Code signatures for plagiarism detection

Dennis Goßler   Dennis Wäckerle

October 30, 2022

Abstract

Contents

1 Introduction

2 How Does Plagiarism Detection Work

3 Use Case and Software Experiment

3.1 Use Case

3.2 Software Experiment

4 Criteria for Evaluation

5 Evaluation of the different Tools

5.1 MOSS

5.2 JPlag

5.2.1 JPlag's comparison algorithm

- Functions in two phases 1. all programs are parsed and converted into tokens 2. tokens strings are compared in pairs. Tries to cover one token stream with substrings of the other token. Percentage of covered token streams is the similarity. (Perchelt et al., 2000, p. 10)

- Tokenizing -> only language dependent process(Perchelt et al., 2000, p. 10) - Tokens represent syntactic elements e.g. statements or control structures(Sağlam, 2022, How are submissions represented? — Notion of Token)

- Transformation - each file is parsed -> result set of Abstract Syntax Trees for each submission - each AST traversed depth first, nodes are grammatical units of language - when entering and exiting a node a token can be created and added to the token list - block type node e.g. classes or if expressions have corresponding begin and end tokens. Token list should have balanced pairs of matching begin and end tokens

- Comparing token strings

5.2.2 The results

5.2.3 Integration into an automated evaluation pipeline

5.3 Plaggie

5.4 AC2

6 Conclusion

References

Perchelt, L., Malphol, G., & Phlippsen, M. (2000, March 28). *Jplag: Finding plagiarisms among a set of programs*.

Sağlam, T. (2022, September 16). *4. adding new languages*. Retrieved October 30, 2022, from https://github.com/jplag/JPlag/wiki/4.-Adding-New-Languages

Perchelt, L., Malphol, G., & Phlippsen, M. (2000, March 28). *Jplag: Finding plagiarisms among a set of programs*.

Sağlam, T. (2022, September 16). *4. adding new languages*. Retrieved October 30, 2022, from https://github.com/jplag/JPlag/wiki/4.-Adding-New-Languages