International Journal of Innovative Computing, Information and Control
Volume **5**, Number **11(B)**, November **2009**

ICIC International ©2009 ISSN 1349-4198
pp. 4237–4247

A NEW SIMILARITY MEASURE FOR IN-CLASS SOURCE CODE PLAGIARISM DETECTION

Asako Ohno[1] and Hajime Murao[2]

[1]Department of Life Design
Shijonawate Gakuen Junior College
4-10-25 Hojo, Daito, Osaka 574-0011, Japan
asako.ohno@mulabo.org

[2]Graduate School of Intercultural Studies
Kobe University
1-2-1 Tsurukabuto, Nada, Kobe 657-8501, Japan
murao@i.cla.kobe-u.ac.jp

Received October 2008; revised February 2009

Abstract. *It is a laborious task for teachers to detect plagiarism in source codes produced by the students in programming classes. It is also difficult to distinguish between plagiarism and coincidental similarity in these source codes, since they are (1) often too short to extract enough algorithmic features and (2) obviously similar to each other because they are produced for the same purpose. We propose a new method to measure similarity between source codes to detect source code plagiarism. Our method does not make pair-wise comparisons to find copied fragments among different students' works; rather, it compares the coding style of a newly submitted source code with a number of source codes that have been produced by the same author. The coding style–a superficial feature appearing in source codes produced by the same author–is represented by a set of HMM-based stochastic models called "Coding Models" and utilized to make author identifications. We conducted an experiment and confirmed that the coding models could distinguish between source codes produced by different students, even if they were algorithmically quite similar to one another, thus indicating that our method can provide useful information for teachers to detect in-class source code plagiarism.*

Keywords: In-class source code plagiarism, Plagiarism detection, Coding model, Coding style, CM algorithm, Hidden Markov model

1. **Introduction.** In academia, the number of programming classes is increasing every year [1]. This tendency has given rise to a serious and growing problem: source code plagiarism. Source codes produced by students in programming classes are generally short in length and algorithmically quite similar to each other because they are produced for the same purpose in accordance with a teacher's instructions. We call this type of source code *in-class source codes*. Generally, in-class source codes are easily plagiarized because they are produced and handled digitally, and detecting plagiarism among source codes is a difficult and laborious task for teachers in charge of programming classes. Therefore, from the viewpoints of both pedagogical and information science fields, source code plagiarism is now considered one of the most serious issues in both pedagogical and information science fields [1-6]. A number of methods have been proposed to automate this task [7-14].

The definition of plagiarism differs among different studies [1]. With regard to in-class source code plagiarism, we classified different types of plagiarism into the following three categories depending on how plagiarism was committed.