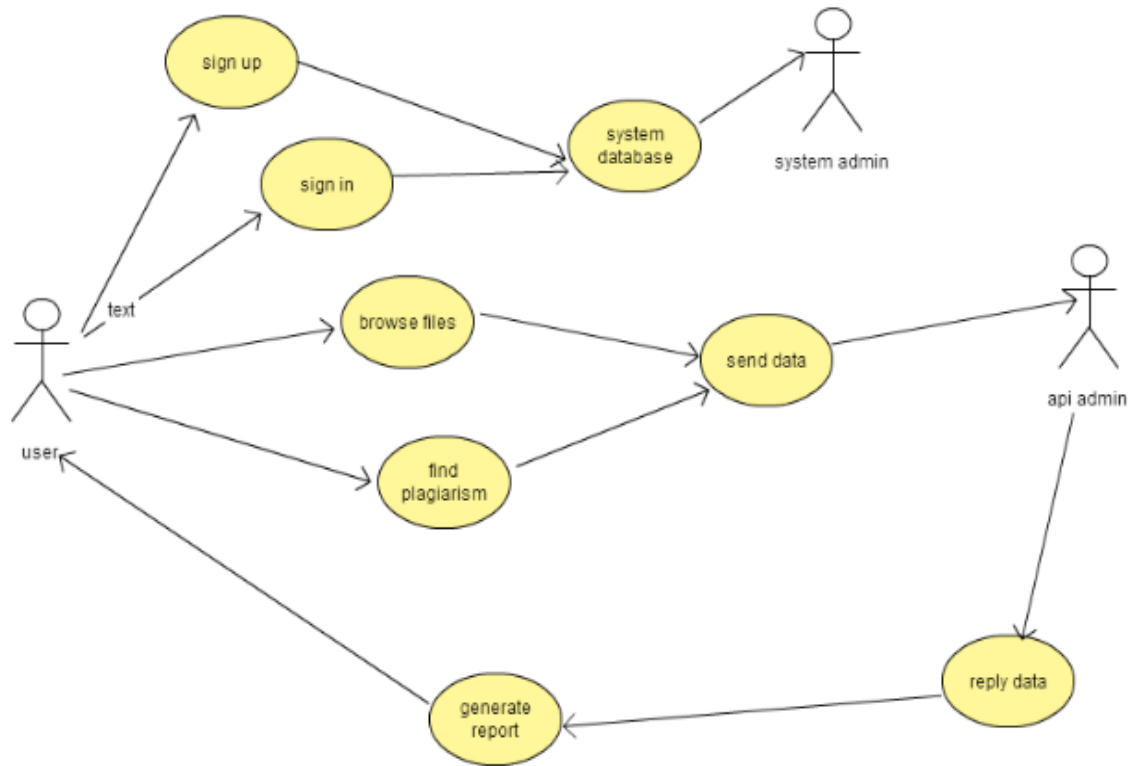
Remarks: Medium plagiarism detected, you better improve the document.

View Insight

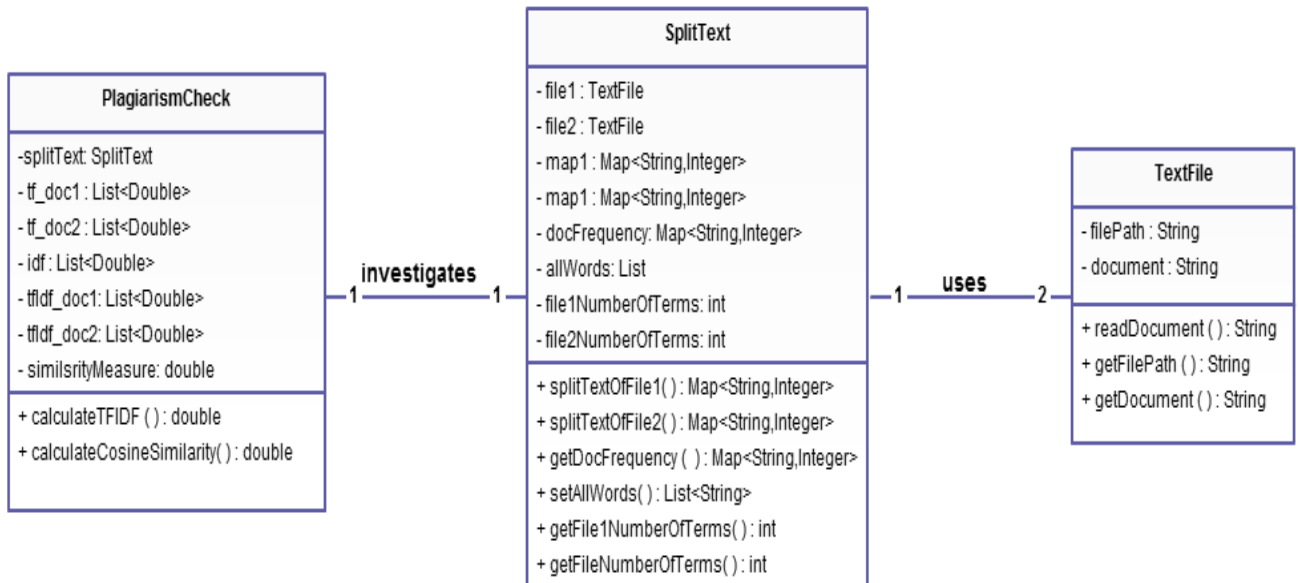
Connect with us

System Architecture:

• Use-Case Diagram



• Class Diagram



Conclusion

The conclusion of the design and implementation of plagiarism detection system on the programming task in a virtual classroom is that software plagiarism detection systems that are designed and tested can make or process any of the following.

- Plagiarism detection systems programming tasks in a virtual classroom with Moodle allows the lecturer to know the existence of similarities among students that if the task is done manually requires considerable effort, especially when the number of students attending is quite a lot.
- Plagiarism detection systems programming tasks in a virtual classroom with Moodle is able to show the percentage of similarity in student assignment, whether the similarity between the two files as well as similarities between one file into many files.
- In addition to show the percentage of similarity detection results, the system was able to show details of the contents of the detected files have similarities.
- System of alerts that are designed in this thesis, to address its students with a more efficient in terms of time, so that students can receive alerts/information as soon as the detection process is completed. Constraints faced in making the application of plagiarism detection system on the programming task in a virtual classroom with Moodle is on the submission of detail resemblance of the detected content file, because the submission of this information must go through the file attachment facility cannot be done on medium SMS.

