TURNITOFF – DEFEATING PLAGIARISM DETECTION SYSTEMS

Lee Gillam
Department of Computing
University of Surrey
Guildford, Surrey, UK
l.gillam@surrey.ac.uk
http://tinyurl.com/leegillam

John Marinuzzi
Department of Computing
University of Surrey
Guildford, Surrey, UK
jm00217@surrey.ac.uk
http://www.cs.surrey.ac.uk

Paris Ioannou
Department of Computing
University of Surrey
Guildford, Surrey, UK
pi00002@surrey.ac.uk
http://www.cs.surrey.ac.uk

ABSTRACT

Defeating plagiarism detection systems involves determining effective approaches for greatest impact at lowest cost with the least likelihood of detection. Relatively simple techniques have been applied elsewhere for avoiding plagiarism detection, demonstrated at the last HEA-ICS conference. In this paper, we discuss defeats for seven plagiarism detection systems, including Essayrater, Seesources, PlagiarismDetector, and the popular Turnitin. We report on initial results of human experiments undertaken on visual similarity to assess the risk of human detection of changes. The systems evaluated are variously susceptible to sufficient numbers of small alterations to characters or words in the text Our results suggest, at minimum, to use at least 2 such systems in combination to reduce the likelihood of failed detection and increase the difficulty for the determined, and yet somehow lazy, plagiarist – otherwise, the discovery and dissemination of simple defeats for plagiarism detection software may mean that we may as well just "Turnitoff".

Keywords

Plagiarism, detection, defeat, visual similarity, risk.

1. INTRODUCTION

Plagiarism detection becomes a key distraction from assessing written work, with an inherent cost in investigating suspicious material. While we actively and strongly discourage "cut+paste" or buying off-the-shelf, pressure to perform makes such strategies a likely last resort. To help identify suspicious writing, student work may be systematically run through plagiarism detection systems, which attracts legal and ethical discussion, with outputs used as an indication of need for further (human) investigation. Here, there is a danger of becoming reliant on the system. However, those knowledgeable of detection strategies may also become adept at avoiding risk of detection by suitable adaptation of the sourced material, bringing the material below any arbitrary threshold of suspicion. Techniques such as essay spinning [4] may be effective if the machine translation system produces results that are sufficiently divergent from the original – there may be an advantage to using machine translation systems that produce lower quality results. However, subsequent efforts required to rewrite such spun material may be significant, and may also introduce the risk that the rewrite lowers the required divergence. Techniques requiring less subsequent effort involve thesaural substitution and character substitution [2] and can be highly effective against plagiarism detection systems because of algorithms that are either highly sensitive to very minor variations, or rely on a specific string span.

In this paper, we discuss simple techniques of character and word substitutions that we have used, following on from [2] and [4], to variously defeat a reasonable number of plagiarism detection systems. The systems evaluated are Plagium, Seesources, PlagiarismDetector, Plagiarism Checker, EssayRater, Plagiarism Detect and Turnitin. As systematic changes should also attempt to avoid suspicions being raised on reading, we have also considered that visual (and semantic) similarity should be retained in character and word changes, and in this paper we assess the risk of human detection in relation to character changes; semantic similarity is beyond the scope of this paper. Visual similarity can also be an issue for both website domain names [6] and webpage/site phishing [5]. Part of our work involves developing a prototype to automate the generation of test texts that will suggest the extent of changes required to pass a detection threshold with a known risk of detection in order to make our experiments more systematic. We have identified systems that are not defeated in the same way as the apparently de facto Turnitin, so some of these might be used in parallel with Turnitin in order to improve the likelihood of plagiarism detection. Consequently, those wishing to risk plagiarism and attempting to avoid detection might find the efforts required to defeat such systems becoming increasingly costly.

2. PLAGIARISM IN THE WILD

There have been a variety of high profile cases of plagiarism in recent years. The Downing Street "dodgy dossier" is perhaps one of the most significant. The dossier was highly similar to an article by Ibrahim Al Marashi, "Iraq's Security and Intelligence Network: A Guide and Analysis" published in September 2002 by the Middle East Review of International Affairs (MERIA), which the journal editor has suggested, possibly on reflection, endorses the quality of their published articles[1]. An example of the similarity is shown in Table 1[2] with differences identified in bold; even "grammatical mistakes made on the internet version ended up in this February 2003 document"[3].The case is further discussed on the FamousPlagiarists.com / WarOnPlagiarism.org website, amongst others.

Marashi	British Intelligence
Its internal activities include:	Its internal activities include:
Spying within the Ba'th Party, as well as other political parties;	**monitoring** the Ba'th Party, as well as other political parties
	monitoring other grass roots
Suppressing Shi'a, Kurdish and other opposition	suppressing Shi'a, Kurdish and other opposition
counter-espionage	counter-espionage
targeting threatening individuals and groups inside Iraq	targeting threatening individuals and groups inside **of** Iraq
spying on foreign embassies in Iraq **and** foreigners in Iraq	**monitoring** foreign embassies in Iraq
	monitoring foreigners in Iraq
maintaining an internal network of informants	maintaining an internal network of informants

Table 1: Similarities and differences (in bold) for fragments from the Marashi article and the dodgy dossier

The FamousPlagiarists.com / WarOnPlagiarism.org website features further alleged, and proven, cases of plagiarism in a variety of professions; amongst these are, in no particular order, Al Gore, Martin Luther King, Osama Bin Laden, J.K. Rowling, Madonna and Britney Spears. The site also includes an unfortunate number of academics[4]. Most notable of these is Bindu Ganga, who was *eventually* sacked by her University (Argosy) and stripped of her (plagiarized) doctorate[5] but was allowed to submit a new project for, and be awarded, another doctorate. This case is notable because Argosy initially defended Ganga against accusations made by Argosy student Marla Decker, who was "disciplined, in part because she pushed the charge"[6] until her allegations of plagiarism were substantiated by a Turnitin originality report.

While plagiarism cases are typically focused on quantities of duplicated text, there is an intriguing case of a phantom paper [3]. Here, there have been many citations by academics and practitioners of a supposed landmark reference in information retrieval, but never been an article with that particular title. At minimum, this suggests that referring to spurious articles will help to identify the diligent researchers.

In part, the three cases discussed above set unfortunate precedents. Acts of plagiarism that appear acceptable and go unchallenged, or are defended at senior levels, and may even be described as unintentional/mistakes[7] with second chances offered, make penalties for small or moderate quantities of plagiarism in educational settings harder to justify.

2.1 A note on why plagiarism detection systems can fail

Plagiarism detection systems typically rely on the selection of (a specific set of) strings of a particular length from a source document, and the relationship of these strings to a set of target documents. Strings may be selected as a number of words or number of characters, and be sampled from the source document or taken in various increments across the entire document. Frequent occurrences of similar patterns in target documents adds to the indication of suspicion. These strings may be hashed, and hash collisions used to indicate similarities; low probability of collision is required to avoid false positives.

---

[1] MERIA Editor's response posted at: http://meria.idc.ac.il/british-govt-plagiarizes-meria.html (Accessed 20 May 2010)

[2] Tony Blair, Colin Powell and the Case of the "Sexed Up" British Intelligence Dossier - A Linguistic Analysis by Dr. John P. Lesko: http://www.famousplagiarists.com/MLSsexedupdossier.ppt (Accessed 20 May 2010)

[3] See footnote above.

[4] http://waronplagiarism.org/academia.htm (Accessed 20 May 2010)

[5] See, also, the (somewhat similar) articles at http://www.ask.com/wiki/Argosy_University#Plagiarism_controversy and http://en.wikipedia.org/wiki/Argosy_University#Plagiarism_controversy (Accessed 20 May 2010)

[6] http://www.ibhe.state.il.us/newsdigest/newsweekly/010408.pdf, p18, (Accessed 20 May 2010)

[7] "Ganga admits there were many mistakes in her paper. [..] she said her errors were "unintentional" and claims she shouldn't be seen in the same light as plagiarists who take credit for the work of others" - http://www.ibhe.state.il.us/newsdigest/newsweekly/010408.pdf, p17, (Accessed 20 May 2010)

Requiring exact string matching, rather than computing string distance, will immediately be susceptible to character or word variations within the string. Such variations would carry through to use of hashes. Modifications within a sufficient number of such strings maybe enough to avoid detection, and the longer the string, the fewer changes required to the text as a whole. Word changes, insertions, or deletions, can be detected if using overlapping strings ("shingling"). Shorter string lengths, smaller increments for shingling, and producing patterns against the entire text will all tend to favour detection; consequently, systems tuned for efficiency by using larger strings, fewer samples, and small or no overlaps, are likely to be less accurate.

2.2 Defeating plagiarism software?

There appear to be relatively few software applications directly "marketed" at defeating plagiarism detection systems. However, searching for "article rewriter" reveals a number of software applications[8,9]. A notable free piece of software which undertakes the latter, but with a title towards the former, is the "Anti-anti-plagiarism detection system" (AAPS)[10], a Perl script that contains a number of substitutions (and their reverse) intended to maintain "grammer [sic.] and spelling". A sample of AAPS substitutions is shown in Table 2. The substitutions do not necessarily have the coverage required to make a significant impact on a document: run against the "dodgy dossier", the full text of which already had several 100% hits at a number of Universities in Turnitin, the reduction was a mere 2%. The extent of impact possible due to other article rewriters remains to be evaluated. However, replacement by alternative phrases, use of "intentional" mistakes, systematic use of a thesaurus, and similar resources[11] could increase the degree of change substantially. The key to this, with reference to the section above, is knowing how many changes to make, and where.

Phrase	Replacement	Phrase	Replacement
in other words	alternatively	then there is	next comes
a plethora of	many	took part in	participated in

Table 2: Example replacements in AAPS

3. Experiments in Defeating Plagiarism Detection Systems

Three experiments were undertaken. The first involving character substitutions within Unicode, the second using thesaural substitutions, and the third assessing the chances of (human) visual detection of character substitutions. For the first and second experiments, a 266 word text[12] was used; for the third, we made use of 20 intelligent humans for segments of a relevant Internet Engineering Task Force (IETF) document[13].

3.1 Experiment 1: Character Substitutions

A set of character substitution tests was formulated around changing, for example, Latin "e", with code U+0065 to Cyrillic "Ie" with code U+0435. These characters look incredibly similar, even at 20pt font (Table 3).

Latin e (U+0065)	Cyrillic e (U+0435)	Latin a (U+0061)	Cyrillic a (U+0430)	Latin o (U+006F)	Greek o (U+03BF)
e	e	a	a	o	o

Table 3: Character similarities in Unicode

The tests were:

- **Test 1:** Change Latin o,c,p,y,a,e to similar Cyrillic and Greek letters;
- **Test 2:** Change Latin i and v to similar Vav and Greek letters;
- **Test 3:** Test 1 plus change Latin A,B,C,E,H,I,J,K,M,N,O,P,T,Y to similar Greek and Cyrillic letters;

Results for the seven systems show six are defeated by these changes (Table 4). Plagiarism Detect appears not to recognize plagiarism for the original article, suggesting its database does not include this. The only

[8] The free Article Changer: http://www.articlechanger.net/ (Accessed 20 May 2010)
[9] The Best Spinner, priced at US$7: http://thebestspinner.com/ (Accessed 20 May 2010)
[10] http://sourceforge.net/projects/aaps (Accessed 20 May 2010)
[11] For example, the Plain English Campaign's A to Z of Alternative Words http://www.plainenglish.co.uk/files/alternative.pdf (Accessed 20 May 2010)
[12] 266 words of "History of London, History of England a unique and stimulating site". http://www.historyofengland.net/content/view/119/49/, (Accessed 20 May 2010), starting at "London 1500 Years Ago" and ending "made London their winter HQ".
[13] Request for Comments (RFC) 4690 on Internationalized Domain Names (IDNs), available at: http://www.rfc-archive.org/getrfc.php?rfc=4690 (Accessed 20 May 2010).

system that appears to perform well is Turnitin, though Test 2 suggests it may be possible to push a text below a threshold if it were only a number of percentage points above it.

	Plagium	Seesources	Plagiarism Detector	Plagiarism Checker	EssayRater	Turnitin	Plagiarism Detect
No change	100%	100%	100%	100%	100%	100%	8%
Test 1	0%	0%	0%	0%	0%	100%	0%
Test 2	0%	0%	0%	0%	0%	60%	0%
Test 3	0%	0%	0%	0%	0%	99%	0%

Table 4: Plagiarism Detection Systems Tests 1 to 3

3.2 Experiment 2: Word Substitutions

Words were manually substituted using the most suitable synonyms at every 5th (Test 4), 6th (Test 5), 7th (Test 6), and 8th (Test 7) word of a sentence using a combination of thesaurus.com and Microsoft Word. Such a process could be automated, to an extent, using an electronic resource such as Wordnet[14]; post-editing may still be needed to retain meaning. Results for the seven systems show five are defeated by changes at the 5th word, though two recover at the 6th word, and two systems remain defeated by substitutions at 8 words (Table 5) - Plagiarism Detect was starting from a low threshold anyway. These results show the likely string span used by the systems that recover from the defeat. Interestingly, Seesources appears to be unaffected by these changes, while PlagiarismDetector has a change in performance relative to length.

	Plagium	Seesources	Plagiarism Detector	Plagiarism Checker	EssayRater	Turnitin	Plagiarism Detect
Test 4	0%	100%	85%	0%	0%	0%	0%
Test 5	0%	100%	93%	0%	100%	0%	0%
Test 6	0%	100%	98%	0%	100%	0%	0%
Test 7	0%	100%	98%	100%	100%	87%	0%

Table 5: Plagiarism Detection Systems Tests 4 to 7

3.3 Experiment 3: Visual Similarity

A small crowdsourcing experiment was undertaken with a group of 20 undergraduate students, predominantly from the Department of Computing, who were each briefed to detect as many changes as they could to a set of paragraphs and given both the original and a revised version containing zero or more character substitutions. For a set of substitutions, intuitively selected, we were looking for approximate detectability for an informed, but moderately motivated, audience.

Replacement letters	e - е	h - һ	v - ѵ	l - ӏ	u - υ	i - í	p - ρ	k - κ
Found	0/20	0/20	3/20	4/20	6/20	9/20	12/20	14/20
Risk of detection	0%	0%	15%	20%	30%	45%	60%	70%

Table 6: Risk of detection based on rates of discovery for visually similar characters

To demonstrate these replacements, the first four, with up to 20% risk, have been applied to a segment of the Marashi article - 16 replacements of “e”, 2 of “h”, 2 of “v” and 4 of “l” (Table 7). Visually, these appear highly similar, and this is also likely to pass detection by most of the systems tested in this paper.

Marashi (original)	Marashi (with substitutions)
Part Two gives up to date details of Iraq’s network of intelligence and security organisations whose job it is to keep Saddam and his regime in power, and to prevent the international community from disarming Iraq.	Part Two gives up to date details of Iraq’s network of intelligence and security organisations whose job it is to keep Saddam and his regime in power, and to prevent the international community from disarming Iraq.

Table 7: A segment of the Marashi article with, and without, character substitutions.

For a small collection of recently published news articles[15], we assessed the average number of changes attributable to these substitutions to determine the average impact on a document of making such changes, and hence to gain a broad understanding of risk versus alteration (Figure 1). Over 50% of words are changed in an apparently risk-free manner by the first two substitutions. We are initially assuming 100% plagiarism, so

[14] http://wordnet.princeton.edu/wordnet/download/ (Accessed 20 May 2010)
[15] 8 texts taken from BBC News, e.g. http://news.bbc.co.uk/1/hi/world/asia-pacific/8681833.stm

amounts already at a distance below this might well be brought under any threshold for suspicion. A similar experiment could be carried out relating to the retention of meaning following thesaural substitutions.

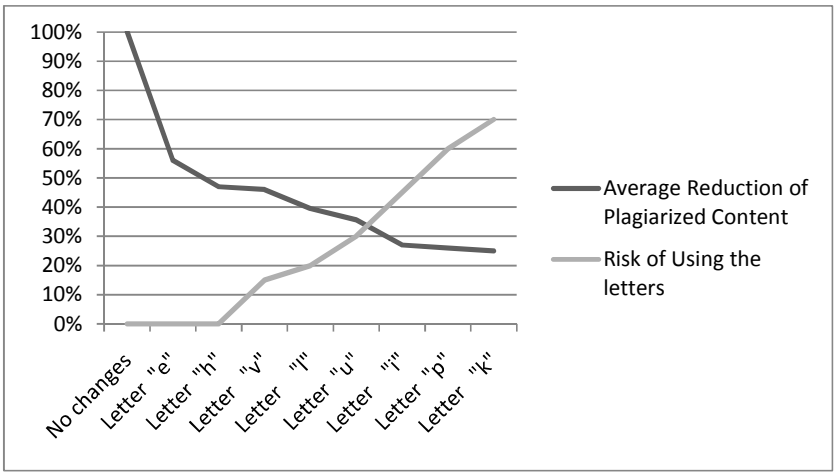

Figure 1: Risk of detection of substitutions versus proportion of words altered in the text (n=8).

4. CONCLUSIONS AND REMARKS

The experiments discussed demonstrate how the determined, but lazy, plagiarist could learn to avoid detection. Most of the systems evaluated are susceptible either to character substitutions, or to systematic word substitutions; some are susceptible to both. Given the relative ease of making such substitutions with typical word processing software, manually or via a macro, or using or buying an article rewriter, reliance on a single plagiarism detection system may be risky. Our results suggest that using Turnitin together with either Seesources or PlagiarismDetector would help to avoid the weaknesses discussed. However, if such systems used certain pre-processing strategies, likelihood of detection could be improved. For example, an initial test for contiguous strings being within the same Unicode code point ranges could flag insertions, and a simple language model may act as confirmation; systematic thesaurally-driven reduction of both source and targets could deal with thesaural variations. Without such pre-processing, the burden of detection remains with the human reader, who has to become increasingly adept at spotting stylistic variations and any other flags relating to such kinds of trickery as may have been used in order to avoid detection.

While evaluations of plagiarism detection systems are reported [1], the majority have not assessed the specific weaknesses of these systems. Relatively recently, an international workshop hosted the first international competition on plagiarism detection within the 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN-09). The competition involves the detection of two kinds of plagiarism within a reasonably large text corpus, and some 47,000 (constructed) plagiarized articles. At the time of writing, the 2nd competition is underway (and involves a second task on detecting vandalism in Wikipedia). Despite the number of detection systems, there seems to be relatively little apparent participation by vendors who we would expect to benefit from such benchmarking by proving the efficacy of their systems, and using such competitions to address the weaknesses.

5. ACKNOWLEDGEMENTS

We are grateful to Neil Cooke for the pointer to [3], and for reviewing a draft of this paper.

6. REFERENCES

[1] Bull, J., Collins, C., Coughlin, C., Sharp, D., Technical Review of Free Text Plagiarism Detection Software, JISC, (2000).

[2] Culwin, F., The Efficacy of Turnitin and Google, Proc. 9th HEA-ICS Conference, Kent, U.K.(2009)

[3] Dubin, D., The most influential paper Gerard Salton never wrote. *Library Trends,* **52-4,** (2004)

[4] Lancaster, S. and Clarke, R., Automated Essay Spinning, Proc. 9th HEA-ICS Conference, Kent, U.K., (2009)

[5] Wenyin, L., Huang, G., Xiaoyue, L., Min, Z., and Deng, X., Detection of Phishing Webpages Based on Visual Similarity, Proc. World Wide Web Conference, (2005).

[6] WIPO Arbitration and Mediation Center, Reuters Limited v Global Net 2000, Inc., Case No. D2000-0441, WIPO (2000)