Efficient Search for Plagiarism on the Web

James A. Malcolm and Peter C. R. Lane
School of Computer Science, University of Hertfordshire
Hatfield, Herts, UK
EMail: j.a.malcolm *or* p.c.lane @herts.ac.uk

***Abstract*— Understanding the characteristics of written English allows Internet search for the source of a document to be carried out efficiently. There is a Zipfian distribution of word frequencies in natural language, with some words common and many words rare. If we take a group of three words, the rarity of most of these triples is extreme. This can be exploited to detect web pages similar to a given target document: while a Google search for some triples from the target may return many hits, other triples will only be found in a few documents on the Internet. These documents may well be similar to the target, and are certainly worth examining more closely. Initial experiments show that this approach is very promising, and it is being implemented in a software tool called WebFerret.**

***Index Terms*— Plagiarism, Search Engines, Ferret.**

Overview

In this paper we review the problem of plagiarism, explain how the Ferret software tool detects similar pairs of documents in a collection, and show how Ferret can be extended to efficiently detect sources of Internet plagiarism.

The Problem of Plagiarism

The problem of plagiarism, whether real or perceived, is an important and emotive issue for both students and staff in higher education; both are concerned to maintain the quality of degrees. Honest students can feel aggrieved that while they are working hard to earn their degrees, others may be gaining their qualifications by cheating. Someone who hands in plagiarised work that is not discovered may even gain a better award than an honest student!

Staff do not want to waste their time marking and giving feedback on work that was not done by the student who submitted it. It is very time consuming to search for the source of plagiarism if it is suspected, and even more time consuming to document the evidence if it is confirmed [1]. It is also very annoying to invest that effort, but then to fail at the investigation stage. Even among staff who have not recognised plagiarism, it is a problem; it raises our expectations of what an average student might reasonably produce. One effect of this is that degrees are devalued over time. The detection and prevention of plagiarism is therefore an important topic.

But according to THES (June 23, 2006, p.4), “53% of students said that they did not believe their tutors would spot cheating” [2]. The same survey of 3200 students (commissioned by JISC PAS) found that 87% supported the use of electronic detection tools.

University procedures when plagiarism is suspected frequently require staff to fully document relevant passages, both in the student’s work, and in the original sources. This is time-consuming when carried out manually, even with the help of Google searches. Not only must the correct source documents be found, but the offending passages must be identified. Automated tools, such as Ferret [3] can help to locate and document copying with the minimum of invested time.

We have identified 4 catagories of plagiarism, based on the source of the copied work [4].

1) If the source is a fellow student, then the offence is generally referred to as collusion. The significant point here is that we have got the source. Somewhere in the pile of work to be marked is the document that was copied. Existing tools such as Ferret rapidly find such copies and are effective also at documenting collusion.
2) If the source is the Internet, then it is likely that the culprit may have used material from more than one source, and may have made changes. But we can find the source. The student presumably typed some fairly obvious keywords into a search engine, which then gave him the raw material for his essay. But if the dish is under-cooked, he has committed plagiarism and we can find his sources in exactly the same way that he did. In fact we shall see later in this paper that we can often identify directly which source he used, with very little effort.
3) The third possible source is an essay bank. Here we can’t easily find the source. Some commercial plagiarism detection services claim to have obtained essays from essay banks, though there is a more honourable and economical way to do this than paying money: simply wait for another student to pay for and submit an essay from the same source.
4) In the final case, where the work was written to order, then it is completely impossible to find the source. Unless the bespoke author used Internet sources to construct the essay, the only avenue open is to find indications that the student who submitted the work is in fact unfamiliar with its content (see e.g. [5]). This is the problem more recently identified as “contract cheating” which has been extensively investigated by Lancaster [6] and others [7].

These four categories of plagiarism are useful in considering approaches to dealing with the problem.

The focus of this paper is on detecting plagiarism from the Internet, and we analyse an extension to our existing plagiarism detection tool Ferret to address this requirement.

WHAT IS FERRET?

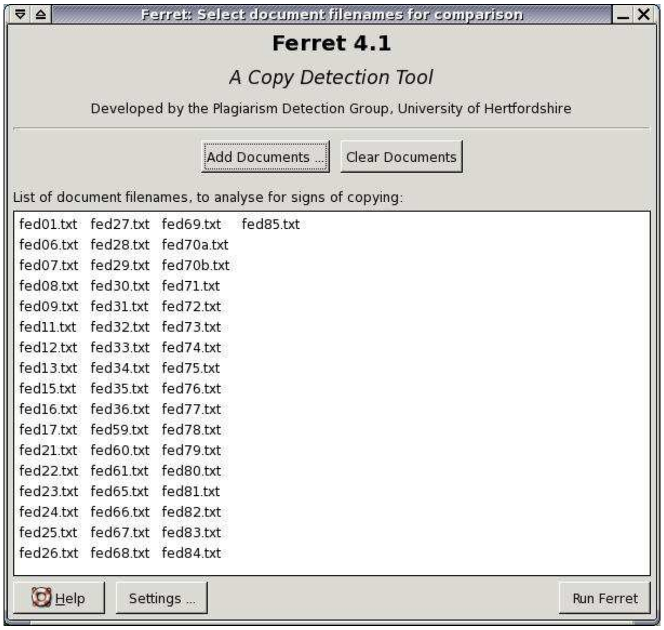
Ferret is a copy-detection tool, which has been produced at the University of Hertfordshire [8]. It has been developed over more than 6 years, is *freely* available on the Internet [9], and has been used by HE institutions around the world. It analyses documents on the user's own computer, extracting text from pdf, Word or RTF formats. It has a fast and intuitive interface and produces reports highlighting any evidence of copying, ranking all pairs of documents based on similarity. It can detect copying in multiple languages (both natural and computer) simply by changing the definition of a "word" within the program code. This can vary: for Chinese we would use a single character in place of a word for western languages [10]. For computer languages it may be helpful to add additional tokens alongside normal words.

Ferret makes it easy to compare large collections of documents for signs of copying. It is also very fast: the algorithm it uses is linear, both in space required and in time taken, as the total number of words in the input documents grows. Comparisons of the Ferret algorithm with other approaches [11], [12] show that its performance is excellent.

However, Ferret has one limitation, which is that it only works on collections of documents provided by the user. A typical question raised by potential users is whether it can search the Internet for related documents. We are developing such a version of Ferret, and this paper describes the main technical issues which we have explored in the process of so doing.

Running the current Ferret has three stages:

1) Select documents to compare, by identifying them using a file selector as shown in figure 1.



Ferret: Select document filenames for comparison

Ferret 4.1

A Copy Detection Tool

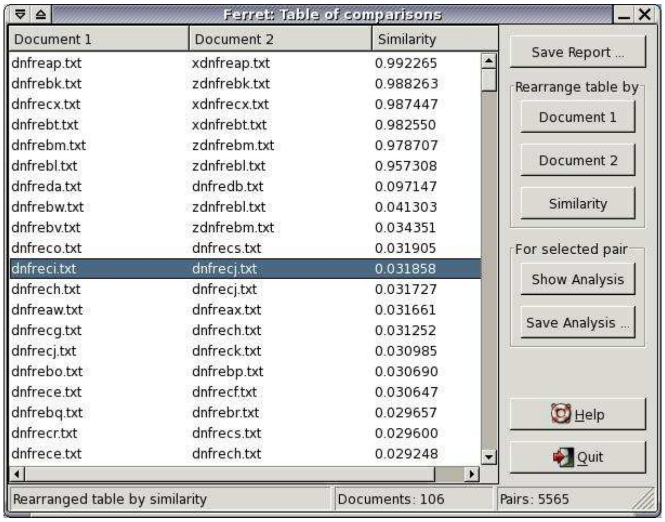
Developed by the Plagiarism Detection Group, University of Hertfordshire

List of document filenames, to analyse for signs of copying:

fed01.txt
fed06.txt
fed07.txt
fed08.txt
fed09.txt
fed11.txt
fed12.txt
fed13.txt
fed15.txt
fed16.txt
fed17.txt
fed21.txt
fed22.txt
fed23.txt
fed24.txt
fed25.txt
fed26.txt
fed27.txt
fed28.txt
fed29.txt
fed30.txt
fed31.txt
fed32.txt
fed33.txt
fed34.txt
fed35.txt
fed36.txt
fed59.txt
fed60.txt
fed61.txt
fed65.txt
fed66.txt
fed67.txt
fed68.txt
fed69.txt
fed70a.txt
fed70b.txt
fed71.txt
fed72.txt
fed73.txt
fed74.txt
fed75.txt
fed76.txt
fed77.txt
fed78.txt
fed79.txt
fed80.txt
fed81.txt
fed82.txt
fed83.txt
fed84.txt
fed85.txt

Fig. 1. Selecting files to check

2) Analyse documents, producing a ranked list of document pairs (as shown in figure 2) and a measure of *similarity* (as explained later).



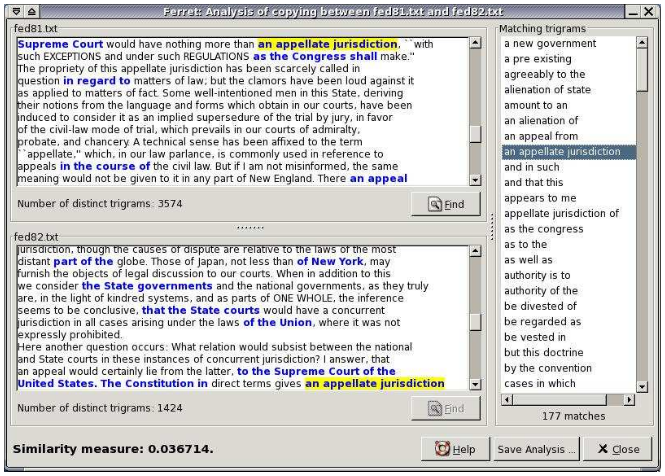
Ferret: Table of comparisons

Document 1	Document 2	Similarity
dnfreap.txt	xdnfreap.txt	0.992265
dnfrebk.txt	zdnfrebk.txt	0.988263
dnfrecx.txt	xdnfrecx.txt	0.987447
dnfrebt.txt	xdnfrebt.txt	0.982550
dnfrebm.txt	zdnfrebm.txt	0.978707
dnfrebl.txt	zdnfrebl.txt	0.957308
dnfreda.txt	dnfredb.txt	0.097147
dnfrebw.txt	zdnfrebl.txt	0.041303
dnfrebv.txt	zdnfrebm.txt	0.034351
dnfreco.txt	dnfrecs.txt	0.031905
dnfreci.txt	dnfrecj.txt	0.031858
dnfrech.txt	dnfrecj.txt	0.031727
dnfreaw.txt	dnfreax.txt	0.031661
dnfrecg.txt	dnfrech.txt	0.031252
dnfrecj.txt	dnfreck.txt	0.030985
dnfrebo.txt	dnfrebp.txt	0.030690
dnfrece.txt	dnfrecf.txt	0.030647
dnfrebq.txt	dnfrebr.txt	0.029657
dnfrecr.txt	dnfrecs.txt	0.029600
dnfrece.txt	dnfrech.txt	0.029248

Rearranged table by similarity | Documents: 106 | Pairs: 5565

Fig. 2. Pairs in order of similarity

3) Compare pairs of documents, as shown in figure 3, to see which parts may have been copied.



Ferret: Analysis of copying between fed81.txt and fed82.txt

fed81.txt

Supreme Court would have nothing more than an appellate jurisdiction, ``with such EXCEPTIONS and under such REGULATIONS as the Congress shall make.'' The propriety of this appellate jurisdiction has been scarcely called in question in regard to matters of law; but the clamors have been loud against it as applied to matters of fact. Some well-intentioned men in this State, deriving their notions from the language and forms which obtain in our courts, have been induced to consider it as an implied supersedure of the trial by jury, in favor of the civil-law mode of trial, which prevails in our courts of admiralty, probate, and chancery. A technical sense has been affixed to the term ``appellate,'' which, in our law parlance, is commonly used in reference to appeals in the course of the civil law. But if I am not misinformed, the same meaning would not be given to it in any part of New England. There an appeal

Number of distinct trigrams: 3574

fed82.txt

jurisdiction, though the causes of dispute are relative to the laws of the most distant part of the globe. Those of Japan, not less than of New York, may furnish the objects of legal discussion to our courts. When in addition to this we consider the State governments and the national governments, as they truly are, in the light of kindred systems, and as parts of ONE WHOLE, the inference seems to be conclusive, that the State courts would have a concurrent jurisdiction in all cases arising under the laws of the Union, where it was not expressly prohibited.
Here another question occurs: What relation would subsist between the national and State courts in these instances of concurrent jurisdiction? I answer, that an appeal would certainly lie from the latter, to the Supreme Court of the United States. The Constitution in direct terms gives an appellate jurisdiction

Number of distinct trigrams: 1424

Matching trigrams

a new government
a pre existing
agreeably to the
alienation of state
amount to an
an alienation of
an appeal from
an appellate jurisdiction
and in such
and that this
appears to me
appellate jurisdiction of
as the congress
as to the
as well as
authority is to
authority of the
be divested of
be regarded as
be vested in
but this doctrine
by the convention
cases in which

177 matches

Similarity measure: 0.036714.

Fig. 3. One similar pair examined

In addition to the displays of the working program, Ferret allows the user to save copies of the analysis and detailed comparisons into pdf reports, for printing or later use, possibly as evidence.

HOW FERRET WORKS

Ferret works by extracting *trigrams* (sequences of three words). If we take as an example the phrase "multicasting is a standard feature in Internet . . . ", then the trigrams in this phrase are: "multicasting is a", "is a standard", "a standard feature", "standard feature in", "feature in Internet" . . .

Note that the number of trigrams is two less than the document length in words. Some trigrams, e.g. "a standard feature", are fairly generic, others, e.g. "multicasting is a", are topic specific.

The reason that Ferret is so fast is that we build an index of trigrams as the documents are read, so if there are n input files, checking all $\frac{n.n-1}{2}$ pairs of documents is done in more or less linear time.

In order to *rank* documents by similarity, we need some metric. If A is the set of trigrams from document 1, and B is the set of trigrams from document 2, then

$$\text{Similarity} = \frac{\text{Number of common trigrams}}{\text{Total number of trigrams}} = \frac{|A \cap B|}{|A \cup B|}$$

For example, suppose document 1 is “multicasting is a standard feature in Internet”. If document 2 is “multicasting is a feature of the Internet” then there is just one common trigram: “multicasting is a”.

The total number of trigrams is 9, as there are 5 in each document but 1 trigram is common. They are “multicasting is a”, “is a standard”, “a standard feature”, “standard feature in”, “feature in Internet”, “feature in Internet”, “is a feature”, “a feature of”, “feature of the”, “of the Internet”. This means that (for this tiny example) our similarity metric would be $1/9$ (which is 0.11 or 11%).

In practice a lecturer looks not for a particular value of similarity, but rather looks at the most similar pair of documents first, then the next most similar, and so on .... The point at which she would stop is when her academic judgement says that the pairs she is examining no longer show any signs of plagiarism. If matching trigrams are scattered over the whole document, copying is not indicated, but if they are in closely packed blocks it is likely that both files in the pair share some common source. In fact we plan to implement a metric that automates that aspect of the lecturer’s judgement.

Why not use Turnitin?

Our aim is to *search* automatically for potential sources of plagiarism, but automated tools for plagiarism detection have existed for many years. The commercial, US-based company Turnitin is perhaps the best known. It offers plagiarism detection against Internet available sources, so it would seem silly for us to compete with an established service.

However documents must be *given* to Turnitin, leading to a transfer of intellectual property; Turnitin *charges* for its use; and Turnitin cannot be *customised*, as it is a closed, commercial system.

The new version of Ferret, WebFerret, will avoid all three of Turnitin’s problems. First, staff will retain ownership of their documents: WebFerret may be used on your own computer, and the web application will not retain copies of documents. Second, WebFerret will be free for staff to use. Third, Ferret has been designed to be extended. Currently, Ferret works on English, other European languages, Chinese and (certain) computer programming languages. As a product of the University, Ferret can be tailored to suit the needs of staff within the University needing to detect plagiarism in different kinds of documents.

Our Approach

How should we select Internet sources to go in our document collection? One approach would be to search on the same keywords that the students are likely to have used. But if you have a document that you suspect may be plagiarised, you can use the same technique that our plagiarism detector uses. Do a Web search of some unusual phrases in the text; do not search for phrases that would indicate the subject matter of the document as that makes it harder to spot whether you have hit the right document.

For example, consider the following piece of text:

> “It is at best a temporary utility that will eventually become obsolete when multicasting is a standard feature in Internet routers. By then there will be an established base of MBone users (which should make the router manufacturers happy).”

The triples in the first sentence of this sample piece of text are shown in table I together with the number of hits produced by a Google search on that exact string.

Triple	**Frequency**	**Common?**
it is at	about 1,770,000	* * *
is a standard	about 1,750,000	* *
is at best	about 1,470,000	* *
at best a	about 1,360,000	*
that will eventually	about 1,290,000	* *
a standard feature	about 743,000	*
will eventually become	about 699,000	*
utility that will	about 461,000	* *
standard feature in	about 116,000	*
eventually become obsolete	about 32,600	
become obsolete when	about 22,400	*
feature in Internet	about 18,100	*
best a temporary	about 16,800	
multicasting is a	about 12,900	* *
in Internet routers	about 10,400	*
a temporary utility	about 2,070	
when multicasting is	about 787	* *
temporary utility that	3	*
obsolete when multicasting	2	*

TABLE I

Google document frequencies for trigrams of sample text

Although the most startling observation is that a few triples only appear 2 or 3 times, it is important to note the frequencies of even the relatively common strings: although some triples are fairly frequent, most are much less so, as the following graph (figure 4) shows. The common words (listed later, and marked with an asterisk here) tend to lead to more common triples (though not entirely so).

Contrast the frequencies of the individual words shown in table II, which have the same rapid fall-off in frequency, but for a starting point more than three orders of magnitude higher. Again the graph in figure 5 makes this plain.

The list of 116 common words (and artefacts of the Internet) is: the, be, to, of, and, a, in, that, have, i, it, for, not, on, with, he, as, you, do, at, this, but, his, by, from, they, we, say, her, she, or, an, will, my, one, all, would, there, their, what, so, up, out, if, about, who, get, which, go, me, when, make, can, like, time, no, just, him, know, take, person, into, year, your, good, some, could, them, see, other, than, then, now, look, only, come, its, over, think, also, back, after, use, two, how, our, work, first, well, way, even, new, want, because, any, these, give, day, most, us, b, c, p, html, s, t, e, br, www, http, h, is,

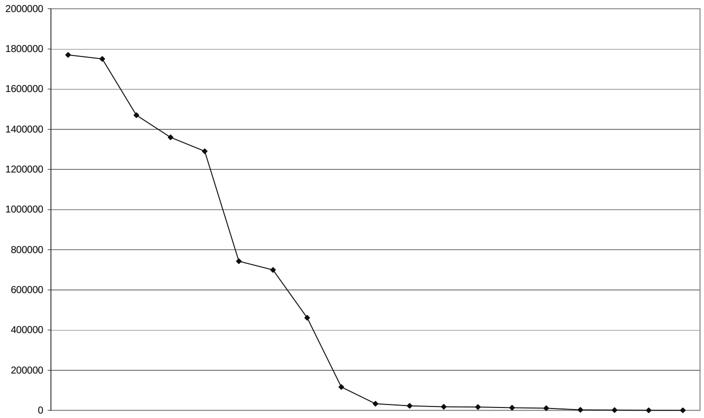


Fig. 4. Declining frequencies of word triples

Word	Frequency	Common?
a	6,800,000,000	*
in	5,540,000,000	*
is	4,000,000,000	*
it	3,050,000,000	*
that	2,890,000,000	*
at	2,800,000,000	*
will	2,620,000,000	*
when	1,920,000,000	*
internet	1,560,000,000	
best	1,510,000,000	
become	705,000,000	
standard	605,000,000	
feature	395,000,000	
eventually	151,000,000	
utility	140,000,000	
temporary	114,000,000	
routers	40,400,000	
obsolete	23,500,000	
multicasting	1,470,000	

TABLE II

GOOGLE DOCUMENT FREQUENCIES FOR WORDS FROM THE SAMPLE TEXT

was, went, were, are. The commonest 8 words in the sample sentence are on this list.

Further evidence of the rarity of certain phrases (even in a database as large as the Internet) can be found by a search for “temporary utility” and “manufacturers happy” – an Alta-Vista search gave three hits on the query: +“temporary utility” +“manufacturers happy”. One is a book on the subject, the second is an acknowledged quote from that book, and the third is a plagiarised student report (accessible via the instructor’s web-site).

We repeated this experiment more recently using Google. As can be seen in table III, the number of copies of that paragraph had increased from three to six: the original, two copies with citation, and three without. We also found further evidence of the rarity of specific word triples.

The addition of a perfectly common word like “that” to the end of a search string reduced the number of hits from around 21,000 to just the six matching documents: more evidence for the effectiveness of trigrams as a basis for automated Internet search.

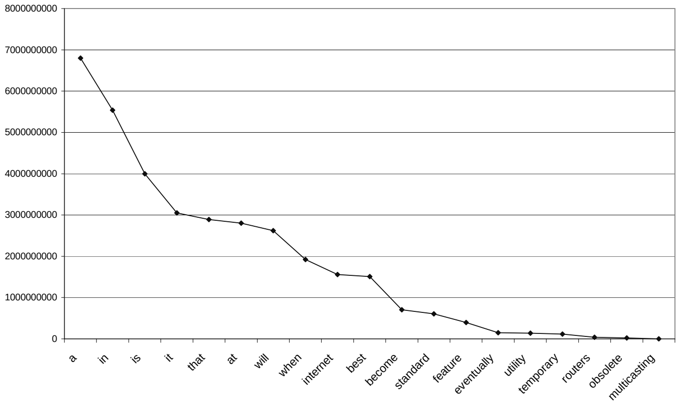


Fig. 5. Sharply declining word frequencies

Search on	Hits
“temporary utility”	about 21,000
“temporary utility that”	6
“router manufacturers”	about 27,000
“manufacturers happy”	about 2,300
“router manufacturers happy”	6

TABLE III

EXTENDING SEARCH STRING BY ONE WORD ELIMINATES ALL IRRELEVANT HITS FROM GOOGLE

STEPS IN WEBFERRET

Our aim is to search automatically for potential sources of plagiarism: we do this by passing search terms to an Internet server, and retrieving the lists of documents resulting from the search. Ferret provides us with *trigrams*, which we can use as search terms. However not all trigrams are equally likely to produce good results, so we adopt the rule that trigrams containing common words are more likely to occur by chance, and so exclude them. Actually, the evidence of the example above suggests that this may not be so, but as a general rule the assumption seems reasonable, especially when each document will contain several hundred trigrams.

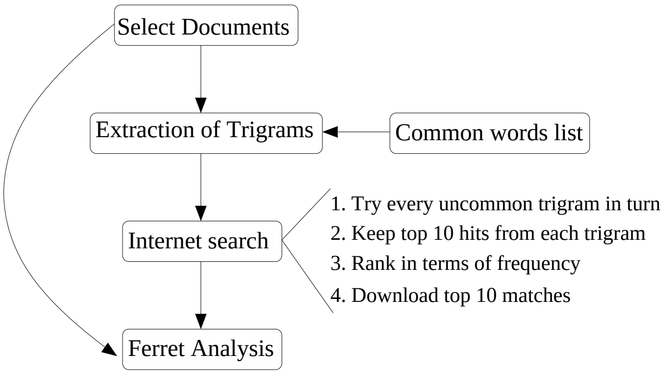

Select Documents
Extraction of Trigrams
Common words list
Internet search
1. Try every uncommon trigram in turn
2. Keep top 10 hits from each trigram
3. Rank in terms of frequency
4. Download top 10 matches
Ferret Analysis

Fig. 6. The process used by WebFerret

WebFerret works in the same manner as Ferret, except during the analysis step, see figure 6. First a list of suitable trigrams is extracted from the documents to be analysed. These trigrams are used to search the Internet for relevant sources, and the sources are downloaded into a folder. The potential sources are added to the set of documents to be compared (but note that potential sources are not compared with other potential sources). These additional steps are hidden from the user (except that they add to the processing time).

The following graph (figure 7) shows that most triples are rare in Google, so most occur in very few documents, so the strategy we adopt is likely to be successful in most cases.

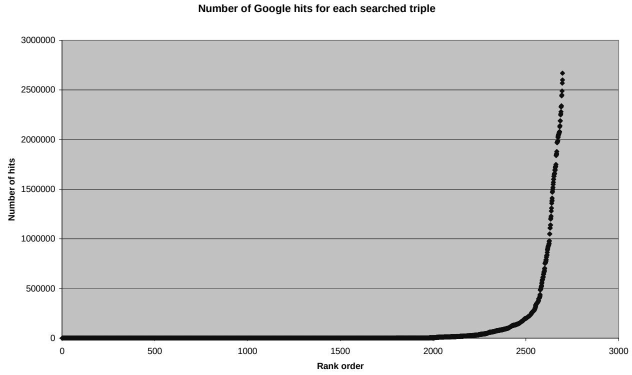


Fig. 7. Number of Google hits per triple

Effectiveness of WebFerret

In order to confirm the effectiveness of our strategy we took some documents (listed in table IV) and used WebFerret's algorithm to see if we could find them on the Internet.

File Name	Size (words)	Description of the document
ad-hoc-thesis.txt	488 words	student thesis, spellings corrected, titles removed [Studi]
education-essay.txt	907 words	copy of student work, not published on Internet [Malcolm]
heberling.txt	872 words	article on plagiarism [Heberling]
ryan-hamlin.txt	932 words	article on plagiarism (accused of copying heberling) [Ryan & Hamlin]
i33.txt	861 words	icmlc02 paper [Zhuang, Meng, Yin & Wang]
i367.txt	946 words	icmlc02 paper (very similar to i33) [Zhuang, Meng, Wang & Yin]
yip-stereo.txt	888 words	wmpmc paper [Yip et al.]

TABLE IV

Summary of test cases used

WebFerret's search process using the Google SOAP search API finds the target document in all cases.

Eliminating trigrams containing any one of the 116 very common words reduces the number of trigrams searched for by about 90%. This will give an order of magnitude speed-up, both on searching *and* on analysis.

Future Work

Once we have produced a trial version of WebFerret, we will get feedback on a number of questions:

- Does user need to alter search criteria?
- How much control should user have on search?
- Should old searches be kept?

The main outcome will be the WebFerret software system which may be installed on a user's own machine; Windows, Linux and Macintosh OS X versions will be created. We plan to develop a web interface also, so that organisations who want to can allow their users to upload student work and retrieve results over the Internet. A further benefit of the web interface is that the results of Internet searches by different staff members may be shared, within and across departments.

WebFerret will provide reports on the comparisons made, estimates of the amount of duplication present between pairs of documents, and detailed analyses of where copying has been found within each document. An evaluation of WebFerret's performance will be undertaken. Once WebFerret has been completed, we will speak to the colleagues managing our VLE: the ideal situation would be for WebFerret to integrate alongside the VLE, automatically producing feedback about potential plagiarism and collusion on submitted assignments.

Summary

Ferret helps staff by alerting them to similarity between pairs of documents; staff must make their own judgement as to whether the copying is "fair use" or "plagiarism".

WebFerret, like Ferret, will accept textual documents of many forms. Currently, documents generated by popular word processors, adobe PDF documents, and plain text files are supported. Because WebFerret will look at the content of the documents, it is not specialised towards any discipline.

Although initially we anticipate a number of cases of plagiarism being detected using the tool, we suggest a major enhancement for teaching and learning will be in deterrence. Knowing that work may be submitted to a fast and powerful plagiarism-detection tool will dissuade students from plagiarising.

But perhaps most useful is the use of such tools to educate students on good practice: highlighting that *this* block of text is copied and insufficiently referenced in a particular student's work can educate them much better than a general exhortation not to plagiarise. We hope that staff will be encouraged to use WebFerret because of its fast and simple interface and that students will be reassured that this emotive problem is being dealt with.

The current version of Ferret is freely available, and the authors welcome comments [9]. We are actively developing WebFerret based on the principles outlined in this paper, and aim to release it later this year.

Acknowledgements

Thanks are due to a University of Hertfordshire learning and teaching development fund grant for funding this work, and to Caroline Lyon, Bob Dickerson, Ruth Barrett and other members of the Plagiarism Detection Group for their input over the years.

References

[1] F. Culwin and T. Lancaster, "Plagiarism issues for higher education," *VINE, Volume 31, Number 2, pp. 36-41(6)*, 2001.

[2] M. Bell, "JISC Plagiarism Advisory Service Academic Integrity Survey," *cited in http://www.jiscpas.ac.uk/documents/FAQs.pdf*, 2006. [Online]. Available: www.jiscpas.ac.uk

[3] C. Lyon, R. Barrett, and J. Malcolm, "A theoretical basis to the automated detection of copying between texts, and its practical implementation in the ferret plagiarism and collusion detector," in *Plagiarism: Prevention, Practice and Policies Conference*, June 2004.

[4] C. Lyon and J. Malcolm, "Experience of plagiarism detection and prevention in higher education," in *Proceedings of the World Congress, Networked Learning in a Global Environment: Challenges and Solutions for Virtual Education*. ICSC-NAISO Academic Press, 2002.

[5] B. S. Glatt and E. H. Haertel, "The use of the cloze testing procedure for detecting plagiarism," *Journal of Experimental Education*, vol. 50, no. 3, pp. 127–136, 1982.

[6] R. Clarke and T. Lancaster, "Eliminating the successor to plagiarism? identifying the usage of contract cheating sites," in *Proceedings of JISC 2nd International Plagiarism Conference, Newcastle*, 2006.

[7] T. Jenkins and S. Helmore, "Coursework for cash: The threat from online plagiarism," in *Proceedings of 7th Annual Conference of the ICS HE Academy*, University of Leeds, 2006.

[8] C. Lyon, R. Barrett, and J. Malcolm, "Plagiarism is easy, but also easy to detect," *Plagiary: Cross-Disciplinary Studies in Plagiarism, Fabrication, and Falsification*, vol. 1, March 2006.

[9] "Plagiarism Detection Research Group," 2007. [Online]. Available: http://homepages.feis.herts.ac.uk/˜pdgroup/

[10] J. Bao, C. Lyon, and P. C. R. Lane, "Copy detection in Chinese documents using Ferret," *Language Resources and Evaluation*, vol. 40, no. 3-4, pp. 357–365, Dec 2006.

[11] J. Bao, C. Lyon, P. Lane, W. Ji, and J. Malcolm, "Comparing different methods to detect text similarity," Science and Technology Research Institute, University of Hertfordshire, Tech. Rep. 461, May 2007. [Online]. Available: http://homepages.feis.herts.ac.uk/˜comrcml/TR1-final.pdf

[12] C. Lyon, R. Barrett, and J. Malcolm, "Experiments in electronic plagiarism detection," Computer Science Department, University of Hertfordshire, Tech. Rep. 388, August 2003. [Online]. Available: http://homepages.feis.herts.ac.uk/˜comrcml/TR5.3.5.doc