ISSN 1559-3096

Plagiary

Lyon, C., Barrett, R., and Malcolm, J. (2006). Plagiarism Is Easy, But Also Easy To Detect. *Plagiary: Cross-Disciplinary Studies in Plagiarism, Fabrication, and Falsification, 1* (5): 1-10 [temporary pagination for advance online copies of articles].

Plagiarism is Easy, but also Easy To Detect

Caroline Lyon, Ruth Barrett and James Malcolm

E-mail: c.m.lyon, r.barrett, j.a.malcolm @herts.ac.uk

Abstract

This paper will first take an overview of plagiarism as a problem, particularly in the field of Higher Education. It will give an outline of pedagogic issues, and approaches to reducing the problem. A significant deterrent is the practice of running students' work through plagiarism detectors, and ensuring that students realise how effectively this can be done. New research indicates that electronic copy detection can also be applied to Chinese text, as is currently done for English and for programming code. We describe one such detector, the Ferret, outlining its application to English text and its potential for use in other domains including Chinese language. We show how the Ferret is based on exploiting underlying characteristics of English word distribution, and that Chinese characters have a similar distribution. The paper concludes by comparing and contrasting man and machine when it comes to identifying copied material, and indicating how their differing memory processes can be harnessed to detect plagiarism.

Introduction

The advent of electronic communication has brought with it many opportunities for plagiarism. The ability to cut and paste presents temptations that did not arise a generation back, when copying meant laboriously typing in someone else's work. This is evident in many fields, not excluding the highest levels of government. See, for example, "Fiasco over the Saddam dossier" (Helm, 2003). However, the problem is particularly acute in Higher Education; studies in USA, Australia and UK suggest that it is reasonable to expect that at least 10% of students' work may be plagiarised (Carroll, 2004).

Plagiarism in Higher Education

In Higher Education cheating with the aid of computers can be broadly divided into two categories. First, students may take material off the Web and use it without proper referencing in essays or reports that are meant to be their own work. This is plagiarism. Secondly, groups of students who have been given the same assignment may work together, sharing disks or other electronic data, when they are meant to be working independently. This is collusion. In both cases students are attempting to pass off someone else's work as their own, and this is the key point in determining whether plagiarism or collusion has taken place.

It is not just the ease of plagiarising that is instrumental in its widespread practice, there are also cultural reasons for its pervasive occurrence. One of these has been an approach in schools to the use of the Internet: pupils are rightly encouraged to make use of this educational resource, and earn credit for taking material off the web, but not enough emphasis has, in the past, been placed on correctly referencing the source. Some students arrive at a Higher Education Institution with the expectation that extensive use of Web sources can result in a good mark. Another underlying cause has been the notion, particularly found among some international students, that it is presumptuous to question authority, and this may misguidedly extend to material found on the Internet. These issues have been well aired recently, for instance at the *Plagiarism: Prevention, Practice and Policy Conference* (Pedden Smith and

Duggan, 2004). The paper by Introna and Hayes (2004) examines how some Chinese students have been taught to memorise precisely, "Capturing the exact expression – through meticulous memorisation – is capturing the reality as such". This gives such students a different perspective on copying, which has to be explicitly addressed. At the University of Hertfordshire, in common with other institutions, we take a number of steps to educate students on the problem of plagiarism, starting as soon as they arrive. New students have classes with informal discussions on what constitutes plagiarism; they are taught that it can be sensible to quote an authority, but this must be properly referenced. In programming they are taught the advantages of coding in a modular structure so that sections of code can be re-used, but they learn that it is cheating to pretend that code written by someone else (or automatically generated) is their own work. We also explore the problem of collusion, and explain where to draw the line between students helping each other in acceptable ways and unacceptable collusion (Barrett and Cox, 2005). One of the most important practices in instilling proper standards is to show students how easy it is to detect plagiarism and collusion, which acts as a deterrent in the first place.

Copy Detection as a Deterrent

A well known commercial plagiarism detection service is Turnitin, a browser-based tool that compares a submitted file against material published on the web and against a database of previously submitted student work. Its web site claims that the database now has more than 4.5 billion pages (Turnitin, 2005). From 2003 to 2005 UK Universities and other institutions have subscribed to the commercial plagiarism detector Turnitin for free, paid for by JISC (Joint Information Systems Committee) a government agency, but from the end of 2005 each institution has to pay for the service.

For each file submitted Turnitin returns the results of the searches, displayed in a "similarity report" for each file, which shows links to web sources and other files in the database. The student's file may be an individual piece of work like a project report, or there may be coursework on the same topic done by a large class of students, in which case the detection of collusion is as important as the detection of copying off the Web. When large cohorts of students are given the same coursework to be done individually, particular problems can arise. A number of members of staff may mark the work, and there is no prospect of detecting copying without an automated system unless the pair of documents concerned happen to be seen by the same marker.

An alternative, primarily used for detecting collusion in students' coursework assignments, is the Ferret copy detector which is a standalone system that can be installed on a standard PC, and is free. This has been developed at the University of Hertfordshire where it is in use to detect plagiarism and collusion in students' essays and reports and also in programming code. To run the Ferret the user collects the files to be processed in a folder on the computer, browses to them, selects them, and then runs the plagiarism detector. We describe below the Ferret copy detector, the algorithm on which it is based, and explain why this can be applied to Chinese, where plagiarism detection is at an early stage of development. The algorithm underlying Turnitin is not made public, so we cannot say whether it resembles the Ferret on a technical basis.

Copy Detection with the Ferret

Theoretical basis

The occurrence of words in English and other European languages follows a Zipfian distribution. This means that some words occur frequently (function words like "the", "to", "of" and so on) but most words are relatively scarce. Quite ordinary words actually have a low frequency. This has been noted for many decades,

initially by Zipf (1949) then by Shannon (1951), and frequently quantified more recently (Manning and Schutze, 1999). As an example, in the Brown corpus of 1 million words, taken from representative samples of everyday texts, 40% of the word forms occur only once (Kupiec, 1992). Now, this distinctive distribution will be more pronounced for word bigrams (two consecutive words), and even more marked for trigrams (three consecutive words).

To illustrate how a text can be converted into a set of trigrams consider the following example. A sentence like *Plagiarism is easy, but also easy to detect* becomes the set

plagiarism is easy	*is easy but*	*easy but also*
but also easy	*also easy to*	*easy to detect*

The Zipfian distribution of English language words is illustrated by statistics cited by Gibbon, Moore and Winski, (1997) on frequency of trigrams in the Wall Street Journal (WSJ) corpus, shown in Table 1. Note that these figures relate to text in the limited domain of financial reporting, where the same topic may be frequently revisited. We see that in about 38 million words 77% of trigrams are singletons, occurring only once. Any article will on average have 77% of its trigrams unique. Even in a very large corpus in a limited domain most trigrams in independently written articles do not match those in other articles.

Illustration from Google

Consider the title of this paper, which at the time of writing does not appear in the Google index of roughly 9 billion documents. Taking each of the seven words alone we find that the least frequent "plagiarism" occurs in about 10 million documents, the most frequent "to" in almost all 9 billion. However, when the words are combined the frequency of bigram and trigram occurrence rapidly falls: the frequency for the trigram "plagiarism is easy" is down to 281.
It is our contention (backed up by the success in practice of our software) that in practice once we combine a number of triples we can "fingerprint" a piece of text just as effectively as

Table 1. Data from Three Sources to Illustrate the High Proportion of Trigrams Unique in Independently Written Text

Source	Number of Words In Corpus	Distinct Trigrams	Unique Trigrams (occur only once)	% of Trigrams That Are Unique
TV News corpus	985,316	718,953	614,172	85%
Federalist Papers (part)	183,372	135,830	118,842	87%
Wall Street Journal	972,868	648,482	556,185	86%
[Gibbon, 1997, p258]	4,513,716	2,420,168	1,990,507	82%
	38,532,517	14,096,109	10,907,373	77%

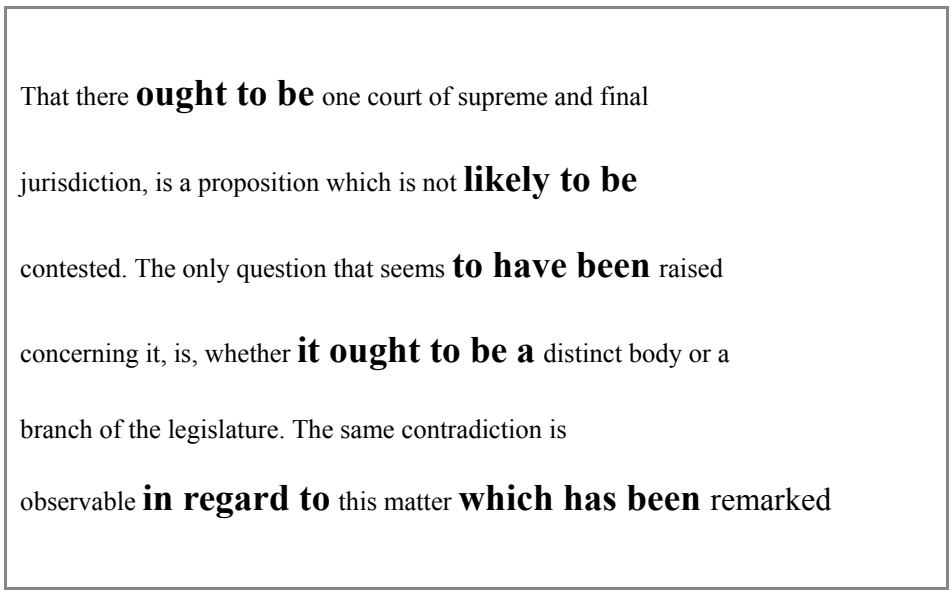

Figure 1. Example from "The Federalist Papers" comparing two independently written texts.
When two papers on the judiciary are compared the following is typical of results: a sprinkling of matching trigrams that are highlighted.

longer string matching, and this method is less susceptible to false negatives when small changes are made to a document in order to hide similarity.

Our initial research in this field was carried out on The Federalist Papers, a collection of essays on which the American constitution was based, written by three authors: Hamilton, Madison and Jay (1787). These constitute a very well known collection, which have been analysed minutely by many researchers. The authorship of some of the papers has been disputed, and we thought that there would be more trigram similarity between papers written by the same author than between those written by different authors. However, this hypothesis turned out to be unfounded: the proportion of matching trigrams was no higher in pairs of papers written by the same author than in those written by different ones. Independently written texts have a low level of matching trigrams, even when they are written by the same person on related subjects at different times (Lyon, Malcolm and Dickerson, 2001). An example of the level of matching is shown in Figure 1.

Practical implementation

Based on these empirical findings we can take a large collection of documents and convert each one to a set of trigrams. Then we can compare each set with every other to see whether there are any suspicious levels of matching. In fact, we use a novel algorithm (developed by one of the authors) so that only one pass has to be made through the collection, and the time taken for processing hundreds of documents is measured in seconds. The Ferret then displays suspiciously similar documents side by side with matching text highlighted, so there is an immediate visual impression when copying has occurred. Compare Figure 1 with Figure 2.

An advantage of basing the detection on short lexical strings is that similarity is not necessarily undermined by deletions, insertions, or substitu-

In particular, I could not understand how FileMaker Pro could provide the server-side processing capabilities of CGI. Consequently, **I used the Internet to find out more about FileMaker Pro. I** read quite a few book reviews **and visited sites with related information.**

Figure 2. Example from students' work, where collusion has taken place. When two pieces of coursework are compared the following is typical of results, with solid blocks of matching trigrams that are highlighted. Note that there can be insertions, deletions and substitutions without obscuring the underlying similarity. Compare with Figure 1.

tions, as shown in Figure 2. With a basis of longer word strings a match is lost if one word in the string is changed. Early experimental work showed that converting documents to a list of single words, or to word pairs, did not have enough discriminating power. Overlapping trigrams were the shortest lexical strings that could effectively detect copying. Very brief experiments on the doomed European Constitution suggested that tetragrams might be necessary for this type of legal document, but are not normally needed, and not included in the current version of Ferret.

Detecting Plagiarism and Collusion in Chinese

Automated language processing in Chinese is particularly difficult because the written language is represented by characters, rather than an alphabetic system, and there is a very large number of characters. Words can be composed of one, two or three characters, but there is no explicit word boundary and finding these boundaries by machine is a hard problem.

However, this problem can be circumvented by using a version of the Ferret, modified to process Chinese characters. The Ferret is a processor of discrete sequential data, where, applied to English, the data items are words. In Chinese we would also process discrete sequential data, but in this case the data items would be characters. We can convert each document in a collection to a set of character trigrams, as has been done with English words, and compare these sets for similarity. The classic problem of finding word boundaries in Chinese is then irrelevant. A set of character sequences would not be a set of meaningful linguistic elements for the most part. If two documents were copies, or partial copies, then we expect that the two sets of character trigrams would have more matches than independently written texts would have. Note that in these initial experiments the choice of documents is arbitrary.

In order for this approach to be effective, the distribution of Chinese characters must have the same Zipfian characteristics as English. Prelimi-

Table 2. Data To Compare With Table 1 to Illustrate That Chinese Characters Have a Similar Distribution To English Words

The statistics for tetragrams (sequences of 4) are also shown. The Chinese corpus is composed of three concatenated documents. They are: a CCTV (China Central Television) Survey, a famous Chinese Martial Art Novel and the Romeo and Juliet Drama (Chinese Version).

Source	Number of Characters In Corpus	Distinct Trigrams (3 consecutive characters)	Unique Trigrams	% of Trigrams That Are Unique
Chinese corpus	89575	61330	54052	88%
		Distinct Tetragrams (4 consecutive characters)	Unique Tetragrams	% of Tetragrams That Are Unique
		72111	68568	95%

nary experiments have been carried out, and this seems to hold. Some results are summarised in Table 2, which should be compared to Table 1. We note that in this comparatively small corpus of 89,575 characters 88% of the trigrams are unique. The statistics for tetragrams (sequences of 4) shows that 95% are unique. The Zipfian distribution is pronounced, as it is for English words, and warrants further investigations of this approach.

There are several different ways in which Chinese characters are represented in digital form. The method used here is GB2312-80, which is the official character set of the People's Republic of China. This is a national standard that defines about 6763 Chinese characters and also symbols such as punctuation marks and numerals. There is also an extended form, GBK, which includes more traditional characters. Both representations use 2 bytes per character. Other representations are standard forms used in Taiwan, Malaysia, Singapore and elsewhere.

We are collecting data from universities in China to carry this work on plagiarism detection further. Work done in the field already includes that of Bao (2003, 2004a, 2004b) which he has initially applied to English, but is extending to Chinese (personal communications, September 2005). Work done so far is based on semantic feature extraction and analysis, but he will be investigating other approaches, including the Ferret.

Detecting Copying in Programming Code

The detection of similar sections of code is another critical area of work, not only in Higher Education but also in industrial situations. For example, in very large software developments program modules may be re-used as clones of the original. If a section of code has to be modified it is necessary to locate and correct all the clones, which may be similar but not identical (Carter, Frank and Tansley, 1993).

The detection of copying in code can be addressed at several levels. The most crude copies may be almost identical, easy to find with many

approaches. However, it is often found that students rewrite textual comments when they copy code, and may also change user defined names, such as variables and classes. The underlying similarity is that of syntax and structure, which is harder to detect. The JPlag system (Malpohl, 2005) is an effective approach to detecting plagiarism in code, whether simplistic or more sophisticated. However, for our first and second year undergraduates we can use the Ferret to detect similar sections of code, since these students' attempts to plagiarise are not sophisticated. This approach is quick and convenient for classes of up to 200 students, where several members of staff will share the job of marking a coursework assignment. We ignore comments and take code as a sequential string of symbols, which are treated as words are in text. Thus for Java, for instance, we adapt the Ferret to take any of the symbols of the language, such as "=", "!=" or "==" as "words". Usually some pre-processing is needed: it is common to start teaching Java in the BlueJ environment (BlueJ, 2006), so the students' programs must be extracted from the environmental code. Also, they may all be given a ready written section of code to incorporate into their program, so this must be extracted too. In developing this system we have analysed 180 second year programming assignments that were manually marked. In these assignments one or two pairs of programs that showed collusion had been found. When we ran the programs through the Ferret these pairs were identified, but so were several others that had slipped through the net as they had been marked by different markers. There were no false positives. This year we will be using the detector as a standard procedure in the School of Computer Science.

At the start of a programming course it is not usually possible to pinpoint copying, since the tasks are simple and do not admit much variety. Similar solutions will be submitted without any collusion. However, after a few months, coursework assignments can be approached in various ways, and copying can be picked up.

Comparison of Plagiarism Detection By Man and Machine

An examination of the use of human language shows that it is based not only on single words, but more often on groups of words. It can be shown that spoken and written English is easier to comprehend if it is divided up into the right sort of "chunks" (Lyon, Dickerson and Nehaniv, 2003b). Thus, common formulaic expressions are often the building blocks of sentences and other types of speech and text (Wray, 2005). This becomes more obvious when we consider that many of the most frequently used words in English and other languages are homophones: words such as *<their, there> <I, eye> <one, won>*, which sound the same but have different meanings. We have no problem disambiguating these because we take them in context, and process short sequential fragments. As human language has evolved we exploit the advantages of processing short sequences of words.

This is consistent with results from recent work using fMRI (functional Magnetic Resonance Imaging) which shows which part of the brain is active when language is processed. A critical component of many human functions, both motor and cognitive, is the primitive sequencing processor that may have originated when our earliest hominid ancestors began to walk. Lieberman (2002) says: "advances in brain imaging and behavioural studies of human subjects support [the] hypothesis that the basal ganglia perform cognitive sequencing functions" and "deficits in sequencing manual motor movements and linguistic sequencing …. were correlated". In human speech and language, sequential processing seems to be necessary at many levels – phonetic, lexical, and syntactic.

Taking text as short lexical sequences as the basis for copy detection has been adopted by automated processors, including the Ferret. This has been found more effective than basing copy detection on single words or alternatively on

long strings of words or characters. Experiments with the Ferret have shown that the Ferret can identify similar paragraphs within 300 texts of 10,000 words each. Then results are presented as an ordered list of file-pairs, ranked according to their similarity. Inspection of the two files will show the highlighted similar paragraphs side by side. Detailed descriptions of actual analyses carried out can be found in Lyon, Barrett and Malcolm (2003).

However, though the basis of short lexical sequences underlies this machine approach to copy detection there is a significant difference from human language processing. For humans, the semantics underlying the group of words is of critical importance: we remember the *meaning* of phrases, not just word strings (Wanner, 1974; Russell and Norvig, 2003, p. 243). In contrast, the machine stores exact word strings, most of which are meaningless. In Chinese this is even more pronounced: splitting up words and taking sequences of characters across word boundaries produces elements that are devoid of meaning. However, this is irrelevant; the system does not require any semantic analysis, in contrast to human language processing. Natural analogues inspire computing processes, but should always be open to scrutiny. Machines have their own very different capabilities that we can exploit.

Summary

It is now very easy to plagiarise the work of others using electronic means, but it is also easy to use these same electronic means to detect plagiarism. We have shown that original work, even within texts from the same author, has an unique distribution of trigrams (three consecutive words) and this can be the theoretical basis of an electronic detection tool. The Ferret detection tool can be used to detect similar passages in English text, and preliminary investigations have indicated that it may also be used with Chinese text using three consecutive characters as the basis for the algorithm. The Ferret is also used in introductory Java programming classes to detect collusion between students. Machine processing of the short lexical or symbol sequences can be very quick and comprehensive, whereas human processing of potentially plagiarised texts requires a semantic component, the recognition of meaning.

REFERENCES

Barrett, R., & Cox, A.L. (2005). 'At least they're learning something': the hazy line between collusion and collaboration. *Journal of Assessment and Evaluation in Higher Education*, *30* (2), 0260-2938.

Bao, J-P., Shen, J-Y., Liu, X-D., Liu, H-Y. & Zhang, X-D. (2004a). Semantic sequence kin: A method of document copy detection. In *Proceedings of Advances In Knowledge Discovery and Data Mining*. *Lecture Notes in Artificial Intelligence (LNAI)*. 3056: 529-538.

Bao, J-P., Shen, J-Y., Liu, X-D., Liu, H-Y. & Zhang, X-D. (2004b). Finding plagiarism based on common semantic sequence model. In Q. Li, G. Wang and L. Feng (Eds.), *Proceedings of the 5th International Conference on Advances in Web-Age Information Management (WAIM)*. *Lecture Notes in Computer Science (LNCS)*. 3129: 640-645.

Bao, J-P., Shen, J-Y., Liu, X-D., Liu, H-Y., & Zhang, X-D., (2003). Document copy detection based on kernel method. In *Proceedings of Natural Language Processing and Knowledge Engineering Conference (IEEE)*, pp. 250-255.

BlueJ Retrieved January 3, 2006, from http://www.bluej.org/index.html

Carroll J. (2004). Institutional issues in deterring, detecting and dealing with plagiarism. Retrieved March 7, 2005, from http://www.jisc.ac.uk/uploaded_documents/plagFinal.doc

Carter, S., Frank, R. & Tansley, D. (1993). Clone Detection in telecommunication software: a neural net approach. In J. Alspector, R.Goodman & T.Brown (Eds.), *International Workshop on Applications of Neural Networks in Telecommunications*.

Gibbon, D., Moore, R., & Winski, R. (1997). *Handbook of standards and resources for spoken language systems*. Mouton de Gruyter.

Hamilton, A., Madison, J., & Jay, J. (1787). *The federalist papers*. Retrieved January 2, 2006, from http://www.foundingfathers.info/federalistpapers/

Helm, T. (2003, February 8) Fiasco over the Saddam dossier. Retrieved December 18, 2005, from http://www.telegraph.co.uk/news/main.jhtml?xml=/news/2003/02/08/ndoss08.xml

Introna, L., & Hayes, N. (2004). Plagiarism, detection and intentionality: on the construction of plagiarists. In A. Pedden Smith & F. Duggan (Eds.), *Plagiarism: Prevention, Practice & Policy Conference*. Northumbria University Press.

Kupiec, J. (1992). *Robust part-of-speech tagging using a hidden Markov model. Computer Speech and Language, 6*(3), 225-242.

Lieberman P., (2002). On the nature and evolution of the neural bases of human language. In the *Yearbook of Physical Anthropology*.

Lyon, C., Malcolm, J. & Dickerson, B. (2001). Detecting short passages of similar text in large document collections. In *Proceedings of Empirical Methods in Natural Language Processing Conference*, pp. 118-125.

Lyon, C., Barrett, R. & Malcolm, J. (2003). Experiments in electronic plagiarism detection. TR 388, Computer Science Department, University of Hertfordshire.

Lyon, C., Dickerson, R. & Nehaniv, C. L. (2003). The segmentation of speech and its implications for the emergence of language structure. In *Evolution of Communication*, *4*(2), 161-182.

Malpohl, G. (2005). *JPlag.– detecting software plagiarism*. Retrieved December 18, 2005 from http://wwwipd.ira.uka.de:2222/

Manning, C., & Schutze, H. (1999). *Foundations of Statistical Language Processing*. MIT Press.

Pedden Smith, A., & Duggan, F. (Eds.). (2004). *Plagiarism: Prevention, Practice & Policy Conference*. Northumbria University Press.

Russell, S. & Norvig, P. (2003). *Artificial intelligence: A modern approach*. Prentice Hall.

Shannon, C. E., (1951). Prediction and entropy of printed English. In N.J.A. Sloane and A.D. Wyner (Eds.), (1993). *Shannon: Collected Papers* IEEE Press.

Turnitin. (2005). Retrieved on December 28, 2005, from http://www.turnitin.com/static/home.html

Wanner, E., (1974). *On remembering, forgetting and understanding sentences.* Mouton, the Hague.

Wray, A. (2005). *'Needs only analysis' in linguistic ontogeny and phylogeny.* In *Proceedings of the 2nd International Symposium on the Emergence and Evolution of Linguistic Communication*.

Zipf, G. K., (1949). *Human Behaviour and the Principle of Least Effort.* Addison Wesley, Cambridge.

Caroline Lyon, Ruth Barrett, and James Malcolm teach and conduct research in the School of Computer Science at the University of Hertfordshire, United Kingdom. As well as plagiarism deterrence, detection and prevention, their work includes areas such as systems development, network security, speech and language processing, and the evolution of language. Visit http://homepages.feis.herts.ac.uk/~comrcml/ for links to software and other papers which report on their ongoing research.