

AUTOMATED CODE INTEGRITY CHECKER: PLAGIARISM DETECTION AND GRADING EVALUATION TOOL FOR COLLEGE ASSIGNMENTS



G SANJITH S PREMKUMARAN S MUKILESH
Department of CSE, Sri Sivasubramaniya Nadar college of Engineering
Final Year Project, May 2024

Highlights of Proposed Model

To develop a system that helps to

- Detect plagiarism and error in codes of the submitted assignments.
- Regularize a systematic way of assignment submission.
- Evaluate similarity amongst submitted assignments along with its extent.
- Optimize grading process for faculties.

Challenges faced while creating model:

- Various template submission made information retrieval process difficult.
- Setting boundaries for extent of similarity.

Functional Modules and Dataset Description

- Information Retrieval
 - [Template Setting](#)
 - [Preprocess code](#)
 - Tokenization and Parsing
 - Comparator
 - [Sequence Matcher](#)
 - [Ratio \(\) Function](#)
 - Evaluation and Grading
- The dataset comprises submissions from 60 students for the "Fundamentals of C Programming" course, including assignments across eight exercises, each containing approximately 8 questions.
 - Rich data containing 4 percent error code , 4 percent of highly unique code , 30 percent of code with various levels of similarity and 36 percent of least unique codes.

Functional pipeline of Proposed System

Information Retrieval from LMS

- Gather student assignments from LMS and store as folders and save the submitted code in a N by X by X matrix, where N stands for number of student and X represents number of question.
- Standardized template has been developed to ensure consistency in submitted assignment submission

Tokenization and Parsing

- convert submitted codes into tokens entailing lexeme classification into Identifiers, Keywords, Operators etc.
- Using a well established set of 63 grammar rules, the given tokenized code is parsed and generates an individual metric score which is stored in a list.
- A LALR parser is used for this process due to its robust capability in handling a diverse array of grammar construct

Comparator

- Creates a matrix that serves as a comprehensive dataset containing metric scores for all students, facilitating further analysis and comparison.
- This metric value is obtained by comparing each code with the entire class and marking the level of similarity found in them.
- The range of this similarity score is from 0 to 1, where 1 is the highest possible value for plagiarism and 0 is the most unique code found.
- This similarity scores of each comparison is then stored in a NxN matrix, where the diogonal elements are 1, because of same code comparison.

Evaluator

- Student submissions are assessed: 0 for syntax errors, 1 for exact duplicates, and scores of 2, 3, or 4 for varying degrees of similarity to others, indicating high, moderate, or minimal resemblance, respectively.
- Average of the above score for each question is taken and grading is done accordingly to the score obtained.
- Grades are determined by a scoring system where scores below 1 result in an F, scores between 1 and 1.5 earn a C, 1.5 and 2 receive a B, 2 and 2.5 achieve a B+, 2.5 and 3 merit an A, 3 and 3.5 warrant an A+, and scores above 3.5 are awarded an O grade.

Performance Analysis

- Accurate Similarity Detection
- Error-Free Code Recognition
- Optimal Threshold Determination
- Optimize Evaluation and Grading

Performance metrics of the system

Label	Given Output	Expected Output	Accuracy
Error Program	1	1	100 percent
Exactly Copied	16	16	100 percent
Least Similar	2	2	100 percent
Moderately Similar	8	9	88 percent
Highest Similar	4	3	75 percent

Table 1. Performance Metrics for Assignment Analysed

Grades Obtained by the Class of 30 students

paul_andrew.docx	F
nikilesh.docx	C
niranjana.docx	C
rohan.docx	B
pravin_m.docx	B
preethi_pratrive.docx	B
nagesh.docx	C
anish.docx	C
mohammad_irfan.docx	C
naren.docx	C
prasanah.docx	C
prithvi.docx	C
muthu_selvi.docx	C
mandalal.docx	C
nishanth.docx	B
pawan.docx	B
pradeep.docx	C
niranjana.docx	B+
priyadarshini.docx	B+
nivetha_dhanakoti.docx	B+
mugil_krishna.docx	B+
nisha_ganesh.docx	B
neha_shamitha.docx	O
neeharika.docx	O
pranav_moorthi.docx	O
oviasree.docx	A
prathiyangira.docx	A+
nandhine.docx	A+
praneetha.docx	A
michael.docx	A

Inference

The development and implementation of a code plagiarism detector and error detection tool represent a significant step forward in enhancing the educational sector's integrity. By providing instructors with the means to detect plagiarism, identify errors, and deliver timely feedback on code submissions, this tool promotes academic integrity, and optimize evaluation process for faculties, and easier grading.

Proposed Model for Code Plagiarism Detector

