

AUTOMATIC STUDENT PLAGIARISM DETECTION: FUTURE PERSPECTIVES

Maxim Mozgovoy¹
University of Aizu
Tsuruga, Ikki-machi, Aizu-Wakamatsu, Fukushima, 965-8580 JAPAN
Phone: +81 242-37-2664
Fax: +81 242-37-2528

Tuomo Kakkonen
School of Computing
University of Eastern Finland
P.O. Box 111, 80101 Joensuu, FINLAND
Phone: +358 (0)13 251 7928
Fax: +358 (0)13 251 7955

Georgina Cosma
Department of Business Computing
P.A. College
P.O. Box 40763, 6307, Larnaca, CYPRUS
Phone: +357 24 021 520
Fax: +357 24 624 989

ABSTRACT

The availability and use of computers in teaching has seen an increase in the rate of plagiarism among students because of the wide availability of electronic texts online. While computer tools that have appeared in the recent years are capable of detecting simple forms of plagiarism, such as copy-paste, a number of recent research studies devoted to evaluation and comparison of plagiarism detection tools revealed that these contain limitations in detecting complex forms of plagiarism such as extensive paraphrasing and use of technical tricks, such as replacing original characters with similar-looking characters from foreign alphabets.

This article investigates limitations in automatic detection of student plagiarism and proposes ways on how these issues could be tackled in future systems by applying various natural language processing and information retrieval technologies. A classification of types of plagiarism is presented, and an analysis is provided of the most promising technologies that have the potential of dealing with the limitations of current state-of-the-art systems. Furthermore, the article concludes with a discussion on legal and ethical issues related to the use of plagiarism detection software. The article, hence, provides a "roadmap" for developing the next generation of plagiarism detection systems.

¹ The corresponding author.

1 INTRODUCTION

Student plagiarism is a growing problem in academic institutions. Plagiarism is often expressed as copying someone else's work (e.g., from other students or from sources such as course textbooks), and failing to provide appropriate acknowledgment of the source (i.e., the originator of the materials reproduced) (Cosma and Joy, 2008).

The prosperity of online resources that exist is a major factor that contributes to the increase of plagiarism incidents in academia since it has made it easier for students to cheat (Lathrop and Foss, 2000). Bennett (2005) conducted a detailed study on factors motivating students to plagiarise and "*means and opportunity*" was one of the factors reported. According to Bennett's study, the fact that resources are readily available and easily accessible over the Internet makes it convenient for students to gain instant and easy access to large amounts of information from many sources. Furthermore, many Internet sites exist that provide ready essays to students, and many of these sites even provide chargeable services for writing custom essays and papers. The ease with which students can obtain material from online sources to use in their academic work, has raised concerns in a number of other plagiarism related studies (Scanlon and Neumann, 2002; Kasprzak and Nixon, 2004; Nadelson, 2007).

Nadelson (2007) performed a survey to gather the perceptions of 72 academics on issues concerned with academic misconduct and reported 570 incidents of suspected plagiarism. The majority of incidents reported were 'accidental/ unintentional plagiarism' with 134 of those incidents involving undergraduate students and 39 involving graduate students. Furthermore, the academics reported that a large number of incidents involved students submitting papers copied from the Internet. Incidents concerning 'purposeful plagiarism', 'class test cheating' and 'take home test cheating' were also reported.

Plagiarism is also a problem in programming courses. Culwin, et al. (2001) conducted a study of source-code plagiarism in which they obtained data from 55 United Kingdom (UK) Higher Education (HE) computing schools. They found that 50% of the 293 academics who participated in their survey believed that plagiarism has increased in recent years. Furthermore, 22 out of 49 respondents provided estimates ranging from 20% to 50% of students plagiarizing in initial programming courses.

In the context of academic work, plagiarism is an academic offence and not a legal offence, and is controlled by institutional rules and regulations (Myers, 1998; Larkham and Manns, 2002). Therefore, what constitutes plagiarism is perceived differently across institutions. All universities regard plagiarism as a form of cheating or academic misconduct, but their rules and regulations for dealing with suspected cases of plagiarism vary widely, and the penalties imposed on cheating depend on factors such as the severity of the offence and whether the student admits to the offence. These penalties vary amongst institutions, and include giving a zero mark for the plagiarised work, resubmission of the work, and in serious cases of plagiarism the penalty can be expulsion from the university (Cosma and Joy, 2008).

Automatic and computer-aided plagiarism detection systems are developed to detect plagiarism in student works, and the detection effectiveness of such systems depends on the types of plagiarism they can detect. Such systems provide invaluable benefits with regards to saving time and effort of academics in performing the detection

process themselves. Computerized plagiarism detection has drawn academic interest in the past two decades due to the fact that the use of such tools reduces academic workload by automating the comparison process and quickly revealing groups of similar student works, which the academics need to scrutinize for suspicious similarity.

The earlier works in the evaluation of plagiarism detection systems have concentrated mostly on describing the various advantages and shortcomings of particular plagiarism detection systems (e.g. Clough, 2000; Lancaster and Culwin, 2004).

The use of computer-aided plagiarism detection also concerns a set of ethical and legal issues (see, e.g., Foster, 2002). These issues are caused both by technical imperfectness of plagiarism detection algorithms (for example, a system might incorrectly suspect a student's work as plagiarized) and by misunderstanding the role of plagiarism detection software in educational process. Due to the importance and the rising interest in ethics of automated plagiarism detection, the paper analyzes these matters and considers the purely technical problems associated with automatic detection.

Kakkonen and Mozgovoy (2010) performed a systematic evaluation of eight existing academic and commercial plagiarism detection systems for student texts. The systems evaluated in the study were *AntiPlagiarist* (ACNP Software, 2010), *EVE2* (Canexus, 2010), *Plagiarism-Finder* (Mediaphor, 2010), *SafeAssignment* (Sciworth Inc, 2010), *SeeSources.com* (2010), *Sherlock* (Joy and Luck, 1999), *TurnitIn* (iParadigms, 2010), and *WCOPYFind* (Bloomfield, 2010). The main result that arose from their work was that currently available detection systems have several drawbacks which can be divided into two main categories:

- shortcomings in the implementation of a particular detection system (for example, issues in the user-friendliness of the system), and
- problems caused by the limitations of the existing technologies for plagiarism detection.

There appears to exist a gap in the literature on evaluations on the limitations of state-of-the-art plagiarism detection systems, and possible solutions to addressing these limitations. The aim of this paper is to continue the work discussed in Kakkonen and Mozgovoy (2010) by elaborating on the limitations of existing technologies and propose ways to address these problems by using the latest results from other fields of research, in particular, *computational linguistics*, *information retrieval* and *natural language processing*.

The article is organized as follows. Section 2, represents our classification of plagiarism types that will be used throughout the study as a basis for the analyses. The section also outlines the various types of plagiarism detection systems that exist. Section 3 shortly discusses the current state-of-the-art in automatic plagiarism detection. In Section 4, provides an analysis of methods that could be applied in advancing beyond the state-of-the-art in plagiarism detection. Section 5 provides a discussion of the various ethical issues connected with automatic plagiarism detection, and finally Section 6 concludes with some final remarks and outlines opportunities for future work.

2 TYPES OF PLAGIARISM AND DETECTION SYSTEMS

2.1 Classification of plagiarism types

Dick *et al.* (2003), for example, categorized the types of cheating behavior related to plagiarism offences into copying, exams, collaboration, and deception. Students may use various techniques for disguising plagiarism in their submitted work, regardless of the type of cheating behavior. A classification of plagiarism types is a necessity in order to understand the difficulties of automatic plagiarism detection systems. Table 1 represents the five levels of the classification inspired by the work of Maurer *et al.* (2006), and developed further in Kakkonen and Mozgovoy, 2010.

Plagiarism type	Examples
(1) Verbatim copying	Copy-paste copying from an electronic source. This includes blatant plagiarism or authorship plagiarism, which refers to taking someone else's text and putting one's own name to it.
	Word-for-word transcription of texts from a non-electronic source.
(2) Hiding the instances of plagiarism by paraphrasing	Adding, replacing or removing characters.
	Adding or removing words.
	Adding deliberate spelling and grammar mistakes.
	Replacing words with words that have similar meaning (synonyms)
	Reordering sentences and phrases (structural changes).
	Effecting changes to grammar and style.
(3) Technical tricks exploiting weaknesses of current automatic plagiarism detection systems	The insertion of similar-looking characters from foreign alphabets. Thus, for example, the letter "O" can be equally well represented with the following three different characters: Unicode 004F (Latin O), 039F (Greek Omicron), and 041E (Cyrillic O).
	The insertion of invisible white-colored letters into what seem to be blank spaces. Most modern text processors allow the user to specify a font color in a document. The plagiarizer could exploit this feature by inserting a white font in a blank space with a white background. This would have the effect of distorting the content of the text even though, to the naked eye, it would be visually identical to the original.
	The insertion of scanned text pages as images into a document. This technique exploits the fact that existing plagiarism detection systems are incapable of comparing images.
(4) Deliberate inaccurate use of references	The improper and inaccurate use of quotation marks: the failure to identify cited text with the necessary accuracy.
	Providing fake references, i.e. made-up references that do not exist (fabrication), and thus fail to cite and reference text accurately.

Running head: Automatic student plagiarism detection: future perspectives

	Providing false references, i.e. references exist but do not match the text being referenced (falsification), and thus fail to cite and reference text accurately.
	The use of “forgotten” or expired links to sources: the addition of quotations or parentheses but a failure to provide information or up-to-date links to the sources.
(5) “Tough plagiarism”, i.e. the types of plagiarism that are particularly difficult to detect for both humans and computers	The plagiarism of ideas: the use of similar concepts or opinions outside the realm of common knowledge without due acknowledgement.
	The plagiarism of translated text: translations unsupported by acknowledgement of the original work.
	The production of text produced by an independent “ghostwriter”.
	Artistic plagiarism: the presentation of someone else's work in a different medium (the end result may involve text, images, voice or video).
	The structure of an argument in a source is copied without providing acknowledgments that the ‘systematic dependence on the citations’ was taken from a secondary source. This involves looking up references and following the structure of the secondary source.

Table 1. Five types of plagiarism.

Clearly, not all types of plagiarism are equally challenging for a computerized plagiarism detector. For example, verbatim copying of a text block (type 1) can be detected with a simple string matching routine. Paraphrasing (type 2) requires the use of natural language processing methods to reveal that both source and plagiarized texts contain the same assertions. Plagiarism of type 3 is technically easy to reveal, but surprisingly most current detection systems do not implement any counter-measures against these simple tricks (Kakkonen and Mozgovoy, 2010). “Tough plagiarism” (type 5) is especially difficult to detect, even for human experts. Some students may plagiarise unintentionally (e.g. by incorrectly referencing material taken from other sources), however, most students are aware that verbatim copying (e.g. copy-paste) constitutes plagiarism and such cases are often intentional.

Marshall and Garry (2005) conducted a survey to gather the perceptions of 181 students concerning what the students understand as plagiarism. They reported that 94% of the students identified scenarios describing verbatim plagiarism (type 1) such as “copying the words from another source without appropriate reference or acknowledgment”. The responses among students were, however, inconsistent regarding scenarios on how to correctly use materials from other sources. This included scenarios

on plagiarism of secondary sources (which involves referencing or quoting original sources of text taken from a secondary source without obtaining and looking up the original source), tough plagiarism (i.e. type 5 - copying the structure of an argument without providing acknowledgements), and paraphrasing (type 2), where 27%, 58%, and 62% of students correctly identified this as plagiarism respectively. Regardless of whether plagiarism was intentional or unintentional, or of the students' motivation to plagiarise, it is important for academics to catch cheating students and, most importantly, to educate those students on plagiarism in order to reduce the number of plagiarism occurrences.

The subsequent sections discuss promising approaches that could address the detection limitations of some of the plagiarism types that go beyond the capabilities of state-of-the-art detection systems.

2.2 Types of plagiarism detection systems

Plagiarism detection systems can be divided into *hermetic* and *web*, and into *general purpose*, *natural language* and *source code oriented*. Web detection systems try to find matches for the suspected document in online sources. Hermetic systems search for instances of plagiarism only within a local collection of documents. Such systems maintain a database of documents. The database may contain, for example, works submitted by other students and the lecture materials used in a particular course.

In case of web detection, wide coverage of accessible online documents is as an important feature as high-accuracy of the document comparison algorithm. Some of the existing web detection systems, such as Turnitin (iParadigms, 2010), also maintain extensive internal collections of documents, including student essays, electronic journals, etc. These systems, hence, are capable of both web and hermetic detection. This work, concentrates on document comparison methods, and hence, the problems related to organization and maintenance of large text databases are not considered relevant.

Some of the existing detection systems are capable of processing text documents of any nature (whether a computer program source code or a text composed in a natural language), and the term *generic detection system* refers to these type of systems. These systems are based on *string matching algorithms*. Being universal, such systems suffer from the lack of specialization, allowing the cheaters to use a wider range of effective plagiarism-hiding tricks.²

Let us consider how different plagiarism detection methods can address plagiarism type 2, paraphrasing (see Figure 1).

² For example, a typical method of concealing plagiarism in a source code of a computer program is to rename all variables and to substitute control structures with their equivalents (e.g. FOR-loops with WHILE-loops). Since this trick is source code-specific, most source code-oriented plagiarism detection systems are aware of it. In contrast, a generic detection algorithm would most likely be unable to overcome this plagiarism technique.

A	<u>I ate the pizza, the pasta, and the donuts.</u>
	<u>I ate spaghetti, the donuts, and the pizza.</u>
B	<u>I ate the pizza, the pasta, and the donuts.</u>
	<u>I ate spaghetti, the donuts, and the pizza.</u>
C	<u>I ate the pizza, the pasta, and the donuts.</u>
	<u>I ate spaghetti, the donuts, and the pizza.</u>
D	<u>I ate the pizza, the pasta, and the donuts.</u>
	<u>I ate spaghetti, the donuts, and the pizza.</u>

FIGURE 1. Detection results on a paraphrased sentence by four different methods of plagiarism detection.

Figure 1 illustrates results from comparing the original sentence “I ate the pizza, the pasta, and the donuts” to its paraphrased counterpart “I ate spaghetti, the donuts, and the pizza” when using four different types of text comparison methods, i.e., simple exact string matching (method A), advanced inexact string matching (method B), natural language parser based algorithm (method C), and natural language parser based algorithm combined with a thesaurus (method D). In Figure 1, words underlined by a solid or dashed line indicate words that have been detected by the comparison method. More specifically, words underlined by a solid line are those which occur in both sentences in an identical form (verbatim copy). Words underlined by a dashed line indicate detected synonymous words occurring in both sentences. Words which are not underlined are those which have not been detected by the particular detection method.

Method A corresponds to a simple string matching procedure, in which a detection algorithm tries to find exact matches between words and searches the input texts left-to-right. The advantage of this comparison method is its efficiency. On the other hand, this method only works reliably for detecting verbatim copying from a source text.

Method B occurs when a more advanced, inexact, string matching algorithm (such as *Running-Karp-Rabin Greedy-String-Tiling algorithm (RKR-GST)* (Wise, 1996)) is applied that allows partial matches. Such algorithms are able to find partial matches, even if they are scattered and do not form a continuous match. On the other hand, string matching algorithms do not take into account the structure of sentences, which can lead to false positive matches. Also, short matches between the two texts are often ignored (so

that the method does not mark every word that matches between two documents as plagiarized), which can distort the overall detection process.

Method C illustrates the usage of a natural language parser to aid text comparison. The sentences are first converted to parse trees (i.e. parsed). Next, words in the parsed sentences are sorted according to their dependency types or *grammatical relations* (GR) that designate the type of the dependency between the words (for example, subject, object, predicate etc.). The words inside each dependency or GR group are then sorted in alphabetical order. For example, Stanford Parser (Klein and Manning, 2003) produces the following parse tree for the example sentence:

[ate, cc[and], conj[donuts, pasta, pizza], det[the, the, the], dobj[pizza], nsubj[I]]

While “spaghetti” is not matched with “the pasta”, all the other words are found in both sentences. Using parsing as a preprocessing stage before the actual text comparison has the potential of allowing the detection of plagiarism in sentences in which the order of words and phrases has been modified. The drawback of the method is that parsing is a computationally complex task. Furthermore, while parsers exist for languages such as English, German, Chinese etc, they are not readily available for all natural languages.

Method D shows that the whole sentence can be matched if parsing is accompanied by a synonym thesaurus, which allows detecting “pasta” and “spaghetti” as synonyms. The major drawback of this matching method is that each language needs its own synonym list. Such lists are only readily available for a handful of languages of the World.

3 CURRENT STATE OF THE ART: AN OVERVIEW

While early plagiarism detection systems were only capable of detecting verbatim (copy-paste) copying, modern systems are able to reveal more advanced types of plagiarism. As demonstrated in the previous section, this capability can be achieved, for example, by employing an approximate string matching method, which finds a set of strings, belonging to both analyzed documents (a suspected file and its potential source). The same method also makes it possible to detect rearrangement of paragraphs and sentences. A recent study by Kakkonen and Mozgovoy (2010) showed that state-of-the-art plagiarism detection systems are insensitive to rearrangements of original document’s text blocks (i.e. structural changes).

Approximate string matching (method B above) also helps to fight against rewording: even if a fraction of words is substituted with synonyms, and the words in the sentence are rearranged, the system is likely to detect similarity between the documents. However, the similarity score in this case would typically be lower in comparison to a text in which verbatim copy-paste plagiarism was utilized. The reason for this is that a purely string matching based method is unable to treat synonymous words as matching pairs. Therefore, rewording and paraphrasing remain as challenges for plagiarism detection systems.

The evaluation by Kakkonen and Mozgovoy (2010) also revealed that state-of-the-art plagiarism detection systems do not have any protection against simple technical tricks (type 3), although these techniques are both easy to perform and easy to reveal. A

possible explanation to this is that many of the plagiarism detection software is created by system developers, and not by academics. System developers' may lack awareness of the various plagiarism techniques that students employ to disguise plagiarism when creating plagiarism detection software.

The methods listed as plagiarism of type 3 above are merely examples of what a plagiarizer can do in order to conceal plagiarism. It is not hard to invent other similar techniques, which obfuscate texts. All modern plagiarism detection systems should be able to reveal these basic types of tricks as they are the more frequently used by students to disguise plagiarism (Marshall and Garry, 2005); otherwise, the use of advanced document comparison algorithms makes little sense.

Our basic claim in this article is that the most fundamental reason for the shortcomings in the existing plagiarism detection systems is their heavy reliance on detection methods that are not based on processing natural languages, but rather on string matching which can only capture simple types of plagiarism. These methods run into problems when faced with complex types of paraphrasing (type 2 in our hierarchy) and they are, and will remain to be incapable of detecting tough plagiarism (type 5).

4 LEGAL AND ETHICAL ISSUES

The use of automatic plagiarism detection raises a number of ethical and legal problems (Foster, 2002; Glod, 2006). Generally, these problems fall into one of the two following categories:

1. Students complain about the low quality of plagiarism detection systems because some systems give rise to a large number of *false detections*. When false detections happen, the students concerned usually feel aggrieved since the software has unfairly marked their work as plagiarized. Students invest time and effort in producing their work and feel that they have been unfairly treated when, for various reasons, plagiarism detection systems report a number of instances of plagiarism in their submitted work.
2. Students object to submitting their essays to an online database because they assert that such an action *violates their intellectual property rights* and taints them with an unwarranted "presumption of guilt".³

The problems that arise in category (1) can be traced to a misunderstanding of what it is that an automatic plagiarism detection system is trying to achieve. Teachers and instructors should be quite clear that a software plagiarism detector should be used as an *auxiliary* tool — and not as a means for providing absolute proof of the existence of plagiarism in a text. It would be more accurate to describe the function of such software as a means for alerting a teacher or instructor to the *possibility* of plagiarism in a particular text. Since all software applications that scan text for dishonest practices are heuristic, and it is a teacher's ultimate responsibility to double-check any essay was great thoroughness before designating it as plagiarized.

³ This issue arises specifically with Turnitin as the system retains an internal database of student essays. See, for example, (Jones, 2007).

Thus, although educators may use computer-aided plagiarism detection tools at the detection stage, it should be kept in mind such tools detect *similarities* between students work which may (*suspicious similarities*) or may not constitute plagiarism (*innocent similarities*), and it is up to the user to judge whether suspicious plagiarism is the reason behind the similarity found in the detected documents. Thus, once similarity is detected, the teacher must go through the detected document pairs to identify and analyse matching text fragments. The next step is to determine whether the similarity between the documents is suspiciously high. Joy and Luck (1999) identify the issue of the burden of proof on gathering appropriate evidence for proving plagiarism: “*Not only do we need to detect instances of plagiarism, we must also be able to demonstrate beyond reasonable doubt that those instances are not chance similarities.*”

Furthermore, according to Hannabuss (2001) plagiarism is a difficult matter because “*evidence is not always factual, because plagiarism has a subjective dimension (i.e. what is a lot?), because defendants can argue that they have independently arrived at an idea or text, because intention to deceive is very hard to prove.*”

Suspected cases of plagiarism in which the original text cannot be found are the most difficult to prove, due to lack of evidence (Larkham and Manns, 2002; Joy and Luck, 1999). In addition, although educators may suspect plagiarism, searching for the original material and finding and collating enough evidence to convince the relevant academic panel in charge of dealing with plagiarism cases can be time consuming (Larkham and Manns, 2002). Finally, once evidence is collated, before a final decision is reached as to whether or not an instance of plagiarism has occurred, a typical process would be that the students involved are confronted with the evidence and only then a final decision is reached, as to whether the works in question contain plagiarism.

Possible responses to the problems that arise in category 2) above are still being heavily debated. The proponents of Turnitinstyle databases of student-authored texts argue, for example, that since the use of a plagiarism-checking system is categorically similar to sanctioning the presence of a referee in a football match, it cannot violate our customary understanding of a person’s presumption of innocence (Foster, 2002). In addition to this, the existence of online database of essays might be validly compared to what is routinely performed by Google’s cache service (a function that automatically collects and stores Internet pages). It is interesting to note in this regard that some recent lawsuits have confirmed Google’s assertion of “fair use” — findings that supports the legality and legitimacy of Internet caches (OUT-LAW News, 2006).

Posner (2007) has pointed out that while there is considerable overlap between the concepts of copyright infringement and plagiarism, they do not represent the same activity; not all plagiarism is copyright infringement and not all copyright infringement is plagiarism. The most important difference is that while copyright only protects the exact form in which ideas are expressed, the “stealing of ideas” more accurately constitutes plagiarism.

5 ADVANCING BEYOND THE STATE-OF-THE-ART

Section 5.1 explores the ways in which the detection of plagiarism types 1 to 3 could be made more accurate and less prone to false detections. Firstly, the use of natural language processing at the level of individual words and word phrases are analyzed. Secondly, it

considers possible approaches to various plagiarism detection problems, such as authorship attribution, which would allow a detection system to detect instances of plagiarism without knowing the exact source text. In addition, some future possibilities for automatically detecting instances of plagiarism type 4 (the inaccurate use of references) are outlined. Section 5.2, considers ways in which type 5 (tough plagiarism) could be detected.

5.1 Improving detection of plagiarism types 1, 2, 3 and 4

5.1.1 *Morphological Analysis and Syntactic Parsing*

Languages such as German, Russian, Japanese, and Finnish that permit a freer word order than, for example, English provide a set of problems that are not so pronounced when detecting plagiarism in languages with more stringent word order constraints. Languages with freer word order provide the plagiarist with means of concealing plagiarisms merely by changing the word order in sentences (plagiarism of type 2).¹ It is a feature of languages of this kind that they also exhibit a rich variety of possible word forms. This makes it even more difficult to detect plagiarism by simple word-to-word or string matching-based comparison methods. Fortunately, however, there are technical solutions that can circumvent the problems caused by the rich morphological possibilities of these languages. Morphological analyzers based on the two-level model that originated in the work of Koskenniemi (1984) and stemmers (such as, for example, Porter's stemmer (Porter, 1980)) are capable of removing suffixes and isolating the word stem for a given inflected word.

The use of syntactic parsers for detecting plagiarism regardless of word order variation was demonstrated in Section 2.2. Using a parser as a preprocessing stage is of a great importance for a detection system aimed at languages free word-order constraints. Such tools are, fortunately, becoming available for an increasing number of languages. A method of detecting instances of plagiarism in which "borrowing" has been concealed by the transposition of individual words, is described in the work of Mozgovoy et al. (2007). This method involves utilizing an existing natural language parser to convert sentences into parse trees with alphabetically sorted branches. Such operation maps into the same parse tree phrases that have been created by the transposition of words in such a way that the meaning is preserved. Once this has been done, the trees are then stored and compared by means of a conventional string matching based plagiarism detecting methods. A similar approach was proposed by Leung and Chan (2007).

5.1.2 *Use of synonym thesaurus*

An efficient method of comparing student texts can be implemented by making use of electronic thesauri. Thesauri are useful tools in the struggle against the substitution of synonymous words in student texts. The best-known example of a resource that offers this type of information is WordNet (Miller, 2010). As illustrated in Figure 1, a system utilizing synonym thesaurus identifies the set of words that are synonyms for a particular

word. It is necessary to use a thesaurus in tandem with word sense disambiguation modules in order to make sure that the set of synonyms that is being extracted is accurate and plausible (Mozgovoy et al., 2006; Leung and Chan, 2007).

5.1.3 Latent Semantic Analysis

The detection of tough plagiarism (type 5) and cases in which the original text has been reworded and paraphrased (type 2) requires a facility that is able to explicate the finest variations in words and sentences that are semantically similar. While plagiarism detection at the level of concepts and ideas is far beyond the limits of today's technologies, it is already possible to overcome certain types of semantic-preserving text alternations.

One of the most well known methods of comparing documents for semantic similarity is *Latent Semantic Analysis* (LSA). LSA is an intelligent document comparison technique that uses mathematical algorithms for analysing large corpora of text and revealing the underlying semantic information of documents (Deerwester *et al.*, 1990; Dumais, 1991). LSA has several characteristics that make it a feasible technique for plagiarism detection. It derives the relationship between synonymous words by analysing the context of word usage. Researchers have explored the level of meaning that LSA can extract from texts and their findings revealed that it can represent meaning from text as accurately as humans do, without the use of word order and syntax as required by humans (Landauer and Dumais, 1997; Landauer *et al.*, 1998; Rehder et al., 1998, Wolfe, et al., 1998).

The effectiveness of using LSA to plagiarism detection from student essays was demonstrated by the SAIF system (Britt *et al.*, 2004). SAIF compares pairs of student essays and considers those with similarity score higher than a given threshold to be possible instances of plagiarism. In Britt et al.'s experiment, SAIF was able to identify approximately 80% of texts that contained sentences that were plagiarised or quoted without the use of citation.

In programming assignments students may use various techniques for hiding plagiarism including verbatim copying, making changes to white space and formatting, renaming identifiers, reordering blocks of code and statements within code blocks, changing data types, adding redundant statements or variables, and replacing control structures with equivalent structures (Jones, 2001). Recent literature discusses the application of LSA for source-code plagiarism detection concerning files written in the Java programming language (Cosma and Joy, 2009a).

Some well-known string matching based systems including YAP3 (Yet Another Plague) (Wise, 1996), JPlag (Prechelt *et al.*, 2002), and Sherlock (Joy and Luck, 1999) hold two main limitations: firstly they often fail to detect similar files that contain significant code shuffling (Prechelt *et al.*, 2002) as they rely on detecting plagiarism by analyzing the structural characteristics of programs; and secondly they convert source-code files into tokens using a parser, which makes them programming language-dependent. The main advantages of LSA over such algorithms are that it does not make use of any thesauri to derive synonyms for a particular word, it is language-independent and therefore it does not require any parsers or compilers for programming languages in

order to provide detection in source-code files, as required by string-matching algorithms. Furthermore, because LSA ignores word order, if two documents are very similar but contain structural changes as an attempt to hide plagiarism, they are likely to be detected by LSA. LSA and string matching algorithms are sensitive to different types of attacks and overall plagiarism detection can improve when combining the two techniques (Cosma and Joy, 2009a).

Based on the literature, LSA appears as a suitable technique for detecting plagiarism types 1, 2, and 3 in both natural language and source-code text. The ability of LSA to identify similar or nearly identical documents that contain semantic changes (i.e. the replacement of words with synonyms or closely paraphrasing text) and structural changes makes LSA suitable for detecting plagiarism attacks of type 1 and 2. LSA can be effectively applied to detect type 3 plagiarism attacks if appropriate document pre-processing (i.e. corpus-preparation) takes place prior to its application.

Although LSA has proven to be a successful method for comparing documents in various applications, it is more effective in detecting instances of plagiarism when integrated with other detection algorithms (Cosma and Joy, 2009a). Furthermore, its capability in identifying the source of ideas and the authors of student writings has not been investigated in the literature. Whether or not LSA detects a similar file pair depends on the semantic analysis of words that make up each file, the mathematical analysis of the association between words, the corpus itself, and the choice of parameters which are not automatically adjustable but influence the behaviour of LSA (Cosma and Joy, 2009b). The fact that relations between terms are not explicitly modeled in the creation of the LSA space makes the behaviour of LSA unpredictable from the perspective of whether it can detect specific plagiarism attacks (Cosma and Joy, 2009a, b). Another limitation of the LSA algorithm for plagiarism detection lies in its incapability to accurately discover the pairs of matching text blocks. By using LSA, the teacher can only obtain overall document-document similarity scores, without specific indication as to which parts of the text are suspicious. Thus, combining LSA with morphological analyzers and syntactical parsers for capturing information about the structure of sentences and determining the similarity about the different parts of sentences is likely to improve the accuracy of the LSA technique for the task of plagiarism detection.

5.1.4 *“Fingerprinting” Authors*

The plagiarism detection systems discussed in the subsections above, access the source document from which the plagiarizer has sourced the text. Depending on the type of the system, the source documents are either received from Internet (web detection) or from a local database (hermetic detection). It is, however, unrealistic to expect that the local database, or even the Internet, contains all possible source documents that a plagiarizer could have used. There exists no Web search engine that would be able to scan the whole Internet. Hence, the assumption of always having access to the source document is unrealistic, especially when cross-language plagiarism and legal and ethical issues (see Section 7) involved in marinating local document collections are concerned. Therefore, methods that can detect probable instances of plagiarism without having to analyze its potential sources arouse special interest.

Running head: Automatic student plagiarism detection: future perspectives

With current *authorship detection* methods, such as those of Diederich et al. (2003) and Putninš et al. (2005), it is possible to create a “fingerprint” of a particular writer on the basis of his or her idiosyncratic vocabulary, syntax and writing style. Such profiles can then be used to identify the author of a text. These methods are currently able to detect authors from a restricted, predefined set of authors only. The methods also require that a “fingerprint” be made of each student’s style before the system is put into action. It would also be possible to determine that two given blocks of text had been composed by two different authors without any explicit attribution of authorship. The smallest amount of continuous text written by a single author should consist of at least 1000 words before it can be reliably attributed with the existing methods. Furthermore, to build an author’s profile that adequately represents his or her stylistic idiosyncrasies, around ten different texts are needed (Stamatatos et al., 1999).

While authorship attribution has not been applied to plagiarism detection so far, forensics is a commonly mentioned area of application for these methods. For example, the work by de Vel et al. (2001) is concentrated on identification of the author of a particular e-mail message by analyzing various message attributes (average word and sentence length, the presence and type of greeting and farewell clauses, the proportion of lowercase and uppercase letters, etc.). The paper (Chaski, 2005) discusses the use of more advanced stylistic attributes, such as punctuation, syntactic, and lexical marks. The method, described in the work is claimed to have 95% detection accuracy, and was used in actual lawsuits to support gathered evidence. Based on these encouraging examples, using authorship attributions methods in plagiarism detection appears to be feasible.

5.1.5 Reference and Citation Tracking

In order to detect the plagiarisms of type 4, namely, the deliberately inaccurate use of references, one needs to have an automatic method of detecting citations and references from texts. “Reference and citation tracking” refers to the process of automatically detecting the citations (abbreviated expressions embedded in the text) and references (information on the author and the publication title and date) in a document. It functions by detecting all the references in a particular document and then matching each individual citation in the text to the relevant reference from that text. Most of the work on reference and citation tracking, such as that undertaken by, for example, Teufel and Moens (2000), describes methods for tracking references and citations in scientific literature. It is, in many ways, a quite straightforward procedure to match a reference index and scientific texts because the reference formats in which they appear have been more or less standardized by scholars throughout the world.

A recent review of the literature revealed that no attempts have been made to apply existing citation and reference tracking methods for detecting plagiarism in student texts. This line of research could provide interesting results. There, however, are some great challenges. One might hope that a text produced by a student would closely resemble the kind of text produced by an experienced scientist. The sad reality, however, is that the referencing and citation styles of most students leave a lot to be desired. Hence, it seems reasonable to assume that the existing tracking methods should be considerably modified before they would be ready to be applied in student plagiarism detection.

5.2 Detecting tough plagiarism: The Problem of Stealing Ideas, Ghost Writers and Cross-language Plagiarism

The detection of type 5 plagiarism (tough plagiarism) represents a problem whose solution remains beyond the capabilities of existing text analysis methods and that it will remain so for foreseeable future. The use of translated texts has been categorized as one of the most difficult forms of plagiarism to deal with. Fortunately, the sheer amount of work and time consumed by manual translation somewhat limits the popularity of cross-language plagiarism. There are, in addition, some indications that translation plagiarism might be detected automatically with some degree of reliability in the foreseeable future by using *machine translation* (MT) systems. While the general quality of MT is still quite poor (Koehn and Monz, 2006), it may be of a sufficient standard for the purposes of detecting plagiarism. A computer can, for example, translate a document into the language of the locally stored document collection and prepare an “image” of the document that reflects its vocabulary and statistical measures. Such an image would not include most of the errors made by an MT system — errors that arise out of incorrect sentence structure and the incorrect use of prepositions and cases. Once this has happened, the image can be used in a document-document comparison mechanism. There are in fact several plagiarism detection systems that make use of such images in document comparison (Schleimer et al. 2003, Nakov, 2000, Stein et al. 2006). A straightforward MT routine, based on a multilingual EuroWordNet dictionary (University of Amsterdam, 2010), was applied to plagiarism detection by Ceska et al. (2008). The authors consider their results as “promising” and continue working in this direction. Cross-language plagiarism problem still remains far from being satisfactorily solved.

The stealing of ideas is probably the most difficult type of plagiarism to detect, both for human beings and computers. The detection of this type of plagiarism would without doubt require extremely precise techniques of conceptualizing and representing ideas and the development of a reliable method for extracting such constructions from texts. There is no reason to believe that such analyses could be carried automatically in the foreseeable future.

The detection of ghostwriters represents another type of plagiarism that is beyond the capabilities of existing plagiarism detection systems. The fingerprinting methods discussed above might eventually indicate the direction in which a solution to this problem will be found. Fingerprinting techniques are still far too primitive to provide a basis for researchers to develop systems that will be able to identify ghost writing in practice. But plagiarism is a complex phenomenon, and computer-aided detection is not the only means for combating cheating. The issue of ghostwriting, for example, has already been addressed in various legal actions (Zobel, 2004).

6 CONCLUSION

Student plagiarism is a complex phenomenon. One anti-plagiarism measure consists of developing computer-aided plagiarism detection instruments. These tools have evolved over the last two decades from simple text-matching programs into powerful tools capable of detecting partial and disjoint blocks of “borrowed” text. However, they are still unable to detect various plagiarism hiding tricks, ranging from simple text

Running head: Automatic student plagiarism detection: future perspectives

manipulations, exploiting detectors' weaknesses to extensive rewording, paraphrasing, and translation of source documents.

Fortunately, today's natural language processing technologies are capable of advancing state-of-the-art in the field of software-aided plagiarism detection. Such tools as syntactic and semantic parsers, morphological analyzers, topic modeling, LSA, citation tracking, and authorship attribution have a potential to become the corner-stones of the next-generation of automated plagiarism detection systems. This claim is supported with a number of published and ongoing research projects that have been reviewed in this article.

Growing quality of computerized plagiarism detectors increases their popularity, which raises non-technical debates about legal and ethical issues that are related to the use of such tools. While it is easy to understand the concerns caused by improper use of detectors, all legal and ethical questions can be addressed in the future.

REFERENCES

ACNP Software. *AntiPlagiarist*. Retrieved February 22nd, 2010, from

<http://www.anticutandpaste.com/antiplagiarist/>.

Ahtiainen, A., Surakka, S., Rahikainen, M. (2006). Plaggie: GNU-Licensed Source Code Plagiarism Detection Engine for Java Exercises. *Proceedings of the 6th Baltic Sea Conference on Computing Education Research, Koli, Finland*.

Bennett R. Factors associated with student plagiarism in a post-1992 University. *Journal of Assessment and Evaluation in Higher Education*, 30(2):137–162, 2005.

Bloomfield, L. A. *Software to Detect Plagiarism: WCopyfind (Version 2.6)*. Retrieved February 22nd, 2010, from

<http://www.plagiarism.phys.virginia.edu/Wsoftware.html>.

Britt, A., Wiemer-Hastings, P., Larson, A., and Perfetti C. (2004). Using intelligent feedback to improve sourcing and integration in students' essays. *International Journal of Artificial Intelligence in Education*, 14:359–374.

Canexus Inc. *EVE2- Essay Verification Engine*. Retrieved February 22nd, 2010, from

<http://www.canexus.com/>.

Running head: Automatic student plagiarism detection: future perspectives

- Ceska, Z., Toman, M., Jezek, K. (2008). Multilingual Plagiarism Detection. *Lecture Notes in Computer Science*, p. 83-92,
- Chaski, C. (2005). Who's At the Keyboard? Authorship Attribution in Digital Evidence Investigations. *International Journal of Digital Evidence*, vol. 4(1), p. 1-13.
- Clough, P. (2000). *Plagiarism in Natural and Programming Languages: an Overview of Current Tools and Technologies*. Internal Report CS-00-05, University of Sheffield, UK.
- Cosma, G. and Joy M. (2008), Towards a Definition of Source-Code Plagiarism, *IEEE Transactions On Education*, 51(2), 195-200.
- Cosma, G. and Joy, M. (2009a). An Approach to Source-code Plagiarism Detection and Investigation Using Latent Semantic Analysis. *IEEE Transactions On Computing*.
To appear.
- Cosma, G. and Joy, M. (2009b). Parameters Driving the Performance of LSA for Source-code Similarity Detection. Under Review.
- Dick M., Sheard J., Bareiss C., Carter J., Harding T., Joyce D., and Laxer C. Addressing student cheating: definitions and solution. *SIGCSE Bulletin*, 35(2):172–184, 2003.
- Deerwester, S., Dumais, S., Landauer, T., Furnas, G., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science* 41, 6, 391–407.
- Dumais, S. (1991). Improving the Retrieval of Information from External Sources. *Behavior Research Methods, Instruments and Computers*, vol. 23:2, 229–236.
- Diederich, J., Kindermann, J., Leopold, E.: Paass, G. (2003). Authorship Attribution with Support Vector Machines. *Applied Intelligence*, 19(1-2): 109-123.

Running head: Automatic student plagiarism detection: future perspectives

- Foster, A. (2002). Plagiarism-Detection Tool Creates Legal Quandary. *Chronicle of Higher Education*, May 17th, 2002, Section: Information Technology, A37.
- Jones, E. (2001). Metrics based plagiarism monitoring. *Journal of Computing Sciences in Colleges*, 16(4):253–261.
- Glod, M. (2006). Students Rebel against Database Designed to Thwart Plagiarists. [Electronic version]. *Washington Post*, September 22nd, 2002, p. A01.
- Hannabuss, S. (2001). Contested Texts: Issues of Plagiarism. *Library Management*, 22(6/7):311- 318.
- iParadigms. *Turnitin.com. Digital assessment suite*. Retrieved February 22nd, 2010, from <http://turnitin.com>.
- Jones, K. C. (2007). Students Sue Anti-plagiarism Service for Copyright Infringement. *InformationWeek*, April 3th. Retrieved April 26th, 2010, from <http://www.informationweek.com/news/internet/showArticle.jhtml?articleID=198702230>.
- Joy, M. and Luck, M. (1999). Plagiarism in Programming Assignments. *IEEE Transactions on Education*, 42(2): 129-133.
- Kakkonen, T. and Myller, N. (2009) AntiPlag - A Sampling-based Tool for Plagiarism Detection in Student Texts. *Proceedings of the 8th European Conference on e-Learning*, Bari, Italy, 2009.
- Kakkonen, T. and Mozgovoy, M. (2010). Hermetic and Web Plagiarism Detection Systems for Student Essays - An evaluation of the state-of-the-art. *Journal of Educational Computing Research*, 42(2):135-139.

Running head: Automatic student plagiarism detection: future perspectives

- Kasprzak, J. and Nixon, M. (2004). Cheating in Cyberspace: Maintaining Quality in Online Education. *Association for the Advancement of Computing In Education*, 12(1):85– 99.
- Karttunen, L. and Martin K. (1985). Parsing in a Free Word Order Language. In Dowty, D., Karttunen, L. and Zwicky, A. (eds.) *Natural Language Parsing*. Cambridge, Cambridge University Press, U.K.
- Klein, D. and Manning, C. (2003). Accurate Unlexicalized Parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423-430.
- Koehn, P. and Monz, C. (2004). Manual and Automatic Evaluation of Machine Translation between European Languages. *Proceedings of the Workshop on Statistical Machine Translation*, New York, USA, pp. 102-121.
- Koskenniemi, K. (1984). A General Computational Model for Word-form Recognition and Production. *Proceedings of the 22nd Conference on Association for Computational Linguistics*. Stanford, California, USA.
- Landauer, T. and Dumais, S. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, 104(2):211–240.
- Landauer T., Laham D., and Foltz P. (1998). Learning Human-like Knowledge by Singular Value Decomposition: A Progress Report. In *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, Massachusetts, USA.

Running head: Automatic student plagiarism detection: future perspectives

- Landauer, T. and Psozka, J. (2004). Simulating Text Understanding for Educational Applications with Latent Semantic Analysis: Introduction to LSA. *Interactive Learning Environments*, **8**(2): 72-86.
- Lathrop, A., Foss, K. (2000). *Student cheating and plagiarism in the Internet era. A wake-up call*. Englewood, Colorado, USA: Libraries Unlimited. Lancaster, T. and Culwin, F. (2004). Using Freely Available Tools to Produce a Partially Automated Plagiarism Detection Process. *Proceedings of the 21st ASCILITE Conference, Perth, Australia*.
- Larkham P. and Manns S. (2002). Plagiarism and its Treatment in Higher Education. *Journal of Further and Higher Education*, 26(4):339–349.
- Leung, C.-H. and Chang, Y.-Y. (2007). A Natural Language Processing Approach to Automatic Plagiarism Detection. *Proceedings of the 8th ACM SIGITE conference on Information technology education*, p. 213-218.
- Maurer, H., Kappe, F., Zaka B. (2006). Plagiarism — a Survey. *Journal of Universal Computer Science*, 12(8): 1050-1083.
- Mediaphor Software Entertainment AG. *Plagiarism-Finder*. Retrieved February 22nd, 2010, from <http://www.m4-software.com/>.
- Miller, G. A. (2010) WordNet. Princeton University. Retrieved February 22nd, 2010, from <http://wordnet.princeton.edu>.
- Mozgovoy, M., Tusov, V., Klyuev, V. (2006). The Use of Machine Semantic Analysis in Plagiarism Detection. *Proceedings of the 9th International Conference on Humans and Computers*, Aizu-Wakamatsu, Japan, p. 72-77.

Running head: Automatic student plagiarism detection: future perspectives

- Mozgovoy, M., Kakkonen, T., Sutinen, E. (2007). Using Natural Language Parsers in Plagiarism Detection. *Proceedings of SLaTE'07 Workshop*.
- Myers, S. (1998). Questioning author(ity): ESL/EFL, Science, and Teaching about Plagiarism. *Teaching English as a Second or Foreign Language (TESL-EJ)*, 3(2):11–20.
- Nadelson, S. (2007). Academic Misconduct by University Students: Faculty Perceptions and Responses. *Plagiary*, 2(2):1–10.
- Nakov, P. (2000): Latent Semantic Analysis of Textual Data. *Proceedings of the Conference on Computer Systems and Technologies*. Sofia, Bulgaria.
- OUT-LAW News (2006). Google Cache does not Breach Copyright, Says Court. Retrieved February 22nd, 2010, from <http://www.out-law.com/page-6571>
- Porter, M. F. (1980). An Algorithm for Suffix Stripping, *Program*, 14(3): 130–137.
- Posner, R. A. (2007). *The Little Book of Plagiarism*. New York, New York, USA: Pantheon Books.
- Prechelt, L., Malpohl, G., Philippsen, M. (2002). Finding Plagiarisms among a Set of Programs with JPlag. *Journal of Universal Computer Science*, 8(11): 1016-1038.
- Putninš, T., Signoriello, D. J, Jain, S., Berryman, M. J., Abbott, D. (2005). Advanced Text Authorship Detection Methods and Their Application to Biblical Texts. *Proceedings of the SPIE*, Brisbane, Australia.
- Rehder, B., Schreiner, M., Wolfe, M., Lahaml, D., Kintsch, W., and Landauer, T. (1998) Using Latent Semantic Analysis to Assess Knowledge: Some Technical Considerations. *Discourse Processes*, 25:337–354.

Running head: Automatic student plagiarism detection: future perspectives

- Scanlon, P. and Neumann, D. (2002). Internet Plagiarism among College Students. *Journal of College Student Development*, 43(3):374–85.
- Schleimer, S., Wilkerson, D. S., and Aiken, A. (2003). Winnowing: Local Algorithms for Document Fingerprinting, Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, pp. 76-85.
- Sciworth Inc.: *MyDropBox*. Retrieved February 22nd, 2010, from <http://www.mydropbox.com/>.
- SeeSources.com: *Instant, Automatic & Free Text Analysis*. Retrieved February 22nd, 2010, from <http://seesources.com/>.
- Stamatatos, E., Fakotakis, N., Kokkinakis, G. (1999). Automatic Authorship Attribution. *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*. Bergen, Norway.
- Stein, B., zu Eissen, S. M. (2006). Near Similarity Search and Plagiarism Analysis. *Selected Papers from the 29th Annual Conference of the German Classification Society*. Magdeburg, Germany.
- Teufel, S., Moens, M. (2000). What's Yours and What's Mine: Determining Intellectual Attribution in Scientific Text. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. Hong Kong.
- University of Amsterdam: *WordNet*. Retrieved February 22nd, 2010, <http://www.illc.uva.nl/EuroWordNet/>.
- de Vel, O., Anderson, A., Corney, M., Mohay, G. (2001). Mining E-mail Content for Author Identification Forensics. *ACM SIGMOD*, vol. 30(4), p. 55-64.

Running head: Automatic student plagiarism detection: future perspectives

Wise, M. (1996). YAP3: Improved Detection of Similarities in Computer Program and Other Texts. *SIGCSE Bulletin*, 28(1):130–134.

Wolfe, M., Schreiner, M., Rehder, R., Laham, D., Foltz, P., Landauer, T., and Kintsch, W. (1998). Learning from Text: Matching Reader and Text by Latent Semantic Analysis. *Discourse Processes*, 25:309–336.

Zobel, J. (2004). "Uni Cheats Racket": A Case Study in Plagiarism Investigation. *Proceedings of the 6th Conference on Australasian Computing Education*.
Dunedin, New Zealand.

ⁱ In languages that permit more freedom in word order than, for example, English, the function of a particular word in a sequence of words can be understood without any additional reference to its particular position in a sentence. For instance, all possible six permutations of the three words *hän* “he” (sg nom), *söi* “eat” (past sg 3rd), and *kalan* “fish” (sg acc) produce grammatically correct sentences in Finnish and express the same meaning (Karttunen and Kay, 1985).