Plagiarism Detection Software: Promises, Pitfalls, and Practices

43

Debora Weber-Wulff

Contents

Abstract

An increasing number of students at universities around the world seem to be submitting plagiarized texts to their instructors for credit, although no exact figures are available, either for how much was plagiarized in the past or how much is plagiarized now. Instructors, overwhelmed with an ever-increasing workload, wish for a simple method – rather like a litmus test – to quickly sort out the plagiarized works, so that they can concentrate their efforts on the rest of the students.

The good news is that software can be used to identify some text parallels that could constitute plagiarism. The bad news is that the reports are often not easy to interpret correctly, software can flag correctly referenced material as

D. Weber-Wulff (✉)
FB 4, University of Applied Sciences, HTW Berlin, Berlin, Germany
e-mail: weberwu@htw-berlin.de

© Springer Science+Business Media Singapore 2016
T. Bretag (ed.), *Handbook of Academic Integrity*,
DOI 10.1007/978-981-287-098-8_19

non-original content, and there are cases in which systems report no problems at all for heavily plagiarized texts, as studies conducted by the author in 2004, 2007, 2008, 2010, 2011, 2012 and 2013 have repeatedly shown. Different systems have also been shown in these studies to report various amounts of plagiarism for identical texts, as they use individual, often proprietary, algorithms and sometimes only examine samples of the text under investigation. This chapter will examine the promises and pitfalls of technology-based plagiarism detection and look at some good practices for using such software in a university setting.

Defining Plagiarism

Plagiarism is not just the exact copy of a portion of text from another person, whether done intentionally or not, but can be any one of a variety of misuses of other people's work without attribution. Before speaking about technology-assisted identification of plagiarism, there needs to be a precise definition of what constitutes plagiarism. That is where the trouble begins. As Teddi Fishman has noted, "Even among academics, there is no standard or agreed upon definition of plagiarism" (2009, p. 1). Even when a university policy or other guidelines exist, there will be cases that arise for which instructors may be unsure where the line is to be drawn between sloppy scholarship and plagiarism. Fishman proposed a five-part definition (2009, p. 5) that includes using words, ideas, or other products that are attributable to another person, but are not attributed. This can include not only copy-and-paste plagiarism, but also paraphrases or patchwritings that are not attributed, as well as translated works. She does not mention intention directly, only noting that she finds it to be plagiarism when it is used for some sort of personal gain. Other definitions focus on misappropriation of intellectual property, or speak of legalistic notions such as copyright violations or stealing. Intention is mentioned as part of a number of definitions of plagiarism, in particular in university policies concerning cases of academic misconduct. In the opinion of this author, a text contains plagiarism, no matter if the person submitting or publishing it intended to plagiarize or not. A reader cannot possibly know what the writer had in mind or the conditions under which the text was produced, and an algorithm inspecting the text will most certainly not be able to evaluate intention. The appearance of intent may be an issue in determining consequences or a sanction, but it has no bearing on the plagiarism found in the text itself. It is somewhat easier to outline what is **not** covered by any definition of plagiarism: falsified data or manipulated pictures, ghostwritten papers, honorary authorship, and such. These acts are most certainly academic misconduct, but they cannot be considered plagiarism. For a more detailed discussion of plagiarism definitions, see Weber-Wulff (2014a).

Thus, it is generally not possible to construct a technological solution for the determination of plagiarism, since any definition is inevitably open for interpretation (see, among others, Pecorari and Petrić 2014). If it is not possible to precisely define the term, it is not possible to write software that is able to reliably detect it. Instead, software is generally only able to signal identical word sequences (that could still be properly referenced) or to a minor extent text similarity. Software is useful as a tool for potentially identifying the use of words from other sources. But it cannot determine plagiarism – that is a judgment call that can only be taken by human beings, not algorithms.

Promises Made

Companies that provide so-called plagiarism detection software are eager to promise instructors what they want to hear: “Advanced online plagiarism detection” (CatchItFirst), “Originality check” (Turnitin), “Easy, quick and accurate” (Ephorus), “based on the latest research in computer linguistics” (PlagScan), “saves time” (Urkund). The expectation is that student papers and theses can be easily uploaded or sent by email to a system, which can quickly and reliably detect all of the plagiarism. Reports are expected that can be presented to an honors board or a unit coordinator that clearly identify the plagiarized portions and deliver a final verdict as to the severity of the copying. Calling the numbers reporting the level of plagiarism “originality scores” or speaking of “originality checking” is rather misleading, as originality cannot be proven. Some companies even promise that they can prove a text to be *free* of plagiarism. They offer “plagiarism-free” certificates that students can obtain to hand in with their papers. However, if no plagiarism is found, it could be that a source has not been identified because it is not available to the system, for various reasons. Only the presence of plagiarism can be demonstrated by comparison with a previously published source.

Even though the price for using some systems can be quite steep, as it is levied at a cost per student per year, institutions will purchase such software in the hope that it can be used to determine which papers are plagiarized and how much of the text is affected. Some schools even encourage the use of such software as a formative device. Students can submit a draft of their work to the system and see where they need to include additional citations, and thus learn more about academic writing. Many of the larger systems now offer integration with various learning management systems (LMS). For instructors who are used to their LMS giving them a protective environment for offering learning materials to their students, this can appear to check a student’s paper only with local databases. Actually, the papers are transmitted to the company servers, which may be located in another country, unless explicit provisions are made for an in-house solution. Some systems do offer the possibility of checking the papers only against a selected subset of the stored texts, for example, only the papers of that institution, but this still happens on the company’s servers.

Another promise can be seen to be a sort of confirmation bias that occurs when instructors test a system that successfully points out major plagiarism. The software was expected to find plagiarism, and it did in this one specific case. It is easy to then believe that the software can thus find all plagiarisms. This is not the case, as will be addressed in the next section.

Pitfalls of Existing Systems

There is a widespread notion in the general public that computers can solve any problem, if they are just given the proper algorithms. Computer science students learn, however, that there are many simple-sounding tasks that are not solvable with a computer. Plagiarism detection belongs to this type of problem, although there exist algorithms that provide partial solutions.

General Functionality

In general the identification of plagiarism in a text consists of two stages: Identifying potential sources and then determining the amount of matching text between each source and the document being examined. Systems generally use one of two methods for finding potential sources: Either a public search machine is queried with search terms extracted from the text, or the system uses its own database of potential sources. Such databases can be constructed and searched using any of a number of different algorithms.

Interpreting the Results

As studies conducted by the author have shown, one must be very cautious about the results that systems return. Most calculate some number that allegedly represents the gravity of the text overlap found. It is important to understand that since the various systems use different and unknown algorithms and often only examine a portion of the text in question, they will return different values when testing the same text.

In one paper prepared as a test case (Weber-Wulff et al. 2013), 92 % of the words were taken verbatim from one source and disguised by applying patchwriting techniques. One system returned a completely irrelevant source; one reported plagiarism, but less than 25 %. Other systems reported 35 %, 60 %, or at most 80 % text overlap. This was quite an extreme example, but in general no two systems will report the exact same value. One system even reports a quite different value if the test is repeated just 10 min later – a different portion of the text is apparently used as the examined sample each time the program is called.

In addition to a number, most systems will generate a report. Some reports are practically unintelligible: problematic language and layout, meaningless or inconsistent numbers, confusing markup – all contribute to making it difficult to interpret the results. Others are only difficult to manage in a university setting. There can be problems encountered in passing them on to others within the institution or to external examiners; they can be difficult to print out for storing in the student's file; and the result can be quite different if the report is later regenerated, because sources have now disappeared from the Internet. None of the systems provide information in their reports that would be necessary for preparing documentation for an academic integrity board, such as including the page and line numbers of the text overlap so that members of the board can check the accusation independent of any software. Some systems do not even provide a side-by-side documentation, so that the users have to search for the reported overlap in each document themselves. The layout and the descriptive text in the reports are often difficult to interpret – only with experience can the system reports be properly interpreted.

Many of the 15 systems in the 2013 test (Compilatio, Copyscape, Docoloc, Duplichecker, Ephorus, OAPS, PlagAware, Plagiarisma, PlagiarismDetect, PlagiarismFinder, PlagScan, PlagTracker, Strike Plagiarism, Turnitin, Urkund) were found to over-report plagiarism, that is, they reported more potential sources than were warranted, or they flagged properly quoted material as non-original, or even reported, as potential sources, documents that had no overlap whatsoever with the paper in question. The latter is quite troubling, especially if a number is reported but no evidence is provided to support that number. This is probably due to programming errors and has been seen in various systems.

False Positives and False Negatives

In addition to issues with the numbers reported and the reports, there is the frequently encountered problem of *false negatives* (Weber-Wulff et al. 2013). This is the case when plagiarism in a text is not flagged because the source was not found by the system. The source could be a book or paper that is not yet digitized, or a text that is not available on the open web and indexed by a search machine, or one that is publicly available, but not stored in the database of the software system. In any of these cases, a plagiarism detection system cannot match this source.

It is also possible that the system registers some plagiarism, but it is at such a low threshold, typically below 5 % of the text that it is considered to be irrelevant. The matches registered could be for minor identical phrases or identical reference items used. Indeed, there are even universities that specifically will not accept papers if the number their system returns is larger than some threshold, as was reported verbally at a user's group meeting the author attended in 2014. This can result in students applying superficial changes to their texts until the system returns an "acceptable" value. The text is still, however, a plagiarism, as it is not the work of the student.

The other side of the coin is *false positives*, which means that the system reports plagiarism where there actually is none. Some systems will present a number suggesting that a text is plagiarized, without being able to demonstrate how that number was calculated. It is often not clear what exactly a number means, even though sometimes presented with two decimal places or in multiple variations.

The text could be properly referenced, but indented or set off with German or French quotation marks; the system could be flagging references, which should actually be the same; or the text is joint work and two students working together submitted the same text, announcing this fact in the text itself. The system could also be registering that a text is identical to itself, meaning the text is already in the company's database.

There can be any of a number of reasons for this situation. The student could have tested their paper using a friend's account at a school that permits formative use of plagiarism detection software. The advisor could have used the system to test a first draft of the thesis, or two independent examiners both used the same software for testing a thesis. The first person to test will receive a negative report, but the second person will see the alarming notice that the thesis is a complete copy.

Database Issues

Quite a number of systems store submitted papers in order to check future papers against past papers. This perhaps sounds like a good idea for term papers that are written every semester by students on similar subjects. But the manner in which some universities use plagiarism detection services involves teachers submitting students' papers to the system. This can technically be a violation of the student's copyright, for example, under European copyright law, which is quite different from American or Canadian copyright law. Only the author, generally the student, can permit the work to be stored in a database belonging to a company. For a thesis that was prepared under a non-disclosure agreement, it is not at all possible for either the student or the teacher to legally submit it to a third-party server.

There are some systems that do not store papers in databases under the control of a company. Instead, the software is installed locally on the instructor's computer. These systems then use a search machine in order to look up phrases or text selections from the text under scrutiny. Some systems just mirror the search machine results; others collate and evaluate the results before presenting them to the user in a ranked order. However, such use of search machines is often limited to a certain number of queries per day. Unless the company offering the plagiarism detection software has a cooperation with a search machine, it may take quite some time to obtain results, especially at times of the year at which term papers or final theses are generally handed in. This type of software is quite ineffective, as it is time-consuming and the results are not very illuminative. Instructors are better off using manual plagiarism detection methods as described in the next section.

Dubious Services

There are also companies offering somewhat dubious services. There are companies that offer "free" plagiarism detection services with the intent of harvesting texts submitted for paper mill use. For example, one system states openly that

"9 months after your scan, we will automatically add it to our student database and it will be published on one of our study sites to allow other students to use it as an example of how to write a good essay" (ScanMyEssay n.d.). Another company offers a money-back deal on the purchase of their software if the buyer is unsatisfied. In Weber-Wulff and Köhler (2008) we note that the software that was delivered after the payment was made did not work, questions about the product to the support email listed went unanswered, and we were not the only ones unable to obtain a refund.

One company was also found that pitched plagiarism detection to students at an affordable price (Weber-Wulff and Pomerenke 2007). Papers submitted were handed in to a pirated Turnitin account. The reports obtained from Turnitin were manipulated to make them appear to have been produced by the company in question. When questioned, the company insisted it was just chance that they had the same results. In 2010 a text was designed that was completely original and stored in Turnitin's database as coming from a non-existent web page. All systems in the test were given this text to evaluate. Indeed, only this company registered 100 % overlap of this text with the non-existent site, proving that it was still using pirated accounts.

Collusion

There is one kind of plagiarism that some systems are able to detect well, however. This is when the software is able to analyze a closed set of documents. Each of the documents can be compared with all of the others to discover text parallels, although this is only effective for small numbers of documents, as the number of comparisons grows rapidly. It might seem unusual for there to be such a closed set available, but there are actually common situations at university in which this is indeed the case.

This is what is called *collusion*, or as Zauner (2014, p. 18) puts it, "*die böse Schwester der Teamarbeit*" (the evil sister of teamwork). Collusion happens, for example, when students have been specifically instructed to work alone and do not, or the instructions to work alone were not clearly communicated, and they work together to produce a text that each of them hands in as their own, perhaps only slightly altered (Sutherland-Smith 2013). This can happen in a large group of students who feel that their quite similar papers will not be noticed in the crowd, or when computing students are stumped by an assignment and hand in code they have copied from a fellow student or the Internet. They do not realize that there are so many ways of writing sentences or of coding algorithms that it is highly unlikely for two students to submit identical work.

A test of 18 systems focusing only on detecting collusion in texts and program code was conducted in 2012 at the HTW Berlin (Weber-Wulff et al. 2012). The test results showed that there is software available that is useful in detecting collusion, since the potential sources are among the papers submitted and not on the open Internet. However, the systems that were good at detecting text collusion were not very useful at detecting collusion in computer programs, and vice versa. It was also easier for the software to find collusion in texts than in program code.

Manual Plagiarism Detection

The previous section has shown that there are some problems involved with using so-called plagiarism detection software. They are not the accurate and reliable, time-saving tools that instructors want. There are, however, quite a number of simple tools other than such dedicated, all-in-one systems that are available to an instructor with a suspected case of plagiarism. This section will discuss a selection of those tools and methods.

The first and foremost strategy is reading a student's paper with a critical eye, as observing small quirks and errors may help spot plagiarism. Instructors are generally well attuned to shifts in writing style. They quickly see spelling errors and detect erroneous statements of facts. There is a good chance of finding the source for a plagiarized passage with only the use of a search machine. Choosing just three to five words from a paragraph on either side of a writing style shift, perhaps including a spelling error, or the exact wording of a factual error, and using these as search terms will often return a link to the source used. Nouns tend to be effective search term choices, since verb forms are easily changed or adjectives inserted or sentences mixed up. If a spelling error is used, make sure that the search machine is actually using the misspelling and not correcting the term.

Google offers additional databases that can be searched: Google Books has digitized many scholarly books, even recent ones, and Google Scholar offers an index of scientific papers and citations. Searching for a misspelled bibliography entry in Google Scholar can lead to an original paper that was the source of a copied passage, as bibliographic mistakes are often not corrected. And even if Google Books does not show more than perhaps a page or even only a few snippets from a potential source, if the material looks promising, it can be obtained from a library using interlibrary loan.

One can, of course, compare the potential source with the page in the paper being read manually. However, if there is suspicion that extensive portions could have been taken from this source, it might be worthwhile to scan a larger portion of the book. Many libraries have book scanners available to their patrons, often costing very little to use. The pages in question can be stored on a memory stick as pictures and then run through software that recognizes and extracts the text from the pictures. This process, called optical character recognition (OCR), is sometimes even offered by the scanner software installed on the machine.

Once a suspected source has been obtained, it can be easily compared with the student's text. It is advantageous if the student's text has been handed in digitally, but if not, it can also be digitized as explained above. Using a highly effective algorithm called SIM_TEXT that was developed by Dutch computer scientists (Grune and Huntjens 1989), the similarities can be quickly marked. A contributor to the German public plagiarism documentation platform VroniPlag Wiki has implemented the algorithm for free use in any browser (VroniPlag Wiki n.d.), as shown in Fig. 1. Since this program runs locally on the user's machine, there is no copy of the text transmitted over the Internet.

According to a study conducted by Turnitin, one of the most popular sources for students is the Wikipedia (Turnitin 2011, p. 3). Wikipedia is perhaps perceived by students as being "free," so they do not see it as plagiarism to copy from it. However, Wikipedia is under a Creative Commons license, CC-BY-SA (http://creativecommons.org/licenses/by-sa/4.0/), which requires that any use of text must include a link back to the Wikipedia article and the list of authors for that particular article, as well as putting the usage under the same license. Wikipedia even offers a link on every page for generating a proper citation to that page.

There is an experimental tool, PicaPica (n.d.), that compares a text with an entire Wikipedia to determine if any portions of the text are close or identical to Wikipedia pages. It is rather reliably able to detect copies from a number of languages, but it does not detect translations of Wikipedia articles (Weber-Wulff 2014b). It is also possible to search in older versions of a Wikipedia in order to see what a page looked like on a particular date or to find the date at which a particular sentence or phrase was introduced into a specific article. WikiBlame (Flominator n.d.) is a simple but useful tool that will look back through an article's revision history and attempt to identify when a particular wording first occurred.

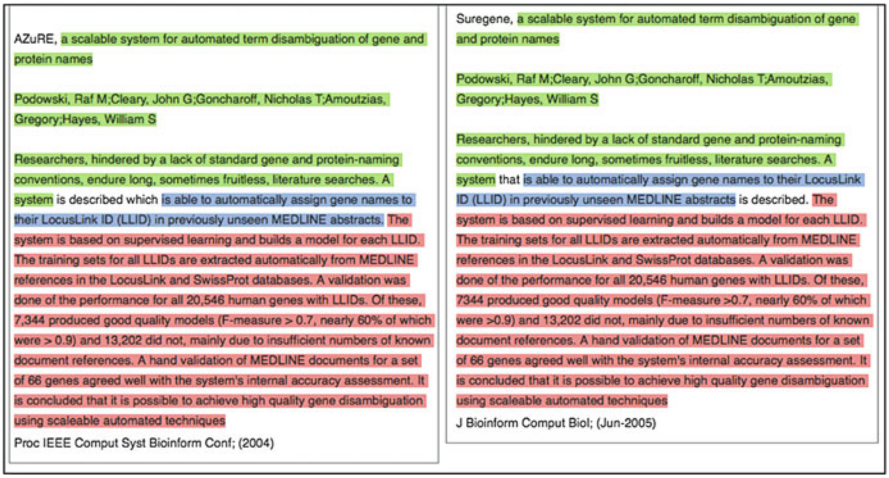
If there is a suspicion that a student has copied a picture from the Internet, there are a few possibilities for finding it. There are tools such as TinEye (n.d.), a free online service that has indexed billions of web pictures, or Google Image search. With both tools a picture can be uploaded to the site, or the URL of an online picture entered and a search is made for sites that have versions of this picture. They can even find pictures that have had modifications made on them. If good keywords describing the picture can be found, both Google Image search and the Wikimedia Commons (n.d.) can be used to find pictures that are potential sources.

Thus, there are a number of reliable, free tools that instructors can use for finding plagiarized sources. It is not necessary for them or their institutions to purchase software for finding text matches. Such software can, however, be an additional tool to use when a text reads as if it is plagiarized, but potential sources cannot otherwise be located.

Current Research in Plagiarism Detection Technology

In order to broaden the ability of software to effectively find text matches, there are a number of areas in which current research is being conducted. However, the systems are not available as products, and none will be able to offer the "litmus test" so many instructors and administrators wish to have, even if they do explore innovative ideas for detecting text similarities.

There is a research group at the University of Weimar in Germany (Meyer zu Eissen and Stein 2006) that is attempting to automate **intrinsic plagiarism detection**, that is, determining that a paper is a plagiarism by analyzing the internal



AZuRE, a scalable system for automated term disambiguation of gene and protein names

Podowski, Raf M;Cleary, John G;Goncharoff, Nicholas T;Amoutzias, Gregory;Hayes, William S

Researchers, hindered by a lack of standard gene and protein-naming conventions, endure long, sometimes fruitless, literature searches. A system is described which is able to automatically assign gene names to their LocusLink ID (LLID) in previously unseen MEDLINE abstracts. The system is based on supervised learning and builds a model for each LLID. The training sets for all LLIDs are extracted automatically from MEDLINE references in the LocusLink and SwissProt databases. A validation was done of the performance for all 20,546 human genes with LLIDs. Of these, 7,344 produced good quality models (F-measure > 0.7, nearly 60% of which were > 0.9) and 13,202 did not, mainly due to insufficient numbers of known document references. A hand validation of MEDLINE documents for a set of 66 genes agreed well with the system's internal accuracy assessment. It is concluded that it is possible to achieve high quality gene disambiguation using scaleable automated techniques

Proc IEEE Comput Syst Bioinform Conf; (2004)

Suregene, a scalable system for automated term disambiguation of gene and protein names

Podowski, Raf M;Cleary, John G;Goncharoff, Nicholas T;Amoutzias, Gregory;Hayes, William S

Researchers, hindered by a lack of standard gene and protein-naming conventions, endure long, sometimes fruitless, literature searches. A system that is able to automatically assign gene names to their LocusLink ID (LLID) in previously unseen MEDLINE abstracts is described. The system is based on supervised learning and builds a model for each LLID. The training sets for all LLIDs are extracted automatically from MEDLINE references in the LocusLink and SwissProt databases. A validation was done of the performance for all 20,546 human genes with LLIDs. Of these, 7344 produced good quality models (F-measure >0.7, nearly 60% of which were >0.9) and 13,202 did not, mainly due to insufficient numbers of known document references. A hand validation of MEDLINE documents for a set of 66 genes agreed well with the system's internal accuracy assessment. It is concluded that it is possible to achieve high quality gene disambiguation using scaleable automated techniques

J Bioinform Comput Biol; (Jun-2005)

Fig. 1 Comparing text with the VroniPlag Wiki tool. One text can be pasted into the box on the left, another into the box on the right. The minimum number of successive, identical words that are to be marked can be changed using the drop-down list, four words is the default value. The resulting page colors identical text in the same color in both columns, changing to another color when some difference is encountered – a word missing or added or changed spelling. The resulting page gives a good idea about the extent of the exact overlap, and the parts that have been changed stand out clearly. Even though this is a simple method, it is effective and it is possible to print out the results

structure and uncovering changes in style instead of finding possible sources. This is closely related to the authorship identification problem. That is, given a document of unknown authorship and a collection of documents for which the author is known, can the text be classified as having been written by one author in particular? A yearly workshop (PAN 2015) is held in which research teams train their experimental systems on data provided and see how well they fare on unknown data.

A research group at the University of Constance in Germany has been looking into **citation-based plagiarism detection**. In this method, the text is ignored and only the identity and order of the references and the citation patterns are compared. Currently, the citations have to be hand-coded, so this precludes the use of the technique on a larger scale. But it has been shown that, for example, translation plagiarism can be detected if the citation patterns used in the text have a strong overlap (Gipp 2014).

There are quite a number of **semantic plagiarism detection** methods under investigation by various research groups around the world. These highly experimental techniques try to map the meaning of a text and look for documents that display a similar meaning structure. This can be as simple as looking for synonym replacement, or word insertion and deletion, or word rearrangement. There are also experimental systems that attempt to glean the meaning from paragraphs for comparison to others, but here, too, there are no systems even close to being available for general use.

The Practice of Software-Based Plagiarism Detection

Software cannot accurately determine plagiarism; it can only indicate potential plagiarism. The decision whether or not a text parallel indeed constitutes plagiarism can only be determined by a person, as has often been stated in this chapter and elsewhere in this handbook. The interpretation of the reports generated by such systems is not an easy task. Training is required in order to be able to use the results to arrive at a conclusion. Basing a decision or a sanction only on a number produced by an unknown algorithm is irresponsible, as this indicates a lack of true understanding of the meaning of the numbers. Different programs will generate different numbers; some will even report a different value if the text in question is re-examined. How should institutions use such software, if at all? It is important that the promises of the software, whether implied by the company marketing the software, or imagined by the purchaser, be tempered with a realistic view of the capabilities of the systems. The pitfalls are many, and they can lead both to false accusations of plagiarism as well as the incorrect assumption of originality. If the use of the systems is so difficult, the question arises as to whether or not text-matching software should be routinely used in a university setting. Since there are three major roles at a university that are affected by plagiarism, each of them needs to be considered independently.

Should *students* be given the opportunity to use such software as a formative device for checking their papers before turning them in? As much as they might want to do so, it is wishful thinking to hope that software can prove originality. This use of the software could encourage novice writers to write to the software, that is, change around their wording enough or substitute enough synonyms for the software not to report too much identical text. There are even free tools available to students that will automatically replace enough words with a synonym so that plagiarism detection software will not identify a text match. The results of such machinations are quite unreadable, and this leads away from the goal of teaching students how to write coherently in their own words.

Should *instructors* be able to use the software on the papers their students write? In general, a comprehensive use of software on all written material from students can send the wrong message that students are assumed to be guilty until “proven” innocent. Instead, a university may wish to make one or two systems available for situations when instructors are suspicious of a paper and have not been successful in finding a source using search engines on the web. Since the use of the systems can be difficult and the interpretation of the reports is not necessarily a simple task, there should be a central service point, perhaps in the university library or computer center, tasked with helping instructors use the systems. Even if many software companies prefer to have all papers submitted because they keep copies in their databases, it is important that the universities make it clear that they are the customers and that they expect to have appropriate pricing models for their desired manner of use. Having multiple systems at the instructor’s disposal is important, as different systems do report different results, depending on the algorithms and databases used.

An extremely important point is to be clear about the copyright situation. Since the students are the authors of their papers, they are the copyright owners. A university may need to adjust their regulations so that students give implicit permission to check their texts if they are enrolled at the university, or they must give explicit permission to use their texts for plagiarism detection every time they submit, since a copy of the paper will be uploaded for checking to a server that is somewhere outside of the control of the university. Even if the software company promises that they do not retain a copy, it must be ensured that this is indeed the case.

Should *researchers* be able to use the software on scientific papers that they have written? This should definitely be an option, but the same caveats apply both with respect to false positives and also to false negatives. Testing their own papers and seeing the results on something that they know they wrote themselves can be sobering—and temper premature accusations against students.

Dealing with plagiarism after it has happened is time-consuming and frustrating. It is better to prevent it from happening in the first place. By far the most effective means of combating plagiarism is to educate students in the art of referencing and about scientific writing. It must be made clear that referencing is not a painful sort of academic torture, but is done for a number of justifiable reasons (Williams and Carroll 2009). Diane Pecorari (2013) has collected many ideas on how to teach students good source use, especially for second-language writers. Students must be taught how to write and given ample opportunity to practice. Other chapters of this handbook provide more detail on this topic.

Summary

This chapter has looked briefly at the promises made by companies marketing plagiarism detection or text-matching software and the expectations of the users at universities for such software, and then some of the many pitfalls and problems that are associated with their use were described. The unclear meaning of the numbers returned by the systems, the reports that are difficult to interpret, and the indication of plagiarism where there is none (false positives) as well as not reporting plagiarism where there is indeed some (false negatives) are the major problems in the use of such systems.

The promises of plagiarism detection systems are plentiful, but the pitfalls are complex and deep. In practice, the software should not routinely be used on all student texts, but only used as an additional tool in the academic integrity toolkit of an institution. Software cannot be the only instrument for determining plagiarism, as algorithms can be badly modeled or wrongly implemented. They can only deliver evidence that must be evaluated by a human being in order to determine if a text is a plagiarism or not.

References

Fishman, T. (2009). "We know it when we see it" is not good enough: toward a standard definition of plagiarism that transcends theft, fraud, and copyright. In *Proceedings of the 4th Asia Pacific Conference on Educational Integrity (4APCEI) 28–30 September, University of Wollongong, NSW, Australia*. http://www.bmartin.cc/pubs/09-4apcei/4apcei-Fishman.pdf. Accessed 17 Apr 2015.

Flominator. (n.d.). *WikiBlame*. [Web page]. http://wikipedia.ramselehof.de/wikiblame.php. Accessed 17 Apr 2015.

Gipp, B. (2014). *Citation-based plagiarism detection: Detecting disguised and cross-language plagiarism using citation pattern analysis*. Berlin: Springer Vieweg.

Grune, D., & Huntjens, M. (1989). Het detecteren van kopieën bij informatica-practica. In *Informatie, 31*(11), 864–867. English translation available at http://dickgrune.com/Programs/similarity_tester/Paper.ps and the program code at http://dickgrune.com/Programs/similarity_tester/. Accessed 17 Apr 2015.

Meyer zu Eissen, S., & Stein, B. (2006). Intrinsic plagiarism detection. In M. Lalmas et al. (Ed.), *Presented at the ECIR 2006* (LNCS 3936, pp. 565–569). London: Springer. http://ccc.inaoep.mx/~villasen/bib/Intrinsic%20Plagiarism%20Detection.pdf. Accessed 17 Apr 2015.

PAN 2015. (2015). *Plagiarism detection, author identification, author profiling. [Yearly Competition at the University of Weimar]*. http://pan.webis.de/. Accessed 12 Apr 2015.

Pecorari, D. (2013). *Teaching to avoid plagiarism: How to promote good source use*. Maidenhead: Open University Press.

Pecorari, D., & Petrić, B. (2014). Plagiarism in second-language writing. In *Language Teaching, 47*, 269–302.

PicaPica. (n.d.). *Compare a text to Wikipedia*. http://www.picapica.org/. Accessed 17 Apr 2015.

ScanMyEssay. (n.d.). *How does viper use my essay/dissertation?* [Web page]. http://www.scanmyessay.com/viper-use-essay.php. Accessed 4 Apr 2015.

Sutherland-Smith, W. (2013). Crossing the line: Collusion or collaboration in university group work? In *Australian Universities Review, 55*(1), 51–58.

TinEye. (n.d.). *Reverse image search*. http://tineye.com/. Accessed 17 Apr 2015.

Turnitin. (2011). *Plagiarism and the web: Myths and realities. An analytical study on where students find unoriginal content on the internet*. [White paper]. http://turnitin.com/static/resources/documentation/turnitin/company/Turnitin_Whitepaper_Plagiarism_Web.pdf. Accessed 12 Apr 2015.

VroniPlagWiki. (n.d.). *Quelle:Textvergleich*. [Web page]. http://de.vroniplag.wikia.com/wiki/Quelle:Textvergleich. Accessed 17 Apr 2015.

Weber-Wulff, D. (n.d.). Test of plagiarism software. [Web site], prepared with assistance from Wohnsdorf, G., Pomerenke, M., Köhler, K., Möller, C., Touras, J., Zarzecki, M., & Zincke, E. http://plagiat.htw-berlin.de/software-en. Accessed 12 Apr 2015.

Weber-Wulff, D. (2014a). *False feathers—a perspective on academic plagiarism*. Berlin: Springer.

Weber-Wulff, D. (2014b). Test of the picapedia system. [Blog entry]. In *Copy, shake & paste*. http://copy-shake-paste.blogspot.de/2014/05/test-of-picapedia-system.html. Accessed 12 Apr 2015.

Weber-Wulff, D., & Köhler, K. (2008). *Test 2008: – S25 Eve2*. [Web page]. http://plagiat.htw-berlin.de/software/2008-3/bewertung/s25-eve2/. Accessed 4 Apr 2015.

Weber-Wulff, D., & Pomerenke, M. (2007). *Eine kuriose Geschichte: Turnitin und iPlagiarismCheck 2007*. http://plagiat.htw-berlin.de/software/2007-2/kurios/. Accessed 17 Apr 2015.

Weber-Wulff, D., Köhler, K., & Möller, C. (2012). *Collusion detection system test report 2012*. [Web page]. http://plagiat.htw-berlin.de/collusion-test-2012/. Accessed 11 Apr 2015.

Weber-Wulff, D., Möller, C., Touras, J., & Zincke, E. (2013). *Plagiarism detection software test 2013*. [Web page]. http://plagiat.htw-berlin.de/software-en/test2013/report-2013/. Accessed 5 Apr 2015.

Wikimedia Commons. (n.d.). *A database of 23,539,005 freely usable media files to which anyone can contribute*. http://commons.wikimedia.org/wiki/Main_Page. Accessed 17 Apr 2015.

Williams, K., & Carroll, J. (2009). *Referencing & understanding plagiarism.* Basingstoke: Palgrave Macmillian.

Zauner, H. (2014). Wissenschaftliches Fehlverhalten—Münsteraner Kettenplagiate. In *Laborjournal, 09*, 17–18.