# Plagiarism Detection Software

Marlin Thomas
Department of Computer Science
and
Samuel Rudin Academic Resource Center
Iona College
New Rochelle, NY 10801

**Abstract**

Plagiarism is a pervasive form of academic dishonesty in collegiate settings. Since it distorts learning and assessment, deterring and detecting it are crucial to maintaining academic integrity. Large class sizes and an increase in writing assignments that result from writing across the curriculum combine to make detection of plagiarism burdensome. The concomitant rapid increase of written material on the Internet and its ease of appropriation contribute to the problem. Plagiarism detection software has emerged in response. The most prominent implementation compares submissions against items in a database and then adds them to that database. It outputs measures of possible plagiarism. Faculty use of the detection software should be reevaluated because of issues related to its efficacy and because of ethical and legal concerns.

## Introduction

Plagiarism fundamentally warps two essential aspects of education, learning and assessment. Students who submit plagiarized work deprive themselves of the learning opportunities afforded by authentic academic productions and by assessments of those productions by educators. Plagiarism substitutes the physical labor of theft and misrepresentation for the labor and growth of learning. Moreover, plagiarism erodes the sense of community that is essential to free academic inquiry. Misrepresentation masks the true identity of members of the community, and it propagates unfairness in the community's awarding of prestige and status. Faculty, therefore, have a professional obligation to deter and to detect plagiarism, but they need to do so as part of a cooperative learning process that reinforces rather than sheers community and that respects rather than demeans academic work and all its producers..

## Definition of Plagiarism

"Plagiarize" derives from a Latin root meaning "to kidnap." Gibaldi (1998), writing under the institutional auspices of the Modern Language Association, defines plagiarism as using another person's ideas or expressions without acknowledging the source, and he asserts that plagiarism gives the impression that something written or thought was original when it was actually borrowed from someone else. It is a form of dishonesty that misrepresents intellectual property and that deprives the creator of intellectual property due recognition. In academia, it is an unpardonable sin. Examples of plagiarism include failure to give appropriate acknowledgement when using another's words, paraphrasing another's argument, and presenting another's line of thinking (Gibaldi, 1998). The Association for Computing Machinery defines plagiarism as "the verbatim copying, near-verbatim copying, or purposely paraphrasing portions of another author's paper" (Boisvert and Irwin, 2006). Although most commonly associated with written productions, such as essays, term papers, theses, and dissertations, in a natural language, plagiarism also occurs in symbolic languages that represent computer programs and mathematical discourse. The Association for Computing Machinery includes the copying of computer codes in its enumeration of offenses against intellectual property (ACM, 2006). Plagiarism can also occur in the graphical presentation of data.

Although the definitions and descriptions of plagiarism have a surface clarity and straightforwardness, making a determination regarding plagiarism poses problems. For example, determining that someone's line of

thinking has been appropriated is problematic when a particular subject, such as abortion or capital punishment, affords relatively few approaches, lines of reasoning, and conclusions. The narrowness of the field of discourse can be extreme when conditioned by the relatively narrow rhetorical and experiential resources available to college students. Furthermore, the sheer volume of writing on some issues makes the duplication of language a statistical rather than an ethical issue.

**Pervasiveness**

Research spanning three decades demonstrates that plagiarism is prevalent at colleges and universities and that the best deterrent is a campus culture that does not accept academic dishonesty. Bowers (1964) surveyed 5,000 students at ninety-nine colleges. His two major findings were that three-fourths of students admitted to practicing some form of academic misconduct, including plagiarism, and that a strong deterrent to such misconduct was the disapproval of peers. McCabe and Trevino (1993) surveyed more than 6,000 students at thirty-one colleges and universities. Their findings were remarkably similar to those of Bowers. They found that two-thirds of students engaged in at least one form of academic dishonesty. In comparing the prevalence of academic dishonesty at institutions with and without honor codes, they concluded that institutions with honor codes had lower rates of such dishonesty. Brimble and Stevenson-Clarke (2005) complemented the above research, which was based on studies in the United States, with research in Australia. They found that 6% of the students surveyed admitted to paying someone else to complete an assignment, 47% paraphrased without acknowledgement, and 40% quoted directly without attribution. Other major findings were that academic staff underestimated the prevalence of academic dishonesty among students, that students viewed academic dishonesty less seriously than academic staff, and that students had little fear of being caught. Sheard et al (2002), in a study of 287 information technology students at two universities in Australia, demonstrated that plagiarism is also an issue in scientific and technological disciplines. They found that, at one university, 85% of the students and , at the other, 69% of the students admitted to some form of academic dishonesty. Copying material from the Internet for an essay ranged from 19% to 22% (Sheard at al, 2002). In a text-based, empirical study of plagiarism among students in an introductory course in programming, Day and Horgan (2005) found that 50% of the students engaged in dishonest behavior by either receiving or supplying tainted work.

**Motivations for Employing Plagiarism Detection Software**

The corrosive effect of plagiarism on educational communities and its pervasiveness obligate faculty to deter the practice and to detect it when deterrence fails. Deterrence can include educating students about the inimical effects of plagiarism on education and about the consequences of its detection, which include institutional and legal sanctions. A key component in deterrence is the expectation on the part of students that instances of plagiarism will be detected and prosecuted.

In recent years, two factors, one social and the other technological, have severely compromised the ability of faculty to detect plagiarism successfully enough to convince students that instances of the practice will be discovered. The social factor is that of large class sizes and an increase in writing and other assignments that results from writing across the curriculum and assessment initiatives. The volume of student writing reduces the amount of time that a faculty member can devote to reviewing student submissions. The technological element is the Internet. It has vastly expanded the availability of potential sources for plagiarism, eased the incorporation of plagiarized material into academic productions, and has thus made the detection of plagiarism burdensome. The traditional countermeasure to the threat of plagiarism, a lone faculty member evaluating academic productions with an eye toward detecting misrepresentation, is rendered impotent by these forces.

These factors have created a need for some means of efficiently scanning large amounts of text for possible instances of plagiarism and for identifying such instances objectively and incontrovertibly. Plagiarism detection software attempts to meet that need.

**Plagiarism Detection Software for Productions in Natural Languages**

The types of algorithms that practitioners predominantly deploy to detect plagiarism in natural language productions divide into two broad categories—corpus based and interrogatory. The two categories reflect different approaches and philosophies. The corpus-based approach encourages the global submission of documents that respond to writing assignments. Most typically, faculty, as a course requirement, have students submit their work to a plagiarism detection service just prior to the submission deadline. Faculty then examine the works that have been identified as meeting criteria for academic dishonesty, and those that are not found to be false positives are further investigated. This approach is exemplified by universality of application and by the placing of computer analysis before human scrutiny. The interrogative approach is based on the assumption that plagiarism is revealed when a student is demonstrably unfamiliar with a suspected submission. A standard implementation has a two-tier form. Faculty read student submissions and, in the course of that reading, identify suspect documents. Under the supervision of the faculty member, software removes words from the suspected document, replaces them with blanks, and prompts the student to fill in the blanks. The program determines familiarity from the speed and accuracy of the responses. This approach inverts the corpus-based approach. It is selective rather than comprehensive. It gives priority to human judgment. It situates plagiarism in human conduct rather than in impersonal pattern finding.

At the highest level of abstraction, corpus-based plagiarism detection software takes as input a suspect document and an archived corpus of authenticated documents, compares the suspect document to the corpus, and outputs passages that the suspect document shares with the corpus and a measure of the likelihood that the author plagiarized material. At a more operational level, a corpus-based protocol is implemented in several ways. TurnItIn.com, the most high profile company in the field, employs document source analysis to generate digital fingerprints of documents, those submitted for authentication, those in the archive, and those available through ProQuest, and it complements the search of its in-house archive with searches of the World Wide Web (iParadigms, 2007). It must be noted that documents present neither in ProQuest nor on the Web are beyond the reach of TurnItIn.com The Essay Verification Engine, which employs the World Wide Web as its corpus, uses techniques to target exhaustively the sites that most likely are sources of plagiarized material and compares their content with the suspect work (CANexus, 2007).

Glatt Plagiarism Services exemplifies an interrogative approach to plagiarism detection. Faculty have students submit a work suspected of being plagiarized to the Glatt Plagiarism Screening Program, which is free standing, non-Web-based software . The program replaces every fifth word of the suspected paper with a standard size blank, and the student is then prompted to supply the missing words (Glatt, 2007). The number of correct responses, the amount of time intervening between responses, and various other factors are considered in calculating a plagiarism probability index.

Although the programs mentioned above illustrate different approaches to plagiarism detection, they share a common characteristic. They are deployed by commercial, proprietary organizations that guard their intellectual property and that keep private any quality control studies that they conduct. These are attributes that clash with the free inquiry and the openness associated with institutions of higher education. Faculty members who employ these programs do not know in detail how they work, nor do they have any means of judging their efficacy.

Niezgoda and Way (2006) provide a partial corrective for these deficits in their description of SNITCH, Spotting and Neutralizing Internet Cheaters. As the name suggests, SNITCH uses Web pages on the Internet as its corpus, and Niezgoda and Way, in providing some insight into the program, implicitly provide insight into other programs in its category. The authors provide a detailed exposition of the corpus-based algorithm that they use to detect plagiarism in scientific and technological writing, an arena of discourse that poses particular problems for plagiarism detection, and they provide data regarding the effectiveness of their algorithm as measured against an oracle and against a competitor, the Essay Verification Engine, Eve 2.

SNITCH employs a sliding window technique that determines the average length of words within a succession of windows (Niezgoda and Way, 2006). SNITCH reads a specified number of words, determines the average number of letters per word, and associates that average with the current window. The procedure is repeated, moving the windows forward one word with each iteration until the end of the document is reached.. Windows are ranked from highest weights to lowest, and the top ranked windows are then submitted to search engines to detect matches. SNITCH's success rate for detecting plagiarism in actual student submissions range from 40% for papers with minimal plagiarism to 63% for papers with a high level of plagiarism, and it produced no false positives (Niezgoda and Way, 2006). SNITCH outperformed Eve 2, a commercial program, in detecting plagiarism in submissions with low actual plagiarism, 40% compared to 12%, and its performance against submissions with a high level of plagiarism, 63%, was almost indistinguishable from that of Eve 2 at 63%. In terms of revealing what the program does and how well it does it, Niezada and Way provide a model approach, one that contrasts with the secrecy of commercial software.

**Plagiarism Detection Software for Productions in Symbolic Languages**

Although plagiarism detection within natural language productions is the most widespread application of such software, work has been done regarding student-produced computer programs. Prechelt, Malpohl, and Philippsen (2002) employed a pair-wise approach to detecting plagiarism in a corpus comprising programming assignments. Their software takes as input a set of programs, compares them pair wise, and outputs a set of HTML pages that allow for exploring and understanding the similarities found. (Prechelt, Malpohl, and Philippsen, 2002). Arwin and Tahaghogh (2006) extend this work by checking for plagiarism in submissions written in different programming languages. In addition, Day and Horgan (2005) employed an artifact-based protocol to detect plagiarism and to trace the flow of plagiarized material by distinguishing between suppliers and recipients. The researchers used digital watermarks that were implanted on student work at the time of electronic submission.

Given the limited range of programming assignments and given the narrowness of the expressive resources of programming languages, it would seem that there would be a high incidence of false positives in detecting plagiarism. This is not the case. Prechelt, Malpohl, and Philippsen (2002), for example, report that, when probing datasets that they do not classify as hard, their software identified all the plagiarized examples and produced no false positives. With datasets described as hard, they identified 66% of the plagiarized programs and produced some false positives. Given that digital watermarks are nearly incontrovertible and highly resistant to alteration (Day and Horgan), false positives and false negatives are not present when they are employed..

**Technical Issues Regarding Plagiarism Detection Software**

An effective evaluation of software can occur only if two questions can be answered: "What does the software do?" and "How well does it do it?" These questions are especially important when the software addresses a social issue and is applied to human subjects. As mentioned above, the commercial products treated here respond to these questions with general, non-empirical answers. Although software companies understandably are concerned about the dilution of their intellectual property that could attend the disclosure of their algorithms, historical evidence, such as the comparison above of SNITCH and Eve 2, suggests that open disclosure and analysis improves the efficacy and social acceptance of software. The question regarding efficacy is particularly crucial in the context of plagiarism detection. Efficacy includes countermeasures used to thwart plagiarism detection. Faculty and students should know details regarding the rate of false positives and false negatives, and they should have confidence that companies are striving to reduce both. See "Sample Empirical Test" below for a study that addresses some of these issues.

Plagiarism detection software that uses the Internet for its corpus is subject to effective countermeasures. One is that Web sites associated with the sale of term papers and computer code sequester their products in locations

that are not openly connected to the World Wide Web. Material acquired through these sites are likely to escape detection. In addition, Web sites can deploy software that repels Web crawlers such as those used by TurnItIn.

Another technical issue is that of false positives and false negatives. Corpus-based programs, such as TurnItIt.com, do an excellent job of finding matches between student submissions and items in their database. These programs, however, do not distinguish between matches that are properly cited and those that are not, so a high index of plagiarism does not necessarily mean that plagiarism has occurred. For example, a quotation that is properly demarcated with quotation marks and properly cited will contribute to a high plagiarism index. The proliferation of such falsely marked passages creates textual noise that faculty have to ignore. If that noise is sufficiently dense within a particular paper or particularly prevalent among a number of papers, faculty can easily ignore true positives. Wading through such noise adds to the burdensome of reading and evaluating student work. Also, final versions of papers can be flagged because they match earlier, draft submissions (Florida State University, 2007). Even when students demonstrate to faculty that their productions have been erroneously accused of being plagiarized, the doubt cast on the work can linger in the faculty member's mind and impact grading decisions. Students can legitimately feel harmed by such false implications. False negatives also occur. Material that appears on the World Wide Web for a transient period will not be found by Web crawlers, and students can sufficiently alter borrowed material to mask plagiarism (Florida State University, 2007).

Corpus-based plagiarism detection software suffers from the inability to distinguish between documents that are the source of plagiarism and those that are the recipients of plagiarized material. Although time stamping the documents might provide some power in this regard, there is no certainty that documents with later time stamps were the recipients of material with earlier time stamps. A watermark based plagiarism detection protocol (Day and Horgan, 2005), which is discussed above, addresses the issue.

**Ethical and Legal Issues**

Plagiarism detection based, at least in part, on the maintenance of a corpus of documents submitted by students raises a technical issue with ethical and legal implications. Although commercial enterprises, such as TurnItIn.com, assure the security of personal data, the possibility for abuse exists. Student submissions reside in a database that provides no context for those submissions. Documents created in response to hypothetical situations or to satisfy purposefully provocative assignments can subject the authors to public embarrassment when discovered long after their submission. For example, a paper defending polygamy, written as an exercise in advocating an unpopular position, can surface during an election campaign or be reported by the press. There is also the potential for the database to be scoured by government agencies or by journalists searching for any sign, no matter how slight, of criminal or socially unacceptable conduct. The possibility that communications intended for or created by a faculty member in a specific academic setting can be read, evaluated, and published in a more public forum threatens academic freedom and unfettered inquiry.

The practice of requiring students to submit their work to a for-profit enterprise raises a number of concerns. The most troubling is that of supporting a commercial enterprise without compensation. Companies that maintain a database of student submissions generate their revenues based on those submissions, but the students receive no compensation in return for the labor of submitting material nor for their surrender of their intellectual property rights. Moreover, colleges and universities pay a fee to companies for their participation in a commercial enterprise that exploits the work of faculty and students alike. Educational institutions and faculty become complicit in a regime of electronic slavery. Although TurnItIn.com maintains that its practices violate neither intellectual property rights nor United States laws, such as the Family Educational Rights Privacy Act, regarding privacy of student records, attorneys specializing in this area disagree (UCLA, 2007). Litigation initiated by high school students in the U. S. District Court in Alexandria, Virginia, may resolve some of the issues related to copyright infringement (Robelean, 2007). Moreover, requiring the global submission of productions to commercial sites can damage an academic community and distort learning. The practice creates a hostile, adversarial atmosphere where

students can readily feel that they are presumptively suspect. In addition, they are taught the ironic lesson that their authorial integrity is violated in order to teach them the value of authorial integrity (UCLA, 2007), and there is the additional lesson that corporate interests supersede their own. Furthermore, the practice of compelling universal submission promotes a culture of surveillance and mistrust, and it runs counter to an ethos of collegiality and mutual respect.

**Sample Empirical Test**

On November 8, 2007, the author submitted four works with known levels of plagiarism to TurnItIn.com using an account funded by Iona College. The results reveal some of the weaknesses of corpus-based plagiarism detection software and illustrate the need for a comprehensive empirical study.

The four submissions were a paper on capital punishment obtained from a freely available Web page (eCheat.com, 2007), that same paper doctored to camouflage plagiarism, a short essay on capital punishment written by the author, and a draft of this article. TurnItIn.com reported an overall similarity index of 57%. The software flagged twenty passages in the approximately 1,600-word document that matched material associated with submissions at multiple educational institutions and Web sites. The software did not detect that the entire paper originated from a readily available Web site. Also, the software flagged passages of utter blandness ("Capital punishment is a very divisive topic in the United States" is an example) that are highly likely to recirculate in responses to such a generic topic as capital punishment. That same paper was then doctored using a technique discussed in UCLA (2007). MSWord's find and replace feature was used to replace every occurrence of the letter "e" with the string "e~." That document received a similarity rating of 2%. The submission on capital punishment written by the author using vacuous language without originality of thought received a similarity index of 0%. An earlier draft of this paper received a similarity index of 12%. Properly cited passages were flagged, but the match was often identified not with the actual source but with another submission, usually one with a collegiate association. Some matches that might trouble a third-party led to dead ends. For example, the passage "with a standard size blank, and the student is then prompted to supply the missing words (Glatt, 2007). The number of correct responses, the amount of time intervening between responses, and various other factors are considered in calculating a plagiarism probability" was marked as marching a student paper submitted at Chapman University. To see the submitted document to determine context requires that the evaluator make a request to the professor to whom it was submitted. There is no assurance that a response will come in a timely manner. The burdensomeness of the procedure outweighs the utility of discovering the match

**Conclusions and Recommendations**

The conclusions and recommendations follow from the social and technical threads that organize this article. Ethical conclusions and recommendations precede technical recommendations.

Plagiarism, its causes, and its consequences are core academic concerns, and those concerns are magnified as the extent of plagiarism becomes known and as it increases. Plagiarism detection software is a tool that can both deter and detect plagiarism, but faculty need to employ that tool without abandoning values and procedures essential to academic discourse. To this end, faculty should use plagiarism detection software as a secondary rather than as a primary means of defense. The software is best applied to work that has been identified as possibly plagiarized by faculty analysis and judgment. In addition, faculty should not coerce students into becoming unpaid employees of commercial enterprises. Since little or no case law exists in the United States regarding the copyright and confidentiality concerns that arise from using plagiarism detection software, students, parents, and advocacy groups should aggressively challenge current practices, especially those adopted by institutions of higher learning.

To complement the above ethical and legal concerns, there are some technical recommendations that derive from the research presented here. One is that commercial enterprises should make public the algorithms used in

their software so that they can be improved. Making public the algorithms also permits third parties the opportunity to evaluate how well the software protects intellectual property and confidentiality. Another recommendation is that large scale empirical studies be conducted to measure the percentage of false positives and false negatives and to determine the burdensomeness of the system.

**Acknowledgements**

The author gratefully acknowledges the support, both monetary and spiritual, that Mr. Jack Rudin and the May & Samuel Rudin Family Foundation have extended to the author, the Samuel Rudin Academic Resource Center, and Iona College. The author also thanks the students of Computers, Technology, and Society, Fall 2007, for their encouragement and their reactions to drafts of this article. The author especially thanks. Ms. Elizabeth Cassara for sharing with me her experiences and insights regarding plagiarism detection software.

**References**

1. Arwin, C. and S. M. M. Tahaghogh (2006). Plagiarism Detection Across Programming Languages. ACM International Conference Proceeding Series, vol. 171. Proceedings of the 29th Australasian Computer Science Conference, Hobart, Australia, vol. 48 ,pp. 277 – 286.

2. Association for Computing Machinery (2006). ACM Policy and Procedures on Plagiarism. Available at http://www.acm.org/pubs/plagiarism%20policy.html. Accessed November 6, 2007.

3. Boisvert, R. F. and M. J. Irwin. (2006). ACM Policy on Plagiarism: Plagiarism On the Rise. Communications of the ACM, vol. 49, n. 6, pp. 23 – 24.

4. Bowers, W. J. (1964) Student Dishonesty and Its Control in College. New York: Bureau of Applied Social Research, Columbia University.

5. Brimble, M. and P. Stevenson-Clarke (2005). Perceptions of the Prevalence and Seriousness of Academic Dishonesty in Australian Universities. *Australian Educational Researcher*, v32 n3 p19-44 Dec 2005. (EJ743503)

6. CANexus (2007). Eve 2: Essay Verification Engine. http://www.canexus.com/eve/index.shtml. Accessed November 5, 2007.

7. Day, C. and J. Horgan (2005). Patterns of Plagiarism. Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education. Pp. 383 – 387.

8. eCheat.com (2007). Capital Punishment Persuasive Essay. http://www.echeat.com/essay.php?t=33395. Accessed November 7, 2007.

9. Florida State University Center for Teaching anf Learning (2007). FSU's Information for Turnitin.com. Available at http://learningforlife.fsu.edu/ctl/explore/bestPractices/docs/turnitin.pdf. Accessed November 7, 2007.

10. Gibaldi, J. (1998) MLA *Style Manual and Guide to Scholarly Publishing*. 2nd ed. New York: MLA.

11. Glatt Plagiarism Services (2007). http://plagiarism.com/. Accessed November 5, 2007.

12. iParadigms (2007). http://www.plagiarism.org/technology.html. Accessed November 5, 2007.

13. McCabe, D. and L. Trevino (1993). Academic Dishonesty: Honor Codes and Other Contextual Influences. *Journal of Higher Education*, (64) n5 p522-38 Sep-Oct 1993.

14. Niezgoda, S. and T. Way. (2006) SNITCH: A Software Tool for detecting Cut and Paste Plagiarism. Proceedings of the 37th Special Interest Group on Computer Science Education. Pp. 51 – 55. New York: Association for Computing Machinery.

15. Pennsylvania State University Senate Committee on Computing and Information Systems (2006). Turnitin: A Tool to Assess Student Plagiarism. Available at http://www.psu.edu/dept/cew/TurnitinFinalReportFS.doc. Accessed November 6, 2007.

16. Prechelt, L., Malpohl, G., Philippsen M., Finding (2002). Plagiarisms Among a Set of Programs with JPlag. *Journal of Universal Computer Science*. Vol. 8, issue 11, pp. 1016 – 1038.

17. Robelean, E. (2007). Online Anti-Plagiarism Service Sets Off Court Fight. *Education Week*, May 9, 2007. vol 26, issue. 36.

18. Sheard, J. M. Dick, S. Markham, I. Macdonald, and M. Walsh (2002). Cheating and Plagiarism: Perceptions and Practices of First Year IT Students. ACM SIGCSE Bulletin , Proceedings of the 7th annual conference on Innovation and Technology in Computer Science Education ITiCSE '02**,** Volume 34 Issue 3, pp. 183- 187.

19. UCLA Office of Instructional Development (2007). Exploring the Controversy Surrounding the Use of TurnItIn. http://www.oid.ucla.edu/training/trainingarticles/turnitin/turnitin-2. Accessed November 6, 2007.