A Web-Enabled Plagiarism Detection Tool

Colin J. Neill and Ganesh Shanmuganathan

Vol. 6, No. 5
September/October 2004

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author’s copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.
A Web-Enabled Plagiarism Detection Tool

The Internet and the World Wide Web have revolutionized information sharing and searching; it is difficult to remember what academic research was like without it, or how we could possibly live without it again. It is an awesomely powerful resource, and therein lays the rub. Never have the phrases, “With great power comes great responsibility,” and “Power corrupts and absolute power corrupts absolutely,” been so true. We could be talking about many issues with that introduction, anything from e-retail to hacking and virus dissemination, but it turns out that we’re talking about academic dishonesty. In particular we’re talking about plagiarism, or passing off another person’s work as your own.

Penn State University considers plagiarism to occur when an individual

• submits a paper written by someone else,
• quotes or paraphrases another paper without proper citation, or
• presents another person’s ideas without attribution.

How serious a problem is this? A report in the Journal of Higher Education stated that 75 percent of college students admit to some form of cheating and half admit to serious cheating on written assignments (D. McCabe, L. Trevino, and K. Butterfield, “Academic Integrity in Honor Code and Non-Honor Code Environments: A Qualitative Investigation,” J. Higher Education, vol. 70, no.2, 1999). An article in the Daily Pennsylvanian quotes McCabe as saying, “Students are growing up with technology that makes Internet plagiarism simple. It is easy to use, and almost all written sources are available on the Internet. … Some students actually believe that they’re not doing anything wrong. They have this attitude that they’re doing research. They don’t think that they need to cite because everything on the Internet is public information” (M.C. Peterson, “Download. Steal. Copy. Cheating at the University,” Daily Pennsylvanian, 27 November 2001). These are astonishing numbers, particularly considering that we have identified only a handful of cases at our campus, the Great Valley Graduate School. The question is, how many cases are we missing?

So this is clearly a big problem for universities and schools, but it is not just an educational problem. The publishing industry, and periodical publications in particular, face the same issues. Two recent editorials addressed the increasing occurrence of plagiarism, simultaneous submission (submitting the same work for review to multiple venues simultaneously), and republication (submitting previously published work without sufficient attribution), and how these practices are in breach of the copyright form signed by all authors, and the integrity at the heart of academic publishing (R.L. Haupt, “Plagiarism in Journal Articles,” IEEE Antennas and Propagation, Aug. 2003, vol. 45, no. 4, p. 102; and W.R. Stone, “Plagiarism, Duplicate Publication and Duplicate Submission: They Are All Wrong!” IEEE Antennas and Propagation, Aug. 2003, vol. 45, no. 4, pp. 47-49).

In fact, combating this worrisome trend has become a significant focus for all IEEE periodicals, which shows in changes at Manuscript Central, the IEEE’s online article submission system. Manuscript Central now requires authors to explicitly state that the submitted work is theirs,
that every author contributed directly in its writing, and that the work has not appeared previously in another publication and is not already in submission elsewhere. Under a proposed new policy, the IEEE will ban authors caught plagiarizing from publishing for up to five years, and will identify the offending article in the archives as such.

Returning to higher education, universities around the world are ramping up their efforts against plagiarism. Penn State has implemented academic integrity boards to rule on contested cases and to set standard sanctions for infractions, to enforce the inclusion of the university’s policy on academic integrity on every course syllabus that students receive, and to ensure that professors read that policy at the start of each course, including how the policy works within the context of that course. We have worked hard on the education and policy side of the problem at Penn State, but the issue of detection has not received enough attention; as we said, we have discovered only a handful of cases at our campus.

The Internet is the offenders’ biggest weapon. Fortunately, the professor can also wield the plagiarizer’s weapon of choice. If students use search engines to find the material to copy, professors can use them to find those original sources. Unfortunately this can be very time consuming. The student is only working on one paper, but the professor must grade and verify all of the students’ papers. To do this by hand can take more than an hour per paper, and this is in addition to the standard grading activities. Thankfully, services and tools are available to aid in the fight. We have also developed our own tool.

**AVAILABLE TOOLS**

Turnitin (http://www.turnitin.com) is the most well-known plagiarism detection service. Originally plagia-rism.com, it is a commercial service that iParadigms developed for registered individual educators or institutions. Professors and teachers submit papers to the site and receive results a day or so later. The site compares papers against an index of Internet content as well as large databases of “paper-mill” essays (essays available for purchase on the Web for use by students as term papers at school and university) and previously submitted papers. Recently a student at McGill University challenged using this service partly on the grounds that Turnitin subsequently adds the paper to its in-house database of material, constituting an economic benefit to Turnitin without compensation to the student. Despite this potential setback, the Joint Information Systems Committee (JISC), an organization representing all higher education institutions in the UK, recently selected Turnitin as its plagiarism detection service in the form of http://www.submit.ac.uk.

WordCHECK (http://www.wordchecksystems.com) is a stand-alone application that detects collusion between students in a course rather than plagiarism of external source material. To use the application, the professor or teacher loads all the documents into the system’s internal archive. The system compares all papers to detect copying within the class. The document comparison is based on keyword profiles (a type of linguistic fingerprint) and phrase matching. Although this system is not strictly detecting plagiarism, it could if the internal archive includes paper mill essays and similar content. Unfortunately, a 2001 review of the tool commissioned by the JISC (http://online.northumbria.ac.uk/faculties/art/information_studies/Imri/Jiscpas/docs/jisc/Detection_Technology.pdf) found its detection performance unsatisfactory.

EVE2 (http://www.canexus.com/ewe) is a commercial application that downloads to a user’s desktop and determines if students have copied material from the Internet. For each paper, the application generates a report, including the percentage that contains plagiarism, the list of URLs, and an annotated copy of the paper with the copied sections highlighted in red. The tool accepts several file formats, including plain text and Microsoft Word documents, but it will only generate annotated copies of papers from the plain text files. Essentially the tool is an interface for Web searches, but this simplicity does not limit its effectiveness. The only drawback, which the 2001 JISC report identifies, is that the searches are only against HTML Web content, and much of the material on the World Wide Web is in alternate formats.

WCopyFind (http://plagiarism.phys.virginia.edu/Wsoftware.html) is a freely available collusion detection tool that Prof. Lou Bloomfield developed at the University of Virginia. At least two versions exist, but the most useful version features a simple graphical interface that lets users load the set of documents into the internal archive of the tool, just as with WordCHECK. The tool compares these documents against each other and optionally against a separate archive of files (that the professor might have collected over the years) for matching phrases. The tool presents the results as HTML files and hyperlinks common phrases between documents to indicate which students in a class were colluding. Although it cannot search against Internet content, the tool is fast, very easy to use, and the results are clear.

**PLAGIARISM DETECTOR**

Some might consider this variety in tools sufficient, particularly as this is not at all an exhaustive list, but each has limitations or costs that accompany their use, and none leverages the most powerful weapon in the available arsenal: Google.com. Google now references more than 4 bil-
lion Web pages. Most importantly, its index includes non-HTML content such as PDF files, which are popular with online publishers, as well as article archives such as Citeseer. Indeed, any tool that cannot search these resources has limited usefulness, at least in an educational environment. The Internet is often the only tool many students use for topic research; the Penn State Great Valley library is practically a ghost town. In light of this, only Turnitin and EVE2 would be useful and each has its limitations. Turnitin is a subscription-only, commercial service that appears to aim more toward the detection of paper-mill essays (prewritten essays that students can purchase and use as their own), which might be common in high schools or undergraduate settings, but are rarely relevant or suitable for graduate school assignments. EVE2 is less costly and doesn’t focus on prewritten essays, but its scope only covers HTML Web content. Given these restrictions, we decided to develop our own Web-based plagiarism detection tool that accessed the Google searching capabilities directly.

The detector engine is a Java application that resides on a server for shared user access. It operates in a periodic sleep/wake mode. When the engine wakes it checks for any files to analyze in the input directory. When it finds files, it starts processing them and moves each processed file to the output directory along with the resultant analysis report and associated files. This makes the tool exceptionally easy to use. Registered users need only place the documents in the specified input directory and then return later to check the output directory for the HTML reports. As an aside, processing a typical term paper of approximately 4,000 words currently takes around three minutes. Figure 1 illustrates how the tool performs analysis in stages. We apologize to those interested readers, but we purposefully chose not to explain in detail the engine’s internal operation and the algorithmic aspects of the analysis and detection process. We are still wrestling with the issue of how to publish these details without revealing, to the enterprising student, a strategy to defeat the tool.

**Grammar analyzer**

The first stages of analysis relate to the grammatical content of the input file. The file is parsed using QTAG (available at http://web.bham.ac.uk/o.mason/software/tagger), a probabilistic part-of-speech tagging package that returns the part of speech for each word in the file (adjective, noun, pronoun). This tagging identifies words that the tool can eliminate from searches, and it determines changes in tense, voice and person as indicators suggesting multiple-authorship or cut-and-paste writing.

Additionally, Flesch scoring performs document readability measuring. The tool computes the Flesch Reading Ease Score and Flesch-Kincaid Grade Level for the entire document and for each paragraph. These are well-known readability measures based on the number of words per sentence and number of syllables per word that the tool uses as indicators of potential multiple-authorship under the assumption that a student’s scores won’t vary greatly section to section or paragraph to paragraph. The tool analyzes dispersion of the paragraph scores about the document mean to identify potential plagiarism “hot-spots” that it excerpts for Web searching.

**Google API**

As mentioned previously, the detector uses the Google search engine for Web searching. To do this automatically the detector accesses the search engine via the Google Web APIs service using the Simple Object Access Protocol (SOAP).

The tool decomposes the sections of text that the detector has identified as potential hot spots by the grammar analysis into search terms. The tool conducts decomposition in two ways: finding exact matches of sentence fragments (Google limits length to 10 words), or finding contextual matches of keywords (which the tool determines from the QTAG speech tagger). A user can configure the detector to do either of these directly, or leave the detector in default mode where context-match searches perform only if the results of exact matches are below a preset threshold.

The tool screens the list of Web links that Google returns to remove redundant and repeated URLs and to determine the most relevant matches for each section of text and the complete input file. The tool uses the number and relevance of “hits” as factors in the plagiarism score for the document, along with the standard deviation of the Flesch scores and grammatical analysis. The plagiarism score is on a scale of 1 to 5 and is indicated in the generated reports as a five-colored scale from green, for innocent, to red, for very guilty.

Some readers might be wondering why we don’t perform Web searches for the entire document sentence-by-sentence, but initial experiments demonstrated that this was impractical given the size and number of the input
documents. In addition, the usage policy of the Google API service limits users to 1,000 searches per day per user account and this ration is quickly wasted when searching sections of text that are unlikely to be plagiarized.

Example reports

We included some examples of the HTML reports that the detection tool generates, to demonstrate its utility. The tool generates a top-level report (Figure 2 shows an example), showing the paper title and the date the report was generated. The report shows the Flesch scores for the entire paper followed by each paragraph of text, the list of relevant URLs for that paragraph, and the difference between the overall Flesch scores and those of that paragraph. The “View” link then opens a separate page for each paragraph, as Figure 3 shows, presenting the paragraph text followed by excerpts from the returned URLs with the words common to both highlighted so that the user can quickly identify what text has found a match and, therefore, if the author indeed plagiarized that section of text.

In the first example, a graduate research paper on Model Driven Development, you can see from the portion of the top-level report in Figure 2 that the tool detected no plagiarism (indicated by the highlighting of the green segment of the score bar). Figure 3 shows the detailed report for the first paragraph and you can see that only individual terms are highlighted in the URL excerpt indicating that the text was not copied and the Google search merely found terms related to the paper topic.

Alternately, Figure 4 is a paper on software process return on investment that included substantial plagiarism. You can see that the red segment of the score bar is highlighted, indicating that the tool detected the plagiarism. When you follow the View link for the second paragraph in Figure 5, you see from the highlighted text that the author copied—without attribution—entire sentences from a Web page, clearly violating standard academic integrity policies, including that used at Penn State.

Future enhancements

Although we are very happy with the operation of the current version of the detection tool, we plan enhancements for future releases. In particular we are interested in
• using more of the information gleaned from grammatical parsing to isolate suspicious sections of submitted documents and therefore lighten the load on Web searching.
• adding reference checking capabilities so source-attributed sections do not factor in plagiarism scoring. Currently the tool does not determine if the author has correctly referenced copied material and this becomes an issue particularly when directly quoting a source; the relevance scores returned from Google will indicate plagiarism even when the author has properly cited the source. This might seem simple, but ensuring the author has properly and accurately cited quoted text is difficult considering the number of ways of doing it.
• experimenting with alternative search-term approaches to narrow search results. The exact-match and context (keyword) match approaches are effective, but we wonder whether alternative schemes of selecting keywords might be more effective, such as series of two-word terms.
• adding capabilities to process PDF, Microsoft Word documents, and HTML files. Currently we can only process ASCII text files. The only format we foresee difficulty with is the Word document since the application is built using Java and Microsoft has not created any standard APIs and the open source or freeware tools available struggle to keep up with Microsoft’s frequent changes in document format.

Of course detecting plagiarism, or any other form of academic dishonesty, is not the cure, we are only treating the symptoms of this social disease, but at least now we can diagnose the condition when it arises, and perhaps the threat of detection will be deterrent enough in most situations.

Colin J. Neill is an assistant professor of software engineering at Penn State. Contact him at cjneill@psu.edu.

Ganesh Shanmuganthan is a senior software engineer at SCT Corp. Contact him at gshanmu@sct.com.