

LB/DON/25/2012

LIBRARY
UNIVERSITY OF MORATUWA, SRI LANKA
MORATUWA

Plagiarism Detection Tool for Students' Programming Assignments

Upul Bandara

MS IT 05/10036



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

004 "11"

004.8 (043)

TM

University of Moratuwa



102505

Dissertation submitted to the Faculty of Information Technology, University of
Moratuwa, Sri Lanka for the partial fulfillment of the requirements of the Degree of
MSc in Information Technology

102505

March 2011

102505

Declaration

I declare that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a Degree or a Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organization.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Name of the Student: M.M.C.U.Bandara

Signature of Student: upul Bandara

Date: 14-11-2011

Supervised by

Name of the Supervisor: Dr. Gamini Wijayarathna

Signature of Supervisor: Gamini Wijayarathna

Date:

Dedication

I dedicate this thesis to Dr. Gamini Wijayarathna with a heart full of gratitude. Without his guidance, support, commitment and his uncommendable patience as the supervisor, the successful completion of this work would not have been possible.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Acknowledgments

I would like to extend my sincere gratitude to Dr. Gamini Wijayarathna for his keen supervision, valuable advice, and dearly guidance from the initial stage of the project. It is also my duty to mention here that I have gained a vast knowledge on every aspect of Information Technology by completing this project under the supervision of Dr. Wijayarathna.

Further, I must make this an opportunity to thank my family members, colleagues and friends who helped me to make the project a success in every possible way.

Finally, many thanks go to the academic and non-academic staff members of the University of Moratuwa for their support in various ways.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Abstract

With the rapid development of Internet technology and freely availability of academic materials in electronic format, plagiarism has become a key issue in universities and colleges. Therefore researchers have been developing various tools to address plagiarism.

Plagiarism can be occurred in any academic field, but developing a generic tool to address all kind of plagiarism is not feasible. So we developed a product to detect plagiarism in programming assignments submitted by students those who are taking programming courses in universities and colleges.

Plagiarism detection in programming source codes can be done by various methods. Our proposed solution uses machine learning approach to plagiarism detection. In addition our proposed has been using several learning algorithms and ensemble learning approach to enhanced performance of the system.

At present this system is outperforming some plagiarism detection tools which are based on the machine learning approach to detect plagiarism.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Table of Contents

Chapter 1 - Introduction	1
1.1 Introduction	1
1.2 Background and Motivation	1
1.3 Problems and Weaknesses of Existing Approaches	2
1.4 Aim and Objective	3
1.5 Proposed Solution	3
1.6 Expected Outcomes	4
1.7 Structure of the Dissertation	4
Chapter 2 - Literature Review.....	5
2.1 Introduction	5
2.2 Source Code Linearization Techniques for Detecting Plagiarized Programs [8]	5
2.3 Plagiarism Detection using Feature-Based Neural Networks [10]	6
2.4 Code Metric Histograms and Genetic Algorithms to Perform Author Identification for Software Forensics [9]	7
2.5 Plagiarism Detection across Programming Languages [3]	9
2.6 Summary	9
Chapter 3 - Machine Learning Techniques for Pattern Recognition	11
3.1 Introduction	11
3.2 Naïve Bayes Classifier	11
3.2.1 How to Apply Naïve for Document Classification	12
3.3 k-Nearest Neighbor Algorithm(kNN)	14
3.4 Summary	14
Chapter 4 - Pattern Recognition Techniques for Source Code Author Identification	16
4.1 Introduction	16
4.2 Source Code Metrics	16
4.3 Converting the Source Code Metrics to a Set of Tokens	18
4.4 Naïve Bayes Classifier for Source Code Author Identification	20
4.4.1 Maximum Likelihood Estimators for Bernoulli and Multinomial Naïve Bayes Learners	21
4.5 k-Nearest Neighbor (kNN) Algorithm for Source Code Author Identification	22

4.6 Summary	22
Chapter 5 - Implementing the Source Code Author Identification System	24
5.1 Introduction	24
5.2 Data Set	24
5.3 High-Level Architecture of the System	25
5.4 Training the System	25
5.4.1 Training the Multinomial Naïve Bayes Learner	26
5.4.2 Training the Bernoulli Naïve Bayes Learner	28
5.4.3 Training the k-Nearest Neighbor (kNN) Learner	30
5.4.4 Ensemble Learning	31
5.4.5 Different Ways of Combining Weak-Learners	33
5.4.6 Using AdaBoost to Improve the Accuracy of Source Code Author Identification Process	34
Chapter 6 - Evaluation and Results	35
6.1 Introduction	35
6.2 Evaluation by the Validation Dataset	35
6.3 Comparing the Research Results with the Results Published by Robert Lange and et al [9]	37
6.4 Summary	38
Chapter 7 - Conclusion and Further Works	39
7.1 Introduction	39
7.2 Different ways to improve the accuracy of the system	39
7.2.1 Changing the Training Dataset Size	39
7.2.2 Using More Weaker-Learners	39
7.3 Limitations of the System	40
7.4 Addressing the Limitations of the System	40
7.5 Summary	41
References.....	42
Appendix A - Machine Learning Algorithms Used in the Research.....	43
Appendix B - Contents of the CD-ROM.....	53

List of Figures

Figure 5.1 - High-level Architecture of the System	25
Figure 5.2 - Confusion matrix for multinomial naïve bayes for 100 source code files	26
Figure 5.3 - Confusion matrix for multinomial naïve bayes for 800 source code files	27
Figure 5.4 - Confusion matrix for multinomial naïve bayes for hideout data set	27
Figure 5.5 - Confusion matrix for Bernoulli naïve bayes for 100 source code files	28
Figure 5.6 - Confusion matrix for Bernoulli naïve bayes for 800 source code files	29
Figure 5.7 - Confusion matrix for Bernoulli naïve bayes for hideout dataset	29
Figure 5.8 - Success Rate vs. Sample Size of the kNN algorithm training dataset	30
Figure 5.9 - Confusion matrix of kNN algorithm for hideout dataset.	31
Figure 5.10 - Confusion matrix of the AdaBoost algorithm for the hideout dataset	34
Figure 6.1 - Confusion matrix for multinomial naïve bayes for hideout data set	35
Figure 6.2 - Confusion matrix for Bernoulli naïve bayes for hideout dataset	36
Figure 6.3 - Confusion matrix of kNN algorithm for hideout dataset.	36
Figure 6.4 - Confusion matrix of AdaBoost algorithm for hideout dataset	37
Figure A.1 - Naïve Bayes Multinomial Algorithm	44
Figure A.2 - Confusion matrix for multinomial naïve bayes for 200 source code files	44
Figure A.3 - Confusion matrix for multinomial naïve bayes for 300 source code files	44
Figure A.4 - Confusion matrix for multinomial naïve bayes for 400 source code files	45
Figure A.5 - Confusion matrix for multinomial naïve bayes for 500 source code files	46
Figure A.6 - Confusion matrix for multinomial naïve bayes for 600 source code files	46
Figure A.7 - Confusion matrix for multinomial naïve bayes for 700 source code files	46
Figure A.8 - Naïve Bayes Bernoulli Algorithm.	47

Figure A.9 - Confusion matrix for Bernoulli naïve bayes for 200 source code files	48
Figure A.10 - Confusion matrix for Bernoulli naïve bayes for 300 source code files	49
Figure A.11 - Confusion matrix for Bernoulli naïve bayes for 400 source code files	49
Figure A.12 - Confusion matrix for Bernoulli naïve bayes for 500 source code files	50
Figure A.13 - Confusion matrix for Bernoulli naïve bayes for 600 source code files	50
Figure A.14 - Confusion matrix for Bernoulli naïve bayes for 700 source code files	51
Figure A.15 - kNN Algorithm	51
Figure A.16 - Ada Boost Algorithm.	52



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

List of Tables

Table 2.1 - Recall and Precision for Plagiarism Detection using Feature-Based Neural Networks	6
Table 2.2 - Source code metrics used by Robert Lange and et al [9]	8
Table 2.3 - Performance of Plagiarism Detection across Programming Languages Technique at Various Precision Levels	9
Table 3.1 - Multinomial vs. Bernoulli Models	13
Table 4.1 - Source Code Metrics Used for Source Code Author Identification	17
Table 4.2 - Document Collection for Naïve Bayes Example	18
Table 4.3 - Coding System of Source Code Metrics	19
Table 4.4 - Output of LineLengthCalculator Metric	20
Table 4.5 - Token Frequencies of LineLengthCalculator Metric	20
Table 6.1 - Comparing results between our the research system and the system developed by Robert Lange and et al [9]	38



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

List of Equations

Equation 2.1 - Nearest Neighbor Algorithm	8
Equation 3.1 - Naïve Bayes Algorithm.....	12
Equation 3.2 - Probability of Document d being in Class c	12
Equation 3.3 - The Best Class in the Naïve Bayes Classification.....	12
Equation 3.4 - Euclidian Distance Between x_i and x_j	14



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk