Plagiarism Detection in Source Code

Mrs. Nilkamal more
Assistant Professor
Department of Information technology
K. J. Somaiya college of engineering

Ms. Aditi Anil Bhootra
UG Student
Department of Information technology
K. J. Somaiya college of engineering

Ms. Charmi Arvind Patel
UG Student
Department of Information technology
K. J. Somaiya college of engineering

Abstract

Source code plagiarism is a very serious problem in academia. Lot of assignment work in programming courses is submitted electronically by the students. This makes it difficult for the faculty to check each and every code separately. Using a plagiarism detection tool makes it easy to check and analyze student's assignment. In programming courses, students submit their work in form of java source code files. There is a possibility that students may copy the Java code files from another source without properly crediting the original writer or programmer, intentionally or unintentionally. This is also a form of plagiarism. The main purpose of this paper is to show a method to detect plagiarism in java source code. To do this first it makes all the submitted java codes into a similar pattern by removing comments. Then tokenization is done. Finally the tokens are compared to get the similar portions and are displayed accordingly.

Keywords: Programming, code Plagiarism, Plagiarism, Plagiarism detection in Java Code, plagiarism detection tool.

I. INTRODUCTION

A. Definition:

Plagiarism is the "wrongful appropriation" and "stealing and publication" of another author's "language, thoughts, ideas, or expressions" and the representation of them as one's own original work. Plagiarism is considered academic dishonesty and a breach of journalistic ethics. It is subject to sanctions like penalties, suspension, and even expulsion. Recently, cases of 'extreme plagiarism' have been identified in academia [3]. Plagiarism is widely found in text, document's, papers, codes, images. The most common types of plagiarism are copy paste, find replace, mix different sources.

B. Plagiarism in Source Code:

Plagiarism in source code can be done in two ways:

1) lexical change
2) structural change

Lexical changes are the simplest type. Anyone can do lexical changes by just doing some simple editing's in the master code. Here programming knowledge is not required.

On the other hand structural changes need programming knowledge. Structural changes are changing iterations, changing conditional statements, changing the order of statements, changing procedure to function and vice versa, changing procedure call within the body of the procedure and vice versa, adding statements that will not affect the output of the program.

C. Plagiarism in Acadamics:

Plagiarism detection is the process of locating instances of plagiarism within a work or document [5].

In Computer Science, like in other disciplines, it is integral that students not only understand the presented material but are also able to apply what they have learned in a practical setting. Therefore, most computer science courses expect students to submit programs as part of their laboratory works and assignments. Generally, plagiarism in coding is hard to be detected because of the similar coding used for the same application. Plagiarism in coding is easy to do, but difficult to detect .Students copy all or part of a program from some source or from different sources and submit the copy as their own work. Such plagiarism is very common, though the true extent is hard to assess. When a teacher in a programming course gives same assignment problems to all students then all students have to work on same problems. So it is the possibility that some students write source code of problems by their

Own and remaining students just take the code from them and amend it like changing of variable names, changing the order of statements, functions and variables of class and submit it. These types of modifications in source code are very difficult to catch. It is difficult to measure the extent of plagiarism by manual inspection of so many files submitted by the students. Manual

inspection thus makes results less efficient and inaccurate. This paper shows a simple method to show similar portions in java code files by which the teacher can predict whether the student as copied or not.

II. PLAGIARISM DETECTION

The application developed has two file browsing options. The first one is where we upload the root folder containing the master java source code file. The other browse option uploads the folder containing all the java source code files submitted by students that are to be checked for plagiarism. After uploading the files successfully the initial step is to remove the comments from all the codes if present. Once the comments are removed tokenization is performed. A token file is created for each source code. Then the comparison of every token file is done. The result of each comparison is a value called percent match. If the percent match of a pair of token files is larger than this minimum value, then the corresponding pair will be judged as a case of suspected plagiarism. The file in the root folder is checked with every file in the folder having the submitted files. Once the similarity detection is performed the portions similar of the submitted file to the master file is displayed. The final result displays the similar portions of the master source code and submitted source code.

Below is a brief description of all the steps followed for similarity detection in java source codes:

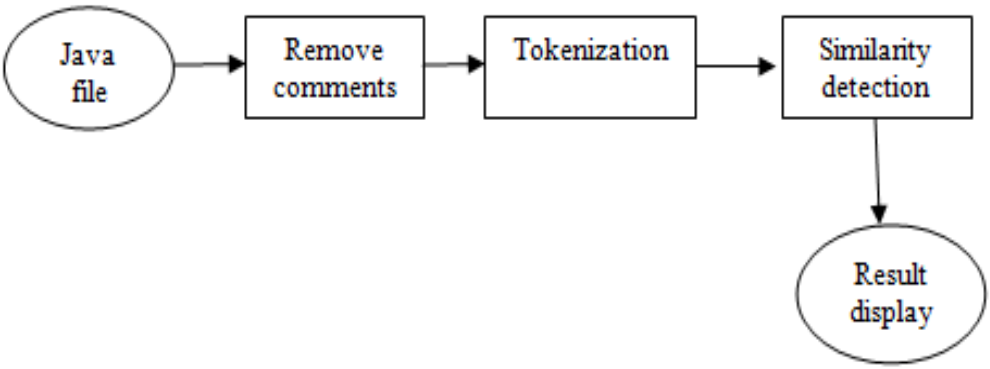

Fig. 1: Plagiarism Detection

A. Input Java Files:

The application will allow user to upload only .java extension files. If user uploads some other extension file it displays an error message that the file is not java file. Thus, user has to upload only java files.

B. Normalization:

Once the files are uploaded correctly the next step is to remove all the comments from the source code. Comments do not have any effect on functionality of the code. Some students only add comments in other's code and submit it as different code. Hence, to verify whether the code is plagiarized or not removing comments is important.

C. Tokenization:

When characters of a Java program are grouped into symbols then it is called as token. A token can be an identifier, keyword, separator, operator, literal and comment. Programmer choose the identifies; keywords are names that are already defined in a programming language; separators are punctuators; operators are symbols that produce results by operating arguments; Literals can be Numeric Textual, Logical and reference and comments are line or block [2].Each line of source files is divided into tokens corresponding to a lexical rule of the programming language. The tokens of all source files are concatenated into a single token sequence, so that finding clones in multiple files is performed in the same way as single file analysis. From all the substrings on the transformed token sequence, equivalent pairs are detected as clone pairs. Each location of clone pair is converted into line numbers on the original source files.

D. Similarity Detection:

The similar parts from the master file and submitted files are calculated after tokenization is done. The similarity detections can check plagiarism in the following conditions:

1) *The entire code is copy/paste and submitted*
2) *The order of the statements is interchanged. For e.g. : code written in lines 5 to 15 is changed to lines 7-17*
3) *Changing iterations, changing conditional statements*
4) *Changing variables*

E. Result Display:

Finally the detected similar parts are displayed.

III. RESULTS AND DISCUSSION

The working of code plagiarism application:

A. Start Application:

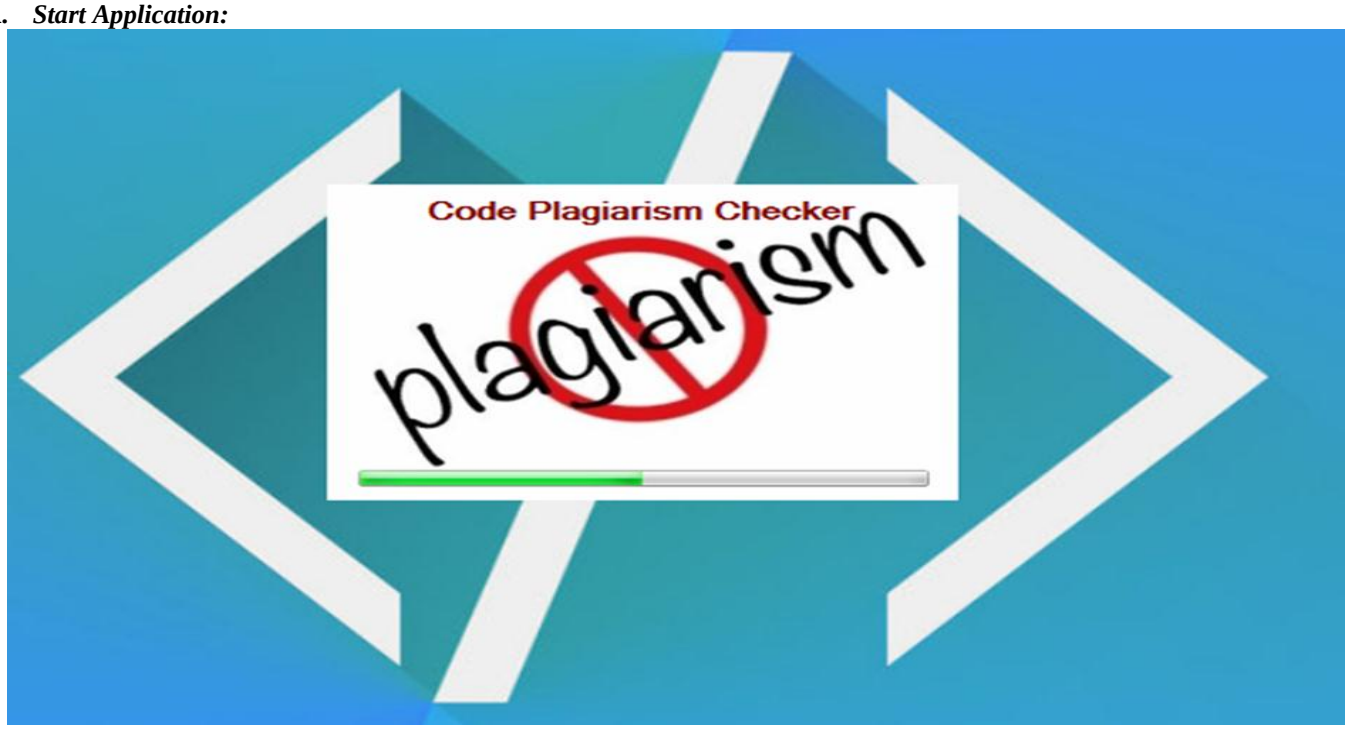

Fig. 2: Start Application:

B. Upload Files:

Folder 1 contains files:

1) *demo.java*
2) *fact.java*
3) *mla.java*
4) *test.java*

Folder 2 contains files:

5) *factorial.java*
6) *manufacturer.java*
7) *tutorial.java*

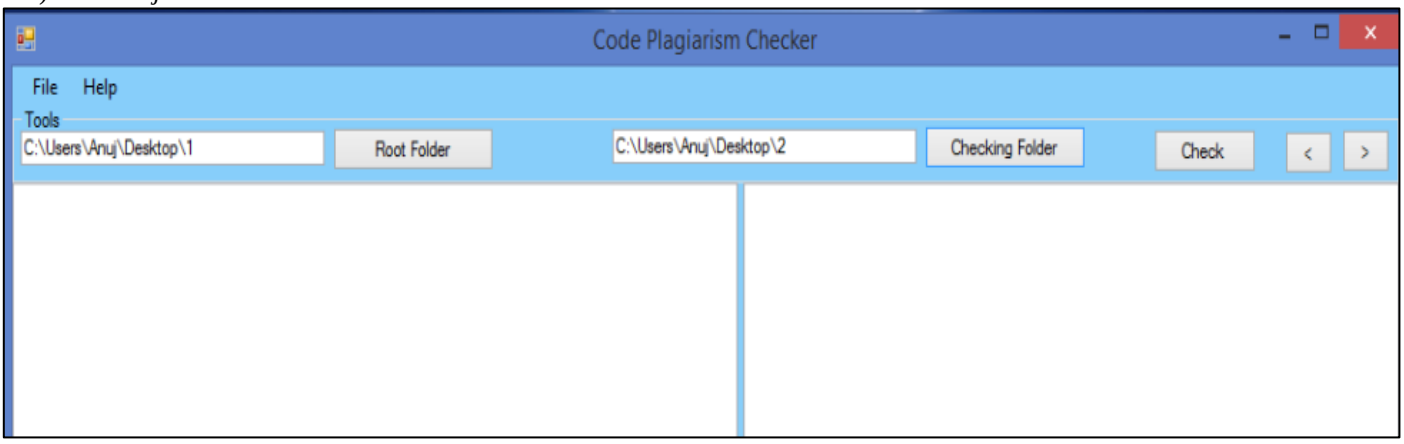

Fig. 3: Upload Files

C. Display Results:

Here the master file i.e. MLA file is compared with the submitted file i.e. manufacturer file. Plagiarism is detected and the similar parts are displayed below. It shows which lines from master file are similar to submitted files. It also displays the number of tokens matching.

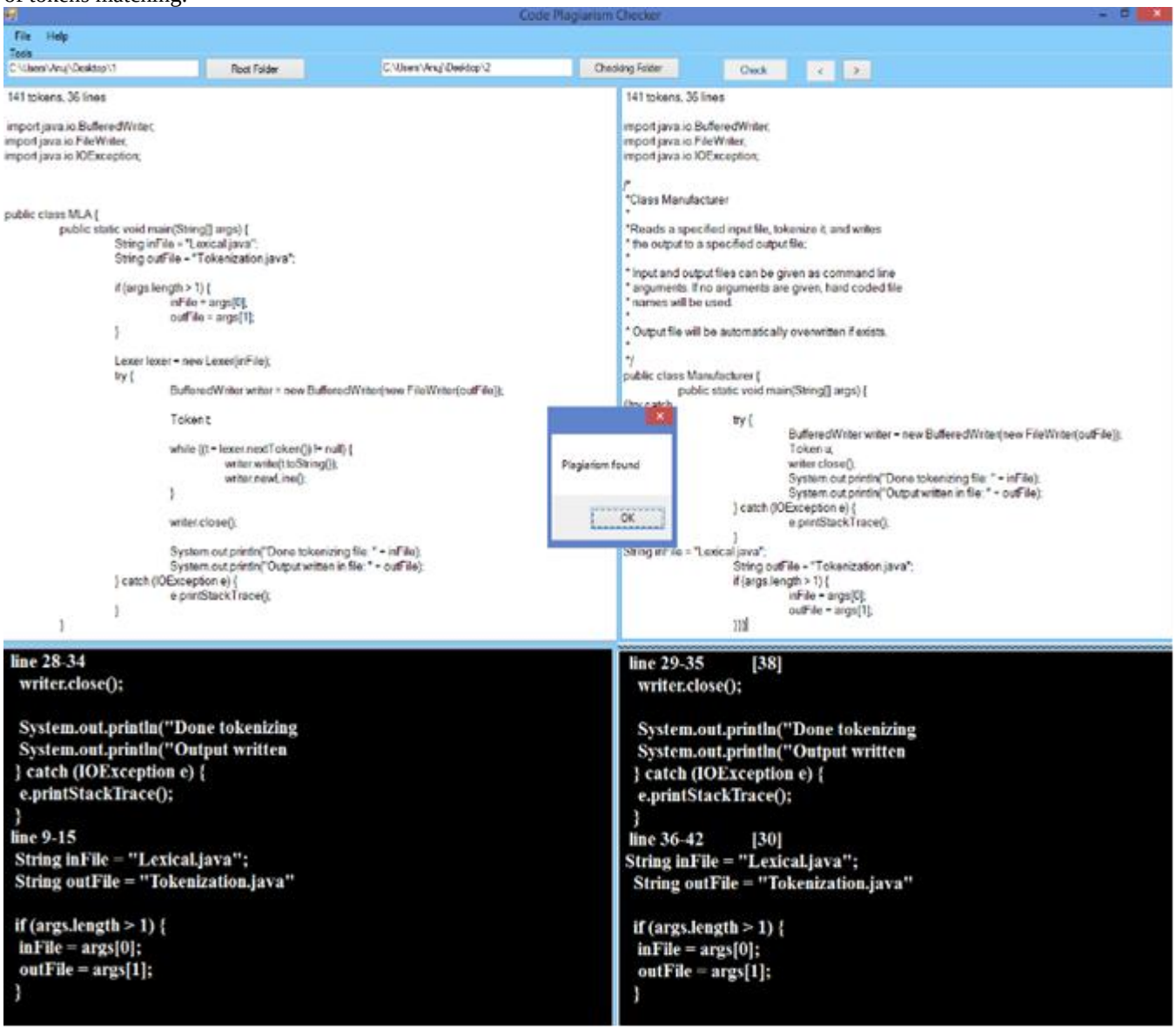

Fig. 4: Display Results

IV. CONCLUSION

In this paper we have shown a method to detect plagiarism in java source codes. It shows how codes different ways in which students can perform plagiarism in source codes and how it is detected. Based on the results obtained the teacher can then make more efficient and accurate results in student's assessment. The teacher can make better decisions by using this application than buy manual inspection. Students will also be graded efficiently according to their originality in their work.

REFERENCES

[1] Wang Kechao, Wang Tiantian, Zong Mingkui, Wang Zhifei, Ren Xiangmin**,** "Detection of plagiarism in students' program using a data mining algorithm", Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference
[2] Ahmad Gull Liaqat & Aijaz Ahmad, "Plagiarism Detection in Java Code".
[3] http://en.wikipedia.org/wiki/Plagiarism.
[4] http://www.plagiarism.org/plagiarism-101/types-of-plagiarism/
[5] http://en.wikipedia.org/wiki/Plagiarism_detection