



Nailah Saleh Alhassoun

Introduction

In the age of information technologies, plagiarism has become more actual and turned into a serious problem especially in university courses with programming assignments. Detecting plagiarism manually is a challenging task because it is difficult and time-consuming. Hence, automated search tools are particularly helpful in detecting similarity between pairs of programs. Plagiarism detection systems are generally grouped into two classes: Attribute-counting systems, and structure-based systems. We propose a new plagiarism detection system, called CAPlag "Computing Assignment Plagiarism", for Java programming assignments, in which we exploit both attribute-counting and structure-based properties in order to identify lexical and structural changes. Our contribution consists in using a Java program profiling to extract the main program characteristics and a comparison method inspired by DNA sequencing. Experimental results show that CAPlag can outperform JPlag, a well-known plagiarism detection system, especially on small program instances.

Objectives

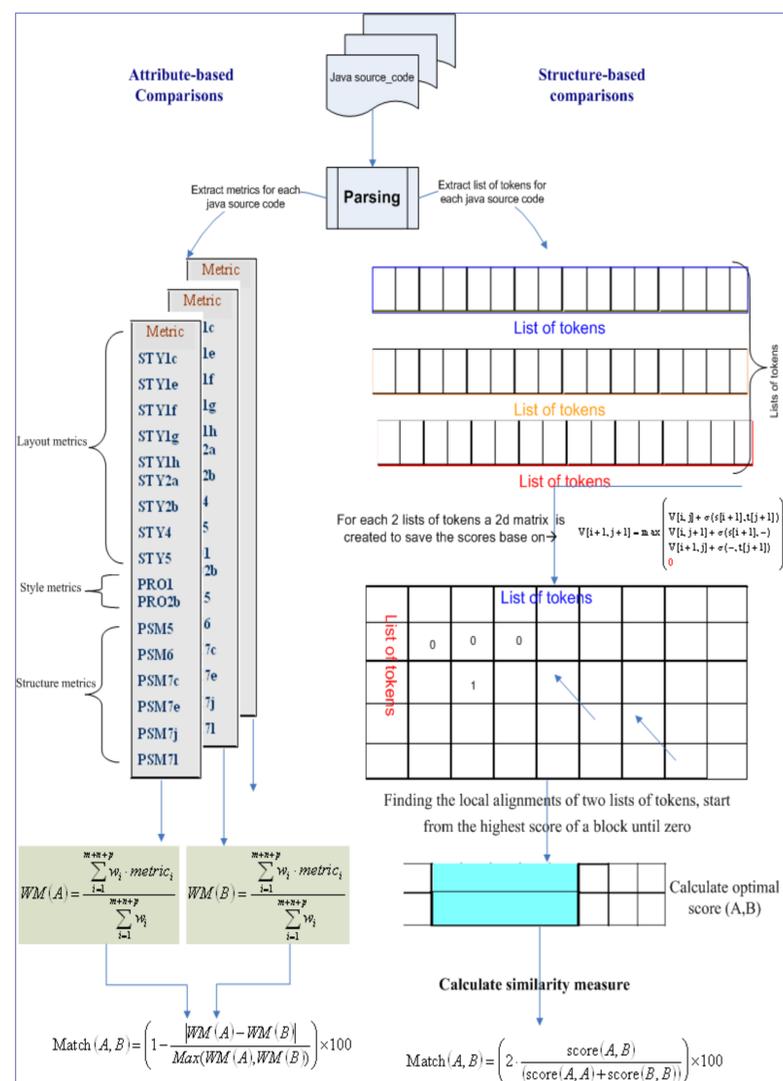
Take advantage of existing detection techniques to produce a competitive system which may assist Java course instructors in indentifying plagiarism student cases.

Methodology

CAPlag consists of two main components:

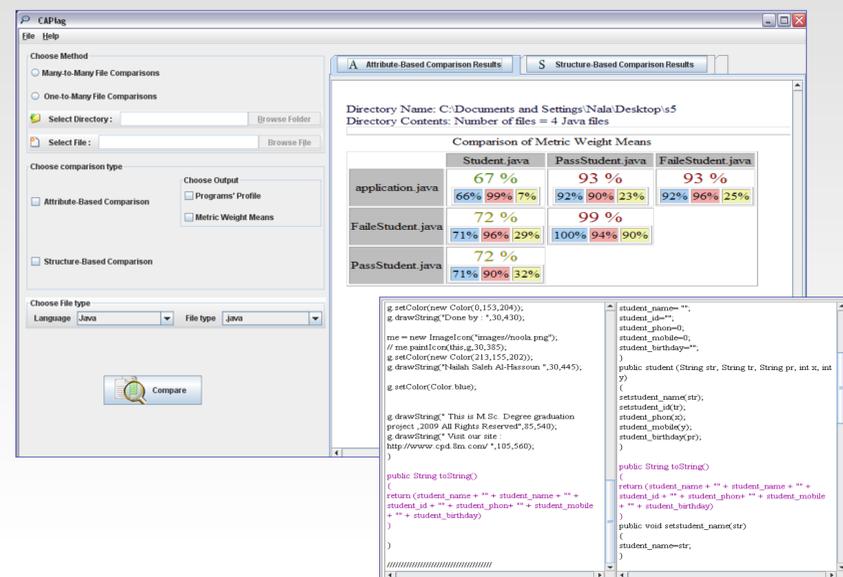
Attribute-counting comparison

- Could be used as a first phase for fast plagiarism detection [1].
- Can be used as authorship identification.
- **Structure-based comparison**
- Could be used to detect more intricate plagiarism cases [2,3].
- DNA local alignment method is used to compare structure of programs.

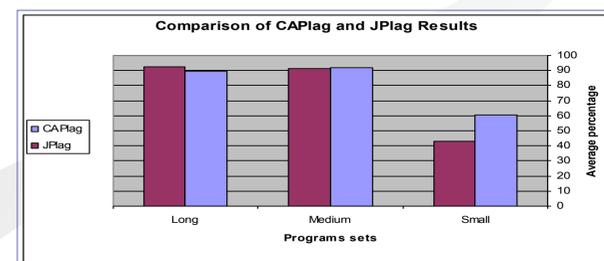


Results

- A stand-alone Java application, called CAPlag, has been implemented with an interactive and easy-to-use interface.
- Qualitative and quantitative results can be visualized.



- We compared CAPlag with JPlag tool which is considered as among the best tools for detecting plagiarism in computing assignments [2, 3]. We tested them on sets of programs consisting of a dummy test set created by hand, with modified levels using widely-known techniques of plagiarism. We considered 3 different sets according to program sizes (small, medium, and long programs)



Experimental results show that both tools can detect similarities in programs with high success rates. CAPlag detected all the similarities in small program instances with better performance than JPlag.

Conclusion

We introduced a new plagiarism detection system (CAPlag) allowing both attribute-counting and structure-based comparisons. Attribute-counting comparison is based on programs' profiling and comparison of their software metric features. Structure-based comparison is based on programs' normalization and alignment using a dynamic programming alignment method.

CAPlag is a freeware software available online (<http://cpd.8m.com>). We hope that instructors will use it and send us their feedback which can help us to improve its performance and to incorporate new features.

Acknowledgement

I would like to express my sincere gratitude and appreciation to my supervisor *Dr. Mohamed El Bachir Menai* for his effort and encouragement throughout the preparation of this project.

References

- [1] A. Parker and J. Hamblen, "Computer Algorithms for Plagiarism Detection," *IEEE Transactions on Education*, vol. 32, no. 2, pp. 94-99, 1989.
- [2] G. Malpohl, L. Prechelt, and M. Philippsen, "Finding plagiarisms among a set of programs with JPlag," *Journal of Universal Computer Science*, vol. 8, no. 11, pp. 1016-1038, 2002.
- [3] F. Culwin, A. MacLeod, and T. Lancaster, "Source Code Plagiarism in UK HE Computing Schools, Attitudes and Tools," South Bank University, London, Tech. Rep. (SBU-CISM-01-01), 2001.