Jl. of Interactive Learning Research (2008) **19***(3)*, 489-508

Design, Implementation and Deployment of PAIRwise

ALLAN KNIGHT AND KEVIN ALMEROTH
University of California-Santa Barbara, USA
aknight@cs.ucsb.edu
almeroth@cs.ucsb.edu

BRUCE BIMBER
University of California-Santa Barbara, USA
bimber@cits.ucsb.edu

Increased access to the Internet has dramatically increased the sources from which students can deliberately or accidentally copy information. This article discusses our motivation to design, implement, and deploy an Internet based plagiarism detection system, called PAIRwise, to address this growing problem. We give details as to how we detect plagiarism and the various development phases we went through before releasing PAIRwise as an open source software tool. Our major conclusion is that it is possible to deploy an automated plagiarism system that is an effective, customizable, and affordable tool for investigating how plagiarism affects academic integrity.

INTRODUCTION

Plagiarism has long been a problem in academia. Compounding this problem more recently is the creation of the Internet, which has greatly expanded the sources from which to plagiarize. Furthermore, a high percentage of certain populations currently use the Internet. For example, Nielsen estimates that 68.6% (Nielsen NetRatings, 2006) of the U.S. population uses the Internet. The danger with the growth of these sources for plagiarism is the destruction of academic integrity. Whereas students of the past only had access to information in libraries and from fellow students, today's students have access to an overabundance of sources: their library, their fellow students, and thousands of libraries and students around the world. And,

not only is there more sources for students to plagiarize from, but students can search for these sources instantaneously. This increasing problem should be leading academic institutions to employ all means necessary to prevent and detect plagiarism. Many institutions, however, have not employed any such plagiarism detection means. Many instructors, and even institutions, are ignoring the problem in the hope that it will disappear. With a new generation of students that are Internet savvy, and who may not even understand that copying the work of others is plagiarism, ignoring the problem can only have a negative impact on everyone's perceptions of academic integrity.

The problem is that traditional manual techniques for plagiarism detection are inadequate, especially given the Internet. Manual identification of plagiarism is time-consuming, inefficient and easily circumvented. In order for instructors or graders to manually identify instances of plagiarism, they are not only required to read every assignment, but must also know all the sources of potential plagiarism. Furthermore, this process takes a great deal of time. Lastly, changing the tense or wording of a copied piece of text often easily circumvents manual plagiarism detection techniques. While deterrence, vigilance, and assignment creation guidelines can reduce the problem of plagiarism, automated techniques provide the best means to make plagiarism detection techniques efficient and harder to circumvent.

To meet the need for automated plagiarism detection, Turnitin.com, MyDropBox, and others have created commercial ventures. These ventures, however, introduce further problems that are not easily overcome. The first problem is expense. Both Turnitin.com and MyDropBox are for-profit organizations and charge considerable fees for their services. The deployment of either system, at any level, from the individual instructor to an entire institution, is quite expensive. This cost can range from $100 per quarter per class, to tens of thousands of dollars per department per academic year. While these systems may be affordable for some institutions, they are not affordable for all. A further problem is the legal issues associated with using such services. There have been several successful court cases concerning the privacy issues associated with using for-profit plagiarism detections services. The result is that students at these institutions are no longer being required to submit assignments to commercial sites. Besides privacy issues, there can be copyright issues as well. Submitting an assignment to one of these sites may transfer copyright ownership to the site, which, in general, is not desirable for students or their institutions. One final issue with these systems is their lack of visualization techniques for comparing student assignments and potential plagiarism sources. Currently, these systems report scores for each assignment as to its overall uniqueness. These reporting schemes also do no address situations where groups of students plagiarize from each other. For example, a group of more than two students might share a significant amount of work.

The goal of the PAIRwise plagiarism detection system is to address all of these issues. First, PAIRwise is open source. This fact means that, not only is PAIRwise affordable, since the source code is available for free, it is also fully customizable. Furthermore, because PAIRwise can be easily deployed for any size of user population by a university, the issues of privacy and copyright ownership do not exist. Finally, PAIRwise also offers a novel visualization module that allows users, not only to see how much of each student's assignment is copied from another source, but also relationships between multiple sources by displaying all the assignment pairs that have verbatim matches above a configurable threshold. This unique feature allows for the identification of groups of students that copy from each other.

The key goals of PAIRwise are to be both affordable and effective. This article illustrates that PAIRwise meets both of these goals. First, as a freely licensed, open source project, PAIRwise is affordable. The only barrier to installing PAIRwise is having a computer with a supported operating system to install it on and the personnel with the necessary computer experience. Second, PAIRwise is demonstrably capable of detecting plagiarism. The evaluation section in this article enumerates examples of PAIRwise identifying potential plagiarism from other students as well as the Internet in students' assignments from real courses at the University of California, Santa Barbara.

The research in this article shows that the possibility exists to effectively and affordably deploy a plagiarism detection system that scales from an individual instructor to a single institution and perhaps beyond. Furthermore, this research contributes to the scientific evaluation of such systems, something that is lacking for commercial services. Finally, PAIRwise, and its related research, facilitates the study of the social impact of both plagiarism detection systems and plagiarism itself. Our hope is that PAIRwise is the beginning of a dialog within institutions and between institutions on the impact of plagiarism on education and the integrity of academia in general.

This article is organized as follows. The second section discusses other research related to PAIRwise. The third section outlines the design, implementation, and deployment of PAIRwise. Next, the fourth section discusses our methodology to evaluate PAIRwise and the results. The last section offers concluding remarks that outline the accomplishments of our research associated with PAIRwise.

RELATED WORK

The work related to our research deals with both the impact of plagiarism and systems for detecting plagiarism. In the following, we discuss this research and discuss how PAIRwise extends the research in this area.

We first discuss the works of others with regard to the concept of plagiarism since the goal of our research is to investigate the impact of plagiarism on aca-

demic integrity. There are two main points in this body of research: quantifying plagiarism and deterring plagiarism. While there is no possible way to put an upper bound on the amount of plagiarism occurring, there have been several attempts to quantify the lower bound. Most notably, the most recent work by McCabe (2002) found that 72% of high school students admitted to having committed at least one instance of serious cheating and 15% have submitted work copied from electronic sources. Others, such as Braumoeller (2001), used electronic means to find that 1 out of 8 students improperly used material from other sources. While these studies surely suggest a lower bound for how prevalent the problem is, they do not give an upper bound.

Even before the possibility of electronically searching for cases of plagiarism, the academic community has sought to inform educators about techniques to avoid plagiarism. The Joint Information Systems Committee (JISC) wrote a guide (Carroll, 2001) entirely devoted to the issue of plagiarism. This guide covers not only how to "design out" plagiarism in assignments, but also gives guidance on how to prosecute offenders and engage students as partners in fighting plagiarism. No mention is made, however, on how to detect plagiarism on a large scale.

The role of PAIRwise is to help quantify plagiarism, guiding educators and deterring plagiarism from the start. By giving educators a tool for reliably detecting Internet plagiarism, as well as plagiarism of other students' assignments, PAIRwise looks to contribute to the problem of how best to protect and maintain integrity in academics.

As for research and projects related to plagiarism detection systems, two Biophysicists at the University of California, Berkeley, originally started the work most related to our project. Their work eventually turned into a private venture: Turnitin.com, formerly known as plagiarism.org (Turnitin.com, 2003). While they seek to accomplish many of the same goals, their approach to the problem is very different from ours. Turnitin.com has its own servers with terabytes of information, collected from crawling the Web, subscribing to term paper mill sites, and collecting past assignments submitted to their services. Entire assignments are then compared against this database of sources.

Where we differ is in the philosophy of how to do the searches. While we both seek to find sources on the Internet, we feel their process is heavyweight. They can catch every single copied sentence but at a significant time cost as they search each sentence against all the data they have collected. We instead look to intelligently select sentences that are likely candidates and flag an assignment as suspicious if those sentences are found in sources on the Internet. And then, like them, let the instructor determine if the student has acted improperly in borrowing from a source. Our framework, like that of Turnitin.com, can easily be connected to any database of text, whether that database is the Internet or a collection of past assignments.

One issue of concern for Turnitin.com is the legality of transferring a copy of each student's assignment to an outside organization. While some argue that transferring the copyright ownership violates students' rights (Paulson, 2002), Turnitin.com reassures institutions this is not an issue since it falls into the category of fair use. However, we still believe that (a) this issue has not been properly resolved, and (b) the use of Turnitin.com may also violate students' privacy. Employing an internal system has the benefit of avoiding these privacy and legal issues.

Another body of related work is CopyFind (Plagiarism Resource Site, 2003). The algorithm and measure of uniqueness used by PAIRwise are very similar and CopyFind has had a great deal of influence on the creation of PAIRwise. While CopyFind's goal is also to detect plagiarism, its search domain is more limited. Rather than finding borrowed text from the Internet, CopyFind compares a collection of student assignments to each other. Assignments with enough similarity are flagged for further inspection. CopyFind, however, has no ability to determine if sources are copied from the Internet. Finally, there is a simple way to circumvent CopyFind. By using the "Track Changes" feature of Microsoft Word, the possibility of adding noise to the saved copy of a document is possible. Therefore, what the user sees is the final version of a document, but the binary representation, however, is significantly different and may fool the CopyFind system into giving a high uniqueness score for exactly the same content. PAIRwise avoids this problem by comparing the onscreen content rather than processing the binary data contained in document files.

Other related work deals with collecting digital media. SCAM (Shivykumar, 1995) and CHECK (Si, Leong, Lau, & Rynson, 1997) find similarities among documents in a common database. While their main focus is to find similar documents in a file system or other database of digital media, they differ from our work in one major respect. They look at similarities of documents as a whole, not at individual sentences. These systems do not flag documents that have less than 25% similarity. We are working in a context where overall similarity is not necessarily important because some plagiarism may involve only copying a few sentences. If we find that the average student assignment is 100 sentences long and we want to flag all assignments with at least five possibly copied sentences, then the threshold for similarity is only about 5%. Furthermore, these systems tend not to take any contextual similarities into consideration. This limitation makes these systems easy to defeat by merely changing words throughout a copied document.

Still another body of work concentrates on finding plagiarism within computer programming assignments. MOSS (Boyer and Hall, 1999) and Sherlock (Joy and Luck, 1999) are two examples. However, these solutions are similar to CopyFind in that they only look for duplicates among a single set of assignments and do not have the ability to look at external data

sources. Another well know system for detecting plagiarism in programming assignments is SID (Chen 2002). While it aspires to the same goal, it is quite different. SID uses a measure of similarity based on Kolmogorov complexity, which is universal. The truly unique feature of SID is the fact that because the Kolmogorov complexity is universal, SID, in theory, is not cheatable. While SID exclusively finds similarities in code, an altered version could possibly detect similarities in plain text.

PAIR

The origin of PAIRwise derives from a research effort called Paper Authorship Integrity Research (PAIR). The PAIR effort is part of the on going efforts of the Center for Information Technology and Society (CITS) at the University of California, Santa Barbara. The goals of the PAIR project are twofold. The first goal is to explore the impact of the Internet on plagiarism in the classroom. The second goal is to understand the impact of an automated plagiarism system when used in the classroom. For example, does checking assignments for plagiarism with an automated system actually reduce the amount of plagiarism, especially if students are aware of these efforts? And, if they do know about these efforts, what are their feelings about the use of an automated plagiarism detection system?

PAIR is an interdisciplinary effort involving scientists from several diverse fields of research as well as university administrators. These research fields include computer science and political science. Together, the researchers combine their expertise for creating an automated system with their expertise for measuring the social implications of such a system. The University of California Santa Barbara's Office of Instructional Consultation supplied computers and resources for both the collection and processing of student assignments in support of this research effort.

Before beginning this research, we knew we needed a system for automated comparison of student assignments to each other, to past instances of classes, and to the Internet. While commercial services already existed to fulfill this need, using these systems, as previously mentioned, was problematic. And while non-commercial research and open-source systems also existed, these systems did not provide all the functionality needed. The lack of a viable system ready-to-deploy within the university led to an effort by the PAIR team to design, implement, and deploy its own system. This new system came to be known as PAIRwise.

PAIRwise

PAIRwise is the realization of a system that compares multiple text documents to each other to determine their similarity. The design and develop-

ment required two phases. The first phase sought to identify techniques to collect documents from the Internet. This phase designed and implemented a rudimentary system to download files found using Internet search engines. These files were compared to student assignments to determine if the assignment contained improper source material. The second phase took the lessons learned from the first and designed, implemented, and deployed a complete system for automated plagiarism detection. This final system, which came to be known as PAIRwise, compares assignments to those found on the Internet, those in the same class, and any other set of archived assignments from past classes. A further description of these two phases now follows.

Phase 1: Preliminary Research

The preliminary phase of this research focused on how to use Internet search engines, such as Google (2003), to find possible instances of plagiarism within the text of students' assignments. There exists a multitude of techniques to determine (a) whether any part of a student's work is plagiarized, and (b) which parts of that work are plagiarized. The question is how to determine which passages of a student's work should be used to create a search string for and Internet search engine. While an exhaustive search would catch the most cheaters, we believed that using a more selective process could significantly reduce the number of searches, and yet, identify all of the instances of plagiarism that an exhaustive search would find. The non-exhaustive techniques that we considered range from using every string of n words in a document, to randomly selecting parts of single sentences. Below are the different techniques we developed and evaluated.

Exhaustive Searching

Exhaustive searching has the potential to be the most effective way to detect possible instances of plagiarism. There can be, however, several different interpretations of what "exhaustive" means. One interpretation would be to search for every combination of words in the assignment. This technique, though, would result in so many queries to Google, that processing a single assignment could take several hours. When applied to 400 or more student assignments, this interpretation of "exhaustive" does not seem viable. Instead, we interpreted "exhaustive" to mean each sentence. This interpretation would break every assignment into its individual sentences and submit the first eight words, and thus greatly reduce the number of required searches. The determination of "the first eight words" is based on our anecdotal observations that this number results in the search engine returning fewer false positives.

The choice of how exhaustive to search affects the number of queries required. For example, consider exhaustively searching approximately 480 assignments from a lower division college course in Political Science. If

each assignment has approximately 10 pages, using a truly exhaustive search would require approximately 1,400,000 searches. If, instead, the system searches using each sentence, the number of searches is only about 55,000; however, the system can still do better. The next step, then, is to look at more intelligent techniques to try and reduce the number of search queries even further. Given that there are quite a number of different possibilities, the challenge is to choose the one that is effective but does not require a very large number of searches.

Using Surface Linguistic Features

Linguists have developed many surface linguistic features over the years (Fry, 1977), (Gunning, 1952), (McLaughlin, 1969). These features usually involve calculations based on the number of words per sentence and distribution of syllables in each word. While these linguistic features give no weight to the context of the sentence, research has shown they are effective in determining the readability of a given work.

A common manual technique used to detect plagiarism is to look for changes in writing style. The question is how to automate the identification of changes in writing style? Our hypothesis is that readability scores can identify these changes in style. By determining the readability score for each sentence and then choosing the scores below a given threshold, the system should be able to identify a similar number of plagiarism instances as the exhaustive techniques do. While intuition suggests that the sentences with the higher-grade level scores seem to be the most likely candidates, this, in fact, is often not the case. Grade level scores give some indication as to the readability and complexity of the text. Previously published writings tend to be more readable – because of the editing process – than those created by students. We believe, therefore, that the most likely candidates for plagiarism are not at the high end of the grade level score, but at the low end.

To compute the surface linguistic feature, we chose the Flesch-Kincaid Grade Level Score. Figure 1 gives the formula. For each sentence a score is

(.39 x ASL) + (11.8 x ASW) – 15.59

where:

ASL = average sentence length (the number of words divided by the number of sentences)

ASW = average number of syllables per word (the number of syllables divided by the number of words)

Figure 1. Flesch-Kincaid Grade Level Score (Microsoft, 2003)

calculated. All sentences with a grade level below 10 are sent to an Internet search engine to see if they exist somewhere in the Internet. All results are saved and later analyzed to determine if any sentences were plagiarized.

Phase 2: Final Realization of PAIRwise

Based on the lessons learned from Phase 1, we set about to design an entire system for plagiarism detection. The new system was designed to be a self-contained, open-source project that is installable and maintainable by any system administrator. Furthermore, this phase incorporated all of the techniques explored in the first phase to find potential Internet plagiarism sources. In the following sections, we outline details of the PAIRwise design.

PAIRwise Design

The PAIRwise plagiarism detection system architecture consists of four modules. This modular approach is important as it separates the tasks of plagiarism detection in such a way that each module is replaceable with a new implementation without modifying the other modules.

The first module, called the *Request Module*, allows a user to request comparisons of student assignments. The second module, called the *Conversion Module*, then converts these documents into a single format used by PAIRwise for document comparison. The format used is plain ASCII text. This conversion allows for easy comparison and visualization of similarities between compared documents. The third module, called the *Comparison Module*, actually compares each assignment to others within a corpus. This module can then record the similarity score of each pair of documents as well as determine relationships between sets of documents with high similarity scores. The final module is the *Reporting Module*. This module provides an easy-to-use interface that displays the similarity scores recorded by the Comparison Module and allows users to examine student assignments to determine if plagiarism likely occurred.

Request Module

The function of the Request Module is to allow a user to request comparisons of student assignments. These requests come in the form of HTML forms that ask the user for information necessary for PAIRwise to perform the comparisons. First, a title is given. The title allows both PAIRwise and the user to organize the results. Next, the module requests an e-mail address. The module uses the e-mail address to inform the user about the completion and location of the results. The email address is entered twice to allow the module to confirm no mistakes were made. Finally, the module displays to the user information about the location of student assignments used in a comparison. The module can display this information in several different forms based on the type of comparison. Figure 2 shows an example of this interface, which shows a

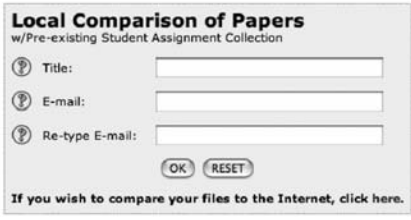


Figure 2. Screen shot of the Request Module

request for a collection of student assignments to be compared against all other students in the same class, as well as all past instances of the class.

One feature not implemented as part of PAIRwise is a document collection module since such systems already exist. For both Internet and class comparisons of assignments, student assignments are expected to already be stored on the server in a directory created by the system administrator. PAIRwise is then configured to find student assignments in a particular directory on the server. The specific manners in which assignments are collected are left to the system administrator and the user. For example, the user can collect the assignments by e-mail, and then create an archive file using a utility such as

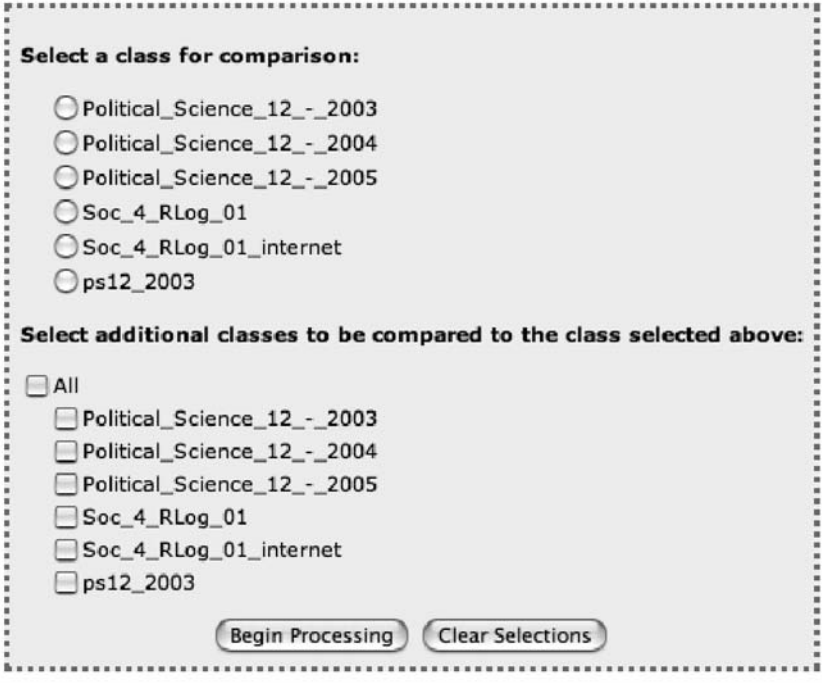


Figure 3. Screen shot of the Request Module, which allows for selection of the primary student assignment set and secondary student assignment sets

Zip. This file can then be submitted to the Request Module. The module will decompress this archive file.

Whether the PAIRwise hosts the collection of assignments directly or accepts them as an archive file, PAIRwise organizes documents based on the underlying file system of the PAIRwise host. By collecting the files in a well-organized directory structure, PAIRwise is able to present the user with a straightforward interface to choose multiple instances of classes, as well as other classes to compare student assignments against. For example, if PAIRwise is configured to find student assignments in a directory called "/documents/pairwise," student assignments can be organized using a subdirectory for each class within this directory.

By organizing assignments in this way, a user can easily choose which set of assignments to compare. Figure 3 shows the interface for selecting assignments to compare. This interface allows the user to select the primary set of student assignments, shown at the top of the figure. Next, the user can select student assignments from other classes at the bottom.

After the assignments are selected and the "Begin Processing" button is pressed, PAIRwise compares the student assignments and sends an e-mail to the user upon completion.

The Request module consists of a series of HTML pages and templates combined with CGI scripts. To maintain the same "look-and-feel" throughout the interface, PAIRwise uses the same HTML templates in the reporting module outlined later in this section.

Conversion Module

Because students use a wide variety of applications to create their assignments, converting these assignments to a common format is necessary. The Conversion Module provides this function. Currently, the Conversion Module is capable of converting most major word processing file formats along with HTML, PDF, PostScript, and plain text. Unfortunately, at the time of development, no libraries existed for converting Microsoft Works® or OpenOffice documents. Future releases of PAIRwise may be able to handle conversion of these file formats. The file format used for comparison is ASCII text because it is easy to process during the comparison phase, and displays easily during the reporting phase.

The conversion module itself consists of a library, scripts, and third party utilities. The process of converting any student assignment into ASCII text starts by determining the file format. A naïve approach to this step is to use the file name extension to determine the file type. This approach, however, fails approximately 10% of the time. A better approach, therefore, is necessary. Fortunately, UNIX provides a utility called "file." The output of this utility is the type of the file along with the name of the possible application that produced it.

Once the proper file format is determined, converting the student assignment to ASCII text is necessary. Again, a naïve approach to this problem is to use a pre-existing utility program or library to directly convert the student assignment into an ASCII text file. This solution, however, produces inconsistent results as each utility tends to output text in different ways, or not at all. A better solution is to convert the assignment to an HTML file and then use the command-line web-browser Lynx, to format and dump the text to a file using the –dump flag. PAIRwise uses this process, which produces the best and most consistent results across all file formats. After completion of the Conversion Module, the result is a collection of ASCII text files. The collection is now ready for comparison by the Comparison Module.

Comparison Module

Comparison of student assignments begins after conversion of each student assignment to ASCII text. The comparison looks for verbatim matches between all pairs of documents, and is where PAIRwise gets its name. PAIRwise does not, however, count every verbatim match at the word level. To avoid the situation where function words or common phrases would count as verbatim matches, it uses a minimum match length to count the amount of verbatim match. PAIRwise does not remove function words. Therefore, it counts all strings of consecutive words of at least the minimum match length. The total number of words in these matches is divided by the total number of words in a document to determine the verbatim match score. Each document of a pair will have its own unique score: the percentage of match relative to the other document. While the Comparison Module counts the verbatim word match length once, the Comparison Module must make two calculations of the final score for each document. These scores are then associated with each document and stored for later use by the Reporting Module.

The Comparison Module requires several steps to count all of the verbatim match windows. First, all punctuation is removed to prevent attempts to circumvent the system by merely changing the punctuation marks within an assignment. Next, all words are extracted from the document, with order maintained, and converted to all lower case letters, again, to prevent simple circumvention techniques. Next, it compares all pairs of student assignments to find minimum length verbatim matches. The module then removes from consideration any words that are part of a verbatim match window. The effect of this step is that the highest possible score is 100%. The module performs this process for all pairs of documents within a corpus of student assignments or documents from other sources.

The Comparison module then passes the scores to the Reporting Module, which visually displays all of the PAIRwise scores for the user. The user then determines if any possible cases of plagiarism exist.

Reporting Module

The goal of the reporting module is to allow users to easily find possible instances of plagiarism across pairs of documents. To meet this goal the Reporting Module produces two graphs: the first is a list of all scores for all pairs of documents sorted by the verbatim match percentage, and the second is a graph that zooms in on all scores above 10% verbatim matching. We chose the score of 10% based on observations that scores below 10% tend to be normal scores for common phrases used in assignments in the same class. Furthermore, prosecuting students that have 10% or less of verbatim match may be difficult if the case is not obvious. Figures 4 and 5 contain examples of these graphs.

The intent of the graph in Figure 4 is to give the user an overall view of how much verbatim match occurs in a collection of student assignments. In this graph, we see that most of the students did not have greater than 10% text in common, as demonstrated by the long green tail. There is, however, some concern about a few documents on the far left that warrants further investigation. Examination of the graph in Figure 5 will help the user identify instances of likely plagiarism.

The graph of Figure 5 highlights several interesting points. First, the assignments at the far left, those with greater than 40% in common, represent actual instances of plagiarism. At first glance, this result may seem like an epidemic outbreak of cheating, however, this is not the case. Remember, each bar represents a pair of assignments, and there are two scores recorded for each pair; therefore two bars are shown for each instance of plagiarism. Furthermore, this example shows 6 instances of cheating. Taken from a class of over 300 students, this result, while a concern, is not indicative of widespread plagiarism.

One might also observe in this graph the "mesa effect" created by assignments with approximately 20% or less verbatim matching scores. While this mesa may point to a large group of student who cheated, upon further investigation, we found that these students used the same series of quotes from Senator Dianne Feinstein. Here, PAIRwise found similar text in all the assignments, but the user determined that no actual plagiarism occurred.

To allow the user to further investigate any pair of documents with verbatim match scores greater than 10%, the Reporting Module provides links to each pair of documents. When a user selects a bar, PAIRwise displays a side-by-side comparison of the selected pair. Also, if the pointer hovers over any bar of the graph, the web browser will display a table with information about the pair. Figure 6 shows a view of this information.

The information displayed in the gray box in Figure 6 shows the assignment pair, with the name of each assignment. Also, the module gives the verbatim match score with respect to the first assignment listed under "PAIR." Here, the first assignment has a 75% match with assignment "848859-9".

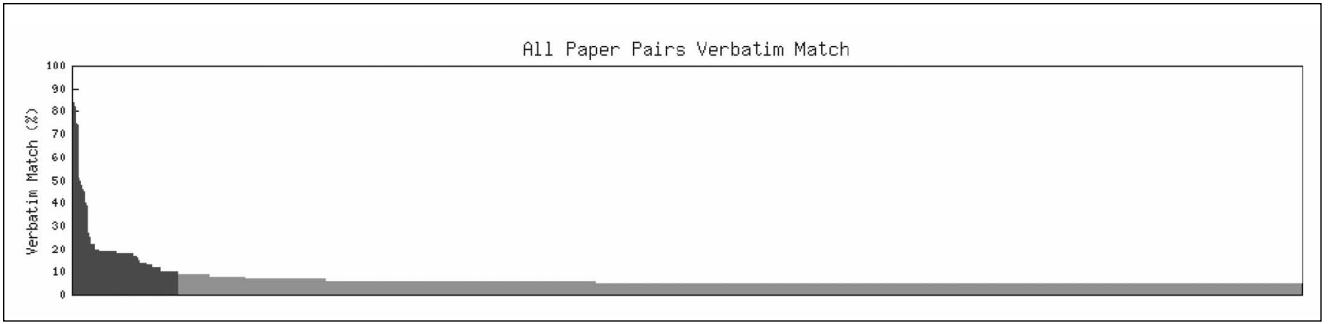


Figure 4. Example graph showing all assignment pair scores

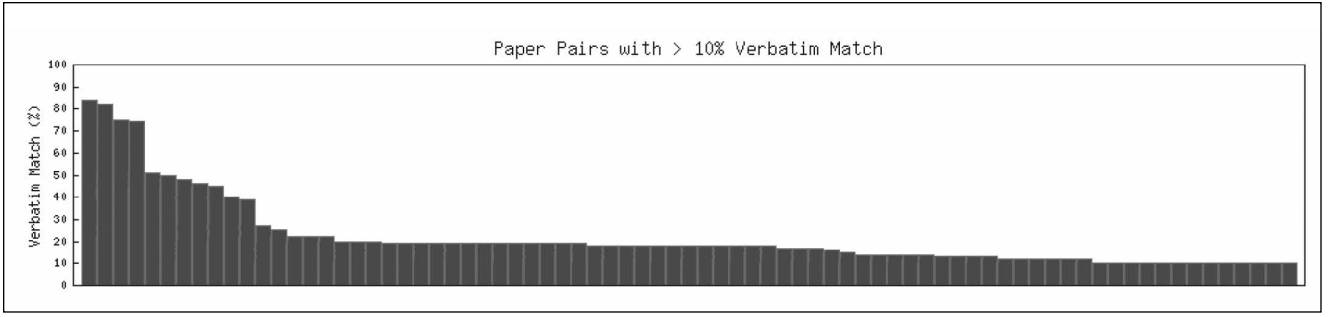


Figure 5. Example graph showing all assignment pairs with a verbatim match score greater than 10%

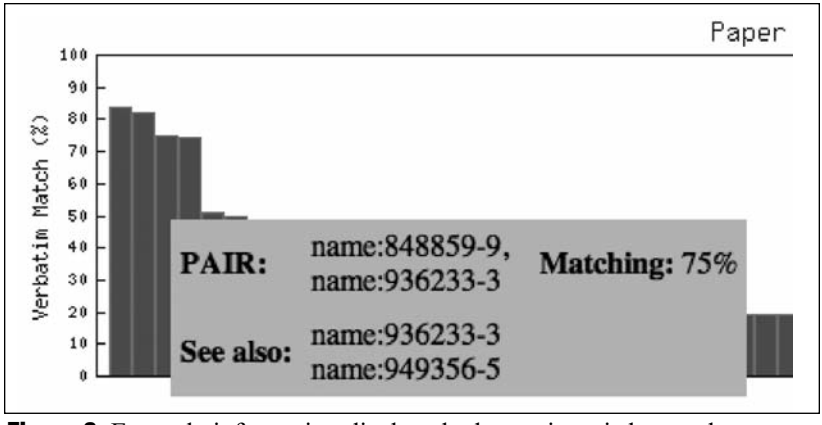


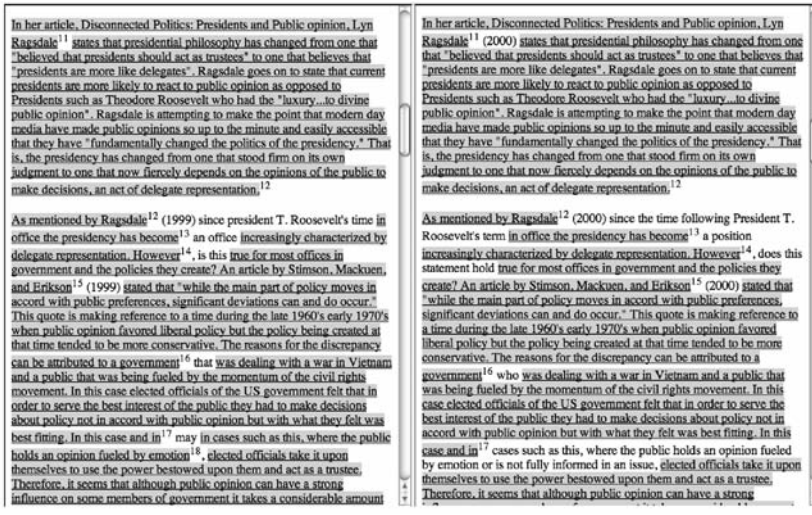
Figure 6. Example information displayed when pointer is hovered over a red bar

Below these two pieces of information is a list of other assignments, related to the first, for the user to examine. These two other assignments represent assignments that also have greater than a 10% match with respect to the first assignment. In this example, there are two listed, the second assignment in this pair and also another assignment. Upon further investigation the user found that a group of three students worked together to produce three separate assignments with at least 50% of the text in common between all the assignments. The school, in this case, decided to suspend each of these students for plagiarizing work from each other.

If the user clicks on any red bar, the Reporting Module displays a side-by-side comparison. An example comparison is shown in Figure 7. This figure shows two assignments with a high degree of verbatim match. The text highlighted and underlined represents matching text. The superscript number next to each chunk of verbatim text is a number that allows the user to correlate which passages are verbatim matches. The user can click on any of the numbers in the assignment on the left to find where they occurred in the assignment on the right. The module underlines matching passages to make identification easier in the printed version of the assignment. The visualization of the side-by-side comparison makes finding instances of plagiarism easy to find and offers good visual evidence when attempting to prosecute those students who have plagiarized.

EVALUATION

We performed a qualitative assessment to ascertain whether we met our two original goals for PAIRwise: for it to be effective and affordable as a tool to detect plagiarism.


In her article, Disconnected Politics: Presidents and Public opinion, Lyn Ragsdale[11] states that presidential philosophy has changed from one that "believed that presidents should act as trustees" to one that believes that "presidents are more like delegates". Ragsdale goes on to state that current presidents are more likely to react to public opinion as opposed to Presidents such as Theodore Roosevelt who had the "luxury...to divine public opinion". Ragsdale is attempting to make the point that modern day media have made public opinions so up to the minute and easily accessible that they have "fundamentally changed the politics of the presidency." That is, the presidency has changed from one that stood firm on its own judgment to one that now fiercely depends on the opinions of the public to make decisions, an act of delegate representation.[12]

As mentioned by Ragsdale[12] (1999) since president T. Roosevelt's time in office the presidency has become[13] an office increasingly characterized by delegate representation. However[14], is this true for most offices in government and the policies they create? An article by Stimson, Mackuen, and Erikson[15] (1999) stated that "while the main part of policy moves in accord with public preferences, significant deviations can and do occur." This quote is making reference to a time during the late 1960's early 1970's when public opinion favored liberal policy but the policy being created at that time tended to be more conservative. The reasons for the discrepancy can be attributed to a government[16] that was dealing with a war in Vietnam and a public that was being fueled by the momentum of the civil rights movement. In this case elected officials of the US government felt that in order to serve the best interest of the public they had to make decisions about policy not in accord with public opinion but with what they felt was best fitting. In this case and in[17] may in cases such as this, where the public holds an opinion fueled by emotion[18], elected officials take it upon themselves to use the power bestowed upon them and act as a trustee. Therefore, it seems that although public opinion can have a strong influence on some members of government it takes a considerable amount

In her article, Disconnected Politics: Presidents and Public opinion, Lyn Ragsdale[11] (2000) states that presidential philosophy has changed from one that "believed that presidents should act as trustees" to one that believes that "presidents are more like delegates". Ragsdale goes on to state that current presidents are more likely to react to public opinion as opposed to Presidents such as Theodore Roosevelt who had the "luxury...to divine public opinion". Ragsdale is attempting to make the point that modern day media have made public opinions so up to the minute and easily accessible that they have "fundamentally changed the politics of the presidency." That is, the presidency has changed from one that stood firm on its own judgment to one that now fiercely depends on the opinions of the public to make decisions, an act of delegate representation.[12]

As mentioned by Ragsdale[12] (2000) since the time following President T. Roosevelt's term in office the presidency has become[13] a position increasingly characterized by delegate representation. However[14], does this statement hold true for most offices in government and the policies they create? An article by Stimson, Mackuen, and Erikson[15] (2000) stated that "while the main part of policy moves in accord with public preferences, significant deviations can and do occur." This quote is making reference to a time during the late 1960's early 1970's when public opinion favored liberal policy but the policy being created at that time tended to be more conservative. The reasons for the discrepancy can be attributed to a government[16] who was dealing with a war in Vietnam and a public that was being fueled by the momentum of the civil rights movement. In this case elected officials of the US government felt that in order to serve the best interest of the public they had to make decisions about policy not in accord with public opinion but with what they felt was best fitting. In this case and in[17] cases such as this, where the public holds an opinion fueled by emotion or is not fully informed in an issue, elected officials take it upon themselves to use the power bestowed upon them and act as a trustee. Therefore, it seems that although public opinion can have a strong

Figure 7. Side-by-side view of two student assignments with a high percentage of verbatim matches

Effectiveness of PAIRwise

The goal of this section is not to outline PAIRwise's comparative effectiveness, but rather to show that PAIRwise is (1) at least as effective as previously used software and (2) capable of finding instances of Internet plagiarism, a feature not available in the previously used detection software. After all, as Braumoeller (Braumoeller, 2001) states, a plagiarism detection system need not find all instances of plagiarism in order to deter students from cheating.

To test PAIRwise's effectiveness, we needed assignments from several classes as well as results from another plagiarism detection system. Since our group had already evaluated another plagiarism detections system, we decided to process the same set of assignments used in this previous effort. The goal of this round of testing was to determine if PAIRwise was at least as effective as the system used in the previous work. In addition, since no Internet comparisons were done as part of the previous work, a large sample of these assignments was also compared to the Internet. These tests, therefore, mirror the exact methodology used in the previous study.

Our evaluation involved collecting all assignments electronically from 10 different offerings of 6 courses involving 7 different instructors. These classes included both general education courses and introductory survey courses at the University of California, Santa Barbara. The instructors for these classes used three different types of disclosure to students about the use of PAIRwise. We briefly comment on the relationship between this disclosure and the number of plagiarism instances detected. Three of the seven instruc-

tors informed their students only of the plagiarism policy, and did not disclose that any plagiarism detection system would be used. Two other instructors disclosed the use of such a system. The remaining two instructors used a compromise of the two extremes: they implied, but did not mention explicitly, that an automated detection system would be used.

After all assignments were collected, PAIRwise was used to compare each assignment to other assignments in the course, all previous instances of the course for which assignments were available, as well as the Internet. The original study only included comparisons of assignments within the same course. We found there were identifiable and prosecutable instances of plagiarism in approximately 1% of the assignments. That is, of the approximately 500 papers compared, we found 7 instances of plagiarism. These instances included 5 from the same course, also found in the previous study. Another instance found was one across multiple instances of the same course, also found in the previous study. Finally, PAIRwise found one instance of Internet plagiarism. This plagiarized assignment was not found during the previous study because the assignments were not checked against the Internet. All others assignments with similarity scores above the 10% threshold were determined to not be prosecutable instances of plagiarism. We forwarded the prosecutable instances of plagiarism to the proper authorities for further investigation. While the conclusions from these results remain anecdotal, they do give some indication that plagiarism detections systems are effective in identifying student plagiarism.

Affordability of PAIRwise

PAIRwise is affordable for at least two reasons. First, PAIRwise is available for download under an open source license. Second, PAIRwise runs on commodity computers. These two characteristics of PAIRwise make it easily available to institutions of any size. With a minimal investment of time and money, institutions as large as universities, and as small as primary schools can easily download and install PAIRwise and begin using it with a minimum of both time and money.

Not only is PAIRwise open source and available for download for free, but so are the libraries it uses. While all the libraries used are either in the public domain or released under the GNU's Not UNIX (GNU) General Public License (GPL), our own university's policy requires us to release PAIRwise under a slightly more restrictive license called the University of California Office of the President (UCOP) License. Because of the intellectual property policies of the University of California system, they have devised an open source license that allows for free use and redistribution for non-commercial purposes, but require some form of compensation for commercial use. While this licensing scheme is more restrictive than the GPL, it does allow most, if not all, educational institutions the ability to use this soft-

ware free of charge. Educational institutions are also free to change and redistribute PAIRwise, as they see fit, a feature not available in commercial plagiarism detection systems.

Another barrier to running one's own plagiarism detection system could be the expense of purchasing and maintain the necessary hardware. PAIRwise, however, avoids this barrier because it can be installed and operated on commodity PCs. Most commercial plagiarism detection systems are either entirely run in-house or require investment in expensive hardware. The large investment in hardware stems from the fact that plagiarism detection systems usually mirror the entire Internet, or at least a portion of it. PAIRwise, however, does not require mirroring the Internet. This feature means PAIRwise has no requirement for large storage devices. Furthermore, the amount of computational power required by PAIRwise is fairly small. The majority of the processing time is spent waiting for search results to be returned.

To date, we have developed PAIRwise for use on several platforms. Originally, we developed PAIRwise for the Linux operating system. This fact means that anyone with the right experience can install PAIRwise, not only on Linux, but other UNIX variants as well. For example, while we have not tested PAIRwise on Solaris, there should be only minor issues with getting such an installation to work. Mac OS X is also a good candidate for running PAIRwise. Since the move to the Mach kernel in version 10 of this operating system, Mac OS X is a fully functional UNIX machine, and as such, a good candidate for PAIRwise.

The Microsoft Windows operating system, however, presents several installation challenges, and so far we have not attempted to resolve these issues. While we have not made attempts to port PAIRwise to the Windows environment, it is possible. An alternative may be to install PAIRwise in a Windows environment with Cygwin installed. Cygwin is a port of many UNIX libraries to the Windows operating system and it may provide the means necessary to successfully install PAIRwise with a minimum of effort.

CONCLUSION

The original intent of the PAIR effort at CITS was to investigate the societal affects of plagiarism and to be the catalyst for a dialogue within and between educational institutions about these effects. At the start of this research, it became apparent that we needed an effective tool to help detect possible occurrences of plagiarism. The result of this need was PAIRwise, which this article has shown to be capable of detecting plagiarism from local sources as well as Internet sources. PAIRwise also possesses unique characteristics, not present in currently available commercial products for plagiarism detection. Because PAIRwise is open source and installable on commodity PCs, it is much more affordable than its commercial counterparts.

Student assignments remain in-house and are not transferred to outside commercial, for-profit companies, which avoids many of the legal issues with commercial systems. PAIRwise is also extendable and customizable because of its open source licensing. This extensibility means, not only can an institution improve upon PAIRwise, but they can also redistribute these changes. No commercially available system provides access to their software code, nor allows access to customize and extend it. This point relates to the last unique feature of PAIRwise: transparency. Because the code can be downloaded and viewed, anyone can test the PAIRwise techniques and algorithms. The commercially available systems have no means by which to measure or compare their effectiveness because of a lack of transparency. These unique features, as well as PAIRwise's effectiveness, make it a tool that has the potential to not only help in the understanding of the societal effects of plagiarism, but also allow it to positively impact academic integrity in a time of growing Internet popularity and use.

References

Bowyer, K.W., & Hall, L.O. (1999). *Experience using "MOSS" to detect cheating on programming assignments*. Paper presented at the 29th Annual Frontiers in Education Conference, FIE '99.

Braumoeller, B., & Gaines, B. (2001). Actions do speak louder than words: Deterring plagiarism with the use of plagiarism-detection software. *PS: Political Science & Politics, 34*, 835-839.

Carroll, J., & Appleton, J. (2001). Plagiarism: A good practice guide. JISC Learner Experience. Oxford Brookes University.

Chen, X., Li, M., Mckinnon, B. & Seker, A. (2002). *A theory of uncheatable program: Plagiarism detection and its practical implementation*. http://genome.math.uwaterloo.ca/SID/

Fry, E. (1977). Fry's readability graph: Clarifications, validity and extension to level 17. *Journal of Reading, 21*. School of Education, Open University.

Google (2003). http://www.google.com.

Gunning, R. (1952). *The technique of clear writing*. McGraw-Hill.

Joy, M., & Luck, M. (1999). Plagiarism in programming assignments. *Education, IEEE Transactions on, 4*(2), 129-133.

McCabe, D, Trevino, Linda K., & Butterfield, Kenneth D. (2002). Honor Codes and Other Contextual Influences on Academic Integrity. *Research in Higher Education, 43*(3), 357-378.

McLaughlin, H. (1969). SMOG grading -A new readability formula. *Journal of Reading, 22*, 639-646.

Merriam-Webster Online. (2003). http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=plagiarism

Microsoft. (2003). *Readability Scores, Spelling and Grammar*. http://office.microsoft.com/en-us/word/HP051863181033.aspx?pid=CH060830131033

Nielsen NetRatings (2006). Monthly Web Usage Data. http://www.nielsen-netratings.com/

Paulson, L. (2002). Professors use technology to fight plagiarism. *IEEE Computer, 35*(8), 23-25.

Plagiarism.org. (2003). Types of Plagiaris, *Research Resources*. http://www.plagiarism.org/learning_center/what_is_plagiarism.html

Plagiarism Resource Site (2003). *CopyFind*, 1.2, http://plagiarism.phys.virginia.edu/software.html

Shivakumar, N., & Garcia-Molina. (1995). *SCAM: A copy detection mechanism for digital documents*. Paper presented at the 2nd International Conference on Theory and Practice of Digital Libraries, Austin, TX

Si, A., Leong, Lau, H., & Rynson W. (1997). CHECK: A document plagiarism detection system. *ACM Symposium for Applied Computing*, February, pp. 70-77.

Turnitin.com. (2003). http://www.turnitin.com