



HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Tools for plagiarism detection

Kristian Wahlroos
April 24, 2017

University of Helsinki
Department of Computer Science



Outline

Introduction

Problem

Plagiarism

Motivation

Methodology

Literature review

Results

Overview

Data

Methodologies

Feature extraction

Discussion



Introduction

- MOOC's have gained popularity in recent years
 - Especially programming related MOOC's¹
 - Independent assignments
 - No live-presence required
- Number of students often large
- Trust is thus usually one-sided
 - Belief that students do tasks by themselves
 - Not actively monitored
 - Cheating is in form of plagiarism
 - Many potential plagiarism scenarios

¹<http://blog.edx.org/>



- Source code plagiarism is a problem consisting many forms
 - Straight plagiarism
 - Too intense group work
 - Code sharing
 - Obfuscation
- Lots of students → impossible to detect manually in reasonable time
 - Lot of data available
 - Need for automated tools



In this study

- Finding a suitable machine learning tool set for detecting source code plagiarism
- Motivated by
 - Could be used in University of Helsinki's course *Introduction to programming*
 - Interesting topic
 - Machine learning methods benefit from a lot of data
- Results reflected to the usage in a academic course



Methodology

- Performing literature review with *Google Scholar*
- Collected 8 papers
- Two-step search process
 - Limit by overall keywords occurrences
 - Limit by title/abstract/keywords
- Keywords
 - Direct matches: **machine learning, plagiarism, code, programming**
 - Non-direct: **authorship, identification**



- Limited years starting from 2006
 - Believed to contain more recent programming languages
 - MOOC's are relatively new concept
 - Machine learning methods have changed
- Doing comparison between papers
 - Model accuracy
 - Data
 - Machine learning methodology
 - Feature extraction



Results

- 8 papers from 2007 to 2015
 - 1) *A machine learning based tool for source code plagiarism detection*, 2011
 - 2) *De-anonymizing programmers via code stylometry*, 2015
 - 3) *Detecting outsourced student programming assignments*, 2008
 - 4) *Pde4java: Plagiarism detection engine for java source code: a clustering approach*, 2008



- 5) *A probabilistic approach to source code authorship identification, 2007*
- 6) *Using code metric histograms and genetic algorithms to perform author identification for software forensics, 2007*
- 7) *Who wrote this code? identifying the authors of program binaries, 2011*
- 8) *An application for plagiarized source code detection based on a parse tree kernel, 2013*



- Studies divide into two categories
 - Attribute counting (4 papers)
 - Structure based (4 papers)
- Attribute counting is easy and fast → directly from the source code
- Structure based require parsing
- Model accuracies are reported in two ways
 - Traditional classification accuracy
 - How close the model was to human labeling



- Accuracies ranged from 69% to over 90%
 - Highest used mixture of stylistic and structural approach
 - E.g. 93% same results compared to human validator
- Plagiarism detection is close to authorship identification
 - Classifying anonymous source code
 - Clustering similar documents together
 - Finding stylistic nuances
 - Trying to capture the logical structure



Data sets in studies

- No clear difference between stylistic studies and structural studies
 - Partially reported data sets
 - Just few authors in stylistic studies

Attr./Paper	1	5	3	6	8	2	4	7
Size	741	200	83	4068	555	N/A	326	203
Authors	10	8	12	20	N/A	1600	N/A	32
Structural	No	No	No	No	Yes	Yes	Yes	Yes

Table: Reported data sets used in papers



- Data sets are often collected from course assignments
- Open source projects are utilized to gather the data set
- Competitions
 - *Google Code Jam*
 - Explains the large number of possible authors



Machine learning methods

Attr./Paper	1	5	3	6	8	2	4	7
Method	E_3	NB/VFI	DT	GA	PTK	RF	DM	K-M/SVM
Structural	No	No	No	No	Yes	Yes	Yes	Yes

Table: Methods used in studies

E_i	Ensemble of i models
NB	Naive Bayes
VFI	Voting Feature Interval
DT	Decision Tree
GA	Genetic Algorithm
PTK	Parse Tree Kernel
RF	Random Forest
DM	Data Mining (DBSCAN)
K-M	K-means Clustering
SVM	Support Vector Machine



- Many various algorithms used
 - Probabilistic
 - Trees
 - Genetic algorithm
 - Clustering
- Structural studies tend to favor tree-structures and clustering
- More variance in stylistic studies
 - Probabilistic models
 - Genetic algorithm
 - Decision tree



Yet another slide

content