Copy detection in Chinese documents using the Ferret: a report on experiments

JunPeng Bao (baojp@mail.xjtu.edu.cn)
Department of Computer Science & Technology, Xi'an Jiaotong University, Xi'an 710049, China

Caroline Lyon (c.m.lyon@herts.ac.uk),
Peter C. R. Lane (peter.lane@bcs.org.uk),
Wei Ji (w.1.ji@herts.ac.uk) and
James A. Malcolm (j.a.malcolm@herts.ac.uk)
School of Computer Science, University of Hertfordshire, College Lane, HATFIELD AL10 9AB, Hertfordshire, UK

Abstract. The Ferret copy detector has been used for some years on English texts to find plagiarism in large collections of students' coursework. This article reports on extending its application to Chinese, which differs from English in many respects: the sequence of characters that make up a Chinese text do not have word boundaries marked, there is a vast Chinese "alphabet", or number of different characters, and they are represented with multi-byte encoding. We discuss issues of representation, focus on the effectiveness of a sub-symbolic approach, and show how the Ferret can circumvent the classic problem of finding word boundaries with an automated system. Corpora of students' coursework from two Chinese universities have been collected, and we apply Ferret to investigate the detection of plagiarism. Our experiments show that Ferret can find both artificially constructed plagiarism as well as actually occurring, previously undetected plagiarism. We also investigate the parameters of the system, and report on typical optimum settings. Experiments reported in this article show that Ferret can work well on Chinese texts, and achieve a consistent performance. The investigation into the representation of written Chinese is likely to be of use in other language processing tasks.

Keywords: Chinese processing, copy detection, Ferret, plagiarism

1. Introduction

Detecting the presence of copied material in documents is a problem confronting many disciplines. In education, students may copy from each other or web sources in an attempt to plagiarise or collude their assessed work. In the commercial world, copying is found in theft of copyright or intellectual property. Detecting copied, or duplicated, material is also of importance in managing language resources, to locate and highlight links between related documents. As much copying is performed at a simple, lexical level, it is appropriate to consider

how computer-based tools can be used to determine which pairs of documents contain high levels of copying.

Ferret (Lyon et al., 2004; Lyon et al., 2001; Lyon et al., 2006) is a tool for detecting similar passages of text in large collections of documents. It has been used on English texts for some years, and also successfully found copied material in Dutch field trials. Ferret is a free, stand alone system designed to be run by users of various backgrounds on their own PCs, giving immediate results; a recent implementation is discussed in Lane, Lyon and Malcolm (2006). Ferret is useful for analysing coursework or essays from a large cohort of students given the same task, in order to detect collusion and some limited plagiarism of web-based material. It can also be used to analyse programming code, and effectively identifies plagiarism in students' programs.

Turnitin (2006) is one of the better known systems for copy detection, using an enormous database of material off the web and previous student work, against which it compares current student work. Documents have to be submitted for processing, and there is a commercial charge. A report on the complementary roles of Ferret and Turnitin is given by Lyon, Barrett and Malcolm (2003).

These tools, although popular, have not, to date, been applied to Asian languages, which have different written forms and structure to European languages. We made the hypothesis that Chinese plagiarism could be detected by Ferret, using the same underlying principle as is applied to English. In this article, we show how this can be done, and present experimental results demonstrating the effectiveness of copy detection with Ferret on Chinese.

Although copying is part of the definition of plagiarism, some copying is perfectly legitimate if it is correctly quoted and cited. Further work with the Ferret will look into the detection of more sophisticated program copying and code cloning (Malphol, 2006). Alternative approaches look at semantic similarities between pairs of documents (Bao et al., 2006; Bao et al., 2004a; Bao et al., 2004b). However, a limitation of most work to date has been its focus on English texts; we are not aware of any working automated system for detecting copied material in Chinese documents.

1.1. The representation of written Chinese

Chinese texts differ from English in many respects: the sequence of characters that make up a Chinese text do not have marked word boundaries, there is a vast number of different characters, and they are represented in computer files with a multi-byte encoding. However, both languages share a crucial characteristic: they are sequences of

discrete data. In English the data items are words, while in Chinese they are characters. That is to say, a text in either language can be converted into a sequence of tokens. We can then apply the same principle to detect copied material no matter which language the text is written in.

Using this approach, Ferret works on both English and Chinese documents, as well as those Chinese documents that have English words and phrases inserted, a phenomenon that is quite common in some domains. The development of representations for written Chinese may also be of use in other language processing tasks.

This article reports that Ferret performs effectively on Chinese texts. Corpora of students' coursework from two Chinese universities have been collected, and we apply Ferret to investigate the detection of plagiarism. Our experiments show that Ferret can find both artificially constructed plagiarism as well as actually occurring, previously undetected plagiarism.

The rest of this article is organised in the following way. Section 2 introduces the Ferret system in more detail. Section 3 discusses the problem of finding words in Chinese, and presents several strategies for representation. Section 4 describes the data collected, and reports on experiments. It discusses the setting of parameters and finding optimal values. Section 5 discusses the influence of different representational strategies on copy detection and concludes the article.

2. Outline of the Ferret system

The Ferret copy detector takes a set of files and compares each one with each other to get a measure of similarity. The first stage in the process is to convert each document to a set of overlapping trigrams. Thus, a sentence like:

A storm is forecast for the morning.

will be converted to the set of trigrams:

```
a storm is      storm is forecast     is forecast for
forecast for the        for the morning
```

Then the set of trigrams for each document is compared with all the others, and a measure of resemblance for each pair of documents is computed. Usually, the results are presented in a ranked table with the most similar pairs at the top. Any pair of documents can be displayed and compared side by side with matching passages highlighted.

If two documents are written independently there will be a sprinkling of matching trigrams, but if there has been collusion or copying there will be solid passages that are all or mostly highlighted indicating a quantity of matching word sequences. The similarity measure still records a significant value if a few words are replaced, deleted or inserted. For instance, if a word is altered in the example above, the sentence

A gale is forecast for the morning.

will still have 3 of the 5 trigrams matching.

Trigrams were found to be the best size tuple in earlier experiments in English, giving greatest discrimination between copied and non-copied texts. With bigrams many false positive matches were produced; with longer tuples more matches were missed through the alteration of a few words. For blatant copying with no alterations this would not matter, but in practice it has been found that students typically make some small changes.

2.1. The Resemblance metric

We use a measure of similarity known as the *Resemblance metric*; this metric is taken from work by Broder (1998), and is also known in the area of feature-vector analysis as the Jaccard coefficient (Manning and Schütze, 2001, page 299). Informally, the measure compares the number of matches between the elements of two sets of trigrams, scaled by joint set size.

Let S(A) and S(B) be the set of trigrams from documents A and B respectively. Let R(A,B) be the resemblance between A and B.

$$R = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|} \tag{1}$$

$0 \leq R \leq 1$

Two identical documents have an R-score of 1. Previous work (Lyon et al., 2001) has shown that scores above 0.04 are typically indicative of copying. However, this does vary with domain. We show later some experimental results from a technique which determines this value from a sample of the dataset.

3. Adapting Ferret for Chinese documents

The Ferret system was created to work on English text. The definition of trigrams is based on words in English. We can adapt Ferret to

work on different kinds of text by basing the definition of a trigram on different kinds of *token*. Such an approach has already been shown to work for analysing computer programs (Lane et al., 2006). In this section, we show how tokens can be defined so that the Ferret may be applied to texts consisting of Chinese characters.

3.1. Automated detection of word boundaries in Chinese

As is well known, Chinese words usually consist of one, two or up to four characters, with no white space or other marker between words. The best way of finding words accurately in a Chinese sentence is still an open issue. We consider several strategies to process the strings of characters that make up a Chinese sentence. They are the "naive" strategy, the dictionary strategy, and the single character strategy.

Naive strategy: Sequences of Chinese characters are segmented by taking as a "word" boundary any element that is not a Chinese character: white space, punctuation, numbers etc. These segments are taken as "words".

Dictionary strategy: Based on a Chinese dictionary, a sentence is separated into a sequence of Chinese words in which each word can be found in the dictionary. One example of this approach is described in detail by Gao et al. (2006), who identify five types of words:

- Entries in the lexicon
- Morphologically derived words
- Factoids - composite elements such as times and dates
- Named entities - names of people, places, organisations etc.
- New words. In spite of using a lexicon with over 98K entries, new words occur.

In fact there should also be a further entry, since it is quite common to find a sprinkling of English words in Chinese documents. So we should have also have the type:

- Foreign words using the Roman alphabet.

However, Gao et al. say "We do not intend ... to give a standard definition of Chinese words. Instead, we treat Chinese word segmentation as a preprocessing step where the best *segmentation units* depend on how they are used in the consuming applications." Thus we come to the final strategy considered here, which is appropriate for use with the Ferret.

(a)

TSP问题，即Traveling Salesman Problem，是一个NP问题。

(b)

"TSP问题" "即Traveling" "Salesman" "Problem" "是一个NP问题"

(c)

"问" "题" "即" "是" "一" "个" "问" "题"

(d)

"问题" "即" "是" "一个" "问题"

(e)

"TSP" "问" "题" "即" "Traveling" "Salesman" "Problem" "是" "一" "个"
"NP" "问" "题"

Figure 1. A Chinese sentence (a) with its words parsed with different strategies: (b) using naive strategy, (c) using single-character strategy, (d) using dictionary strategy, and (e) using mixed strategy.

Single character strategy: Instead of finding words, characters are processed singly. In essence, each individual character in the text file is treated as a token.

3.2. Examples

As an example of these approaches see Figure 1(a) which shows a Chinese sentence. In English this means "TSP is an NP problem" (TSP means the Travelling Salesman Problem). Using the naive strategy, we get three Chinese tokens in the sentence because it is segmented by two punctuation marks, as shown in Figure 1(b). With the single character strategy, we get 8 Chinese tokens because there are 8 Chinese characters in it, as shown in Figure 1(c). With the dictionary strategy, we get 5 Chinese words as tokens, as shown in Figure 1(d).

3.3. Defining a token

As mentioned above, Ferret considers a document as a sequence of tokens. A token can differ depending on the context: it can be a word, a symbol, a phrase or a Chinese character. We have adapted Ferret to process three types of documents, using the same core algorithm for

detecting similar passages: the types are typeText, typeChinese, and typeMix, described as follows:

typeText A token is a sequence of consecutive alphabetic characters, such as English words, with boundaries marked by white space or punctuation marks. For Chinese documents we can take Chinese characters in a similar way for the naive strategy. This is illustrated in Figure 1(b). We refer to the Ferret system using this type as Ferret_T.

typeChinese A token is a single Chinese character without any other symbols. Chinese characters are processed singly and any alphabetic characters are ignored. This is illustrated in Figure 1(c). Obviously, this type can only be applied to Chinese documents. We refer to the Ferret system using this type as Ferret_C.

typeMix A token is either a sequence of consecutive Roman alphabetic characters, such as an English word, or a single Chinese character. This type of mixed text with a few foreign terms is commonly found in modern Chinese documents, especially in scientific literature. This is illustrated in Figure 1(e). We refer to the Ferret system using this type as Ferret_M.

In the case of typeMix, Ferret combines the naive strategy and single character strategy so that it processes English text with the naive strategy and Chinese with the single character strategy. That enables Ferret to avoid missing out English words in a Chinese document. For example, Figure 1(a) is a Chinese sentence including English words. Figure 1(c) shows that treating the sentence as typeChinese loses some words, and may lead to potential errors. In the case of typeText, as shown in Figure 1(b), no words are lost but a token may be too long to accord with the Ferret philosophy. A long token may lead to less discriminative power, since a single change in a string will mean there will be no match between 2 similar strings, even if parts are in fact the same. For blatant copying, with no attempt at disguise, this will not matter. Figure 1(e) shows the tokens parsed from the sentence shown in Figure 1(a) as typeMix.

We do not use the dictionary strategy in the experiments described here.

4. Experiments

4.1. Chinese corpora

We have run experiments on 2 raw Chinese corpora. One is named Stu04Rpt, and the other is named Gu05. Stu04Rpt is a collection of individual pieces of coursework submitted in 2004 by students at Xi'an Jiaotong University, China. The reports were collected by the first author, and consist mostly of reports on artificial intelligence topics. Gu05 was collected by Leonard Gu from University of Shanghai Electric Power, China, in which most documents are student groups' reports written in 2005 on solving mathematical questions with the help of computer programs and the technical software package, MATLAB. This is a high-level language and interactive environment widely used by technicians and researchers. There is a slightly different specification document for each group to explain. In both cases the raw materials are MSWord files. The first stage in processing with Ferret is to convert these .doc files to .txt. We use Antiword `http://www.win_eld.demon.nl/` to convert them into plain texts in UTF-8 encode.

4.1.1. *Creation of pseudo-plagiarised texts*

52 unique documents are selected from Stu04Rpt, from which 156 plagiarised documents are made artificially by means of copy, cut, paste, and mix actions. Each unique document is chopped into sections of the same size except the last one. Then a new plagiarised document is made of mixed sections that are randomly selected from several documents. The size of each copied section, called as a copied unit, varies from about 50 characters to 500 characters. Hence, we get a corpus including pseudo-plagiarised documents named as Stu04Rpt_Pn, where n indicates the minimum size of each copied unit in characters, giving an indication of the level of plagiarism because the plagiarised sections are known. Table I shows the details of these corpora. Each pseudo-plagiarised file has one or more copied units. (The numbers in Tables IX and X should be interpreted as a consequence of this method of creating pseudo-plagiarised texts).

4.2. Detecting naturally occurring plagiarism

Our first experiment explores the effect of the different strategies and document types for processing unsegmented strings of Chinese characters. The three versions of Ferret are named Ferret_T, Ferret_C, and Ferret_M respectively (see above for the definition of these three types).

We processed the complete set of documents for the two corpora, Stu04Rpt and Gu05, with the three forms of Ferret, and recorded the

Table I. Details of the corpora

Corpus	Total files	Number of tokens*			Pseudo-plagiarism	Total plagiarised document pairs
		Average	Max	Min		
Stu04Rpt	320	4136	25474	104	No	N/A
Gu05	124	1125	21762	102	No	N/A
Stu04Rpt_P50	156	4600	13756	191	Yes	1031
Stu04Rpt_P100	156	4740	13756	384	Yes	1083
Stu04Rpt_P200	156	5023	13756	694	Yes	1104
Stu04Rpt_P300	156	5321	13756	1337	Yes	1153
Stu04Rpt_P400	156	5579	13756	1448	Yes	1146
Stu04Rpt_P500	156	5801	13756	1448	Yes	1188

* A token is a single Chinese character or an English word.

number of times the Resemblance metric for a pair of documents fall within a range $[a, b)$, where a and b are numbers between 0 and 1, and a number r falls within the range $[a, b)$ if $a \leq r < b$. Figure 2 illustrates the Ferret Resemblance scores distribution on Stu04Rpt in a histogram; similar results were obtained for the Ferret resemblance scores distribution on Gu05, which is illustrated by Figure 3. Table II and table III list the details.

Having run these experiments we initially inspected samples manually to see if the results matched our subjective judgements. This experiment shows that most of the pairs of documents produce a low similarity score, but a significant minority have higher scores. This is particularly clear for Ferret_T, where most items have a score between 0 and 0.01. For Ferret_C and Ferret_M, most items have a score less than 0.04. The long trigrams in Ferret_T are less likely to match those in other independently written documents than the shorter ones using other strategies. However, see the final section for a discussion of the advantages and disadvantages of this.

Ferret also finds document pairs whose resemblance score is 1 in both Stu04Rpt and Gu05, which implies that some documents are identical, except for non-alphabetic characters such as punctuation. We checked all of these document pairs manually and confirm that they are really identical. In terms of document pairs whose scores are 1, Ferret_C and Ferret_M have the same result throughout However, Ferret_T differs in one instructive case on Stu04Rpt. The difference between R181.txt and R63.txt in the Stu04Rpt is only a dot character as shown in Figure 4. Hence, although the Ferret_C and Ferret_M scores are 1, the Ferret_T score is 0.892857. (In this case the trigram count is 131 for both files

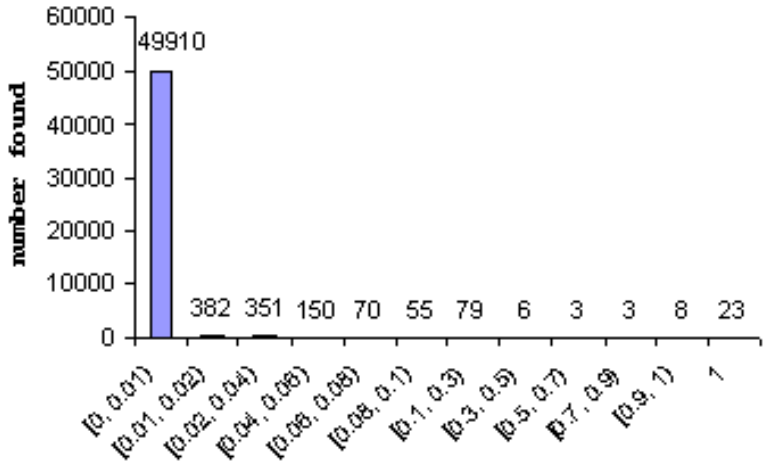
Table II. The Ferret resemblance scores distribution on Stu04Rpt

Score	Ferret_T		Ferret_C		Ferret_M	
interval	count	proportion	count	proportion	count	proportion
$[0, 0.01)$	49910	0.977861	15205	0.297904	15627	0.306172
$[0.01, 0.02)$	382	0.007484	12503	0.244965	13316	0.260893
$[0.02, 0.04)$	351	0.006877	18628	0.364969	17692	0.34663
$[0.04, 0.06)$	150	0.002939	2451	0.048021	2253	0.044142
$[0.06, 0.08)$	70	0.001371	741	0.014518	676	0.013245
$[0.08, 0.1)$	55	0.001078	396	0.007759	399	0.007817
$[0.1, 0.3)$	79	0.001548	1010	0.019788	972	0.019044
$[0.3, 0.5)$	6	0.000118	59	0.001156	60	0.001176
$[0.5, 0.7)$	3	5.88E-05	9	0.000176	8	0.000157
$[0.7, 0.9)$	3	5.88E-05	5	9.8E-05	4	7.84E-05
$[0.9, 1)$	8	0.000157	9	0.000176	9	0.000176
1	23	0.000451	24	0.00047	24	0.00047
total	51040	1	51040	1	51040	1

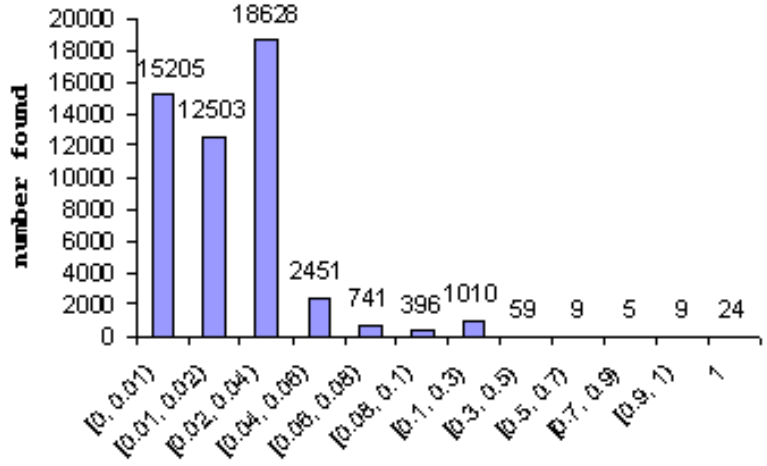
Table III. The Ferret resemblance scores distribution on Gu05

Score	Ferret_T		Ferret_C		Ferret_M	
interval	count	proportion	count	proportion	count	proportion
$[0, 0.01)$	6036	0.791503	2730	0.357986	2903	0.380671
$[0.01, 0.02)$	332	0.043535	964	0.12641	964	0.12641
$[0.02, 0.04)$	153	0.020063	1377	0.180566	1465	0.192106
$[0.04, 0.06)$	57	0.007474	824	0.108051	725	0.095069
$[0.06, 0.08)$	44	0.00577	399	0.052321	282	0.036979
$[0.08, 0.1)$	18	0.00236	161	0.021112	118	0.015473
$[0.1, 0.3)$	101	0.013244	302	0.039601	298	0.039077
$[0.3, 0.5)$	604	0.079203	448	0.058746	505	0.066221
$[0.5, 0.7)$	254	0.033307	370	0.048518	321	0.042093
$[0.7, 0.9)$	19	0.002491	38	0.004983	33	0.004327
$[0.9, 1)$	1	0.000131	6	0.000787	5	0.000656
1	7	0.000918	7	0.000918	7	0.000918
total	7626	1	7626	1	7626	1

(a) Ferret_T



(b) Ferret_M



(c) Ferret_C

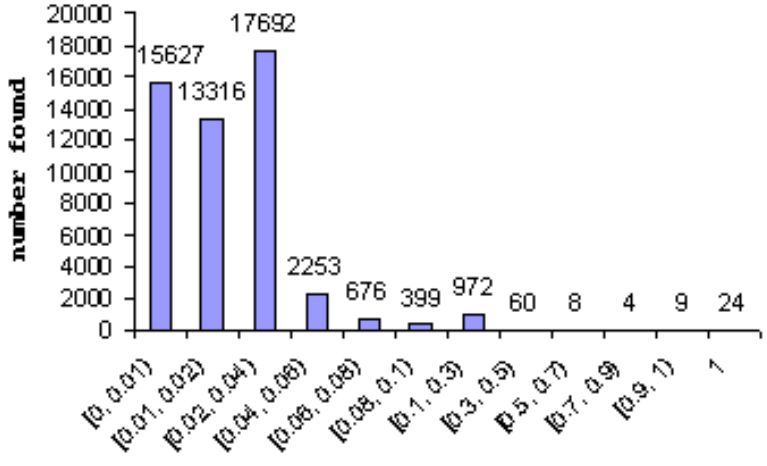
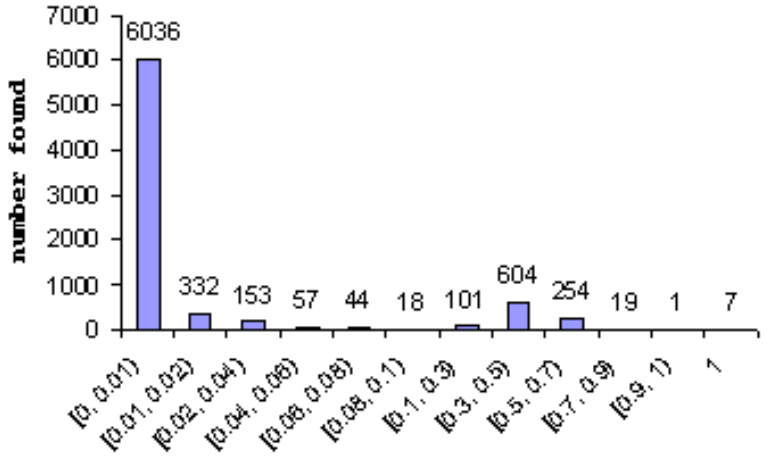
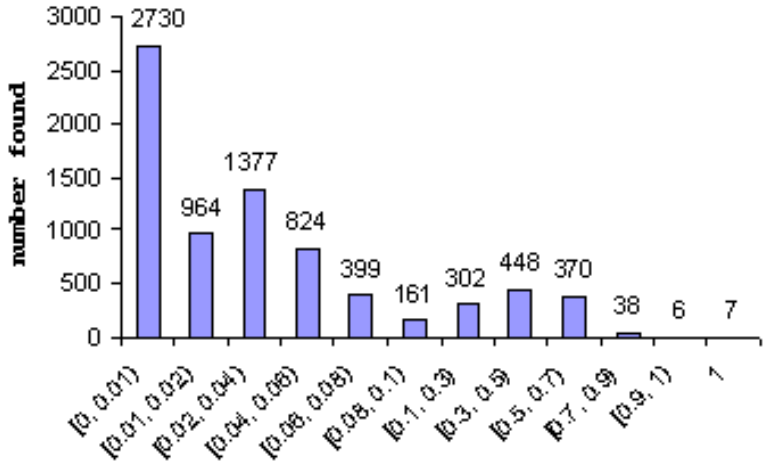


Figure 2. Ferret resemblance scores distribution on Stu04Rpt, out of 51,040 pairs. (Note that the horizontal axis varies.)

(a) Ferret_T



(b) Ferret_M



(c) Ferret_C

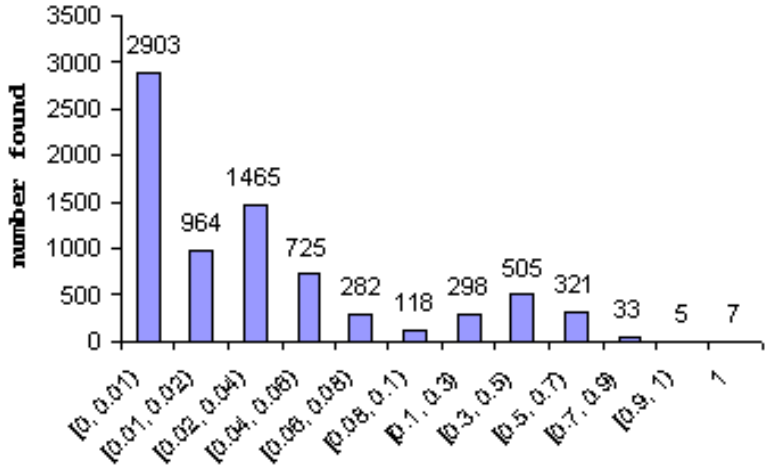


Figure 3. Ferret resemblance scores distribution on Gu05, out of 7,626 pairs. (Note that the horizontal axis varies.)

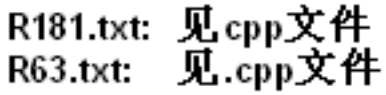


Figure 4. The only difference between document R181.txt and R63.txt in Stu04Rpt

with Ferret_C, 138 for both files with Ferret_M; for Ferret_T with its long tokens there are 26 trigrams in one file, 27 in the other, of which 25 are common.)

This difference is a consequence of the different Chinese separating strategies. As mentioned above, Ferret_T uses the naive strategy, and considers the sentence in R181.txt as a single token, but considers the sentence in R63.txt as two tokens, because the dot character indicates the end of a sentence. Ferret_C uses the single-character strategy, and ignores all non-Chinese characters, so it considers the two sentences as the same. Ferret_M combines the naive strategy and the single-character strategy, and separates Chinese characters from English characters automatically, so it parses the same four tokens from the partial documents shown in Figure 4.

Ferret_T has a different score distribution on the same corpus. However, when the score is greater than 0.9, the rank of all 3 Ferret types are almost the same. When it is less than 0.9, the rank of Ferret_T is different from Ferret_C and Ferret_M increasingly. As expected, the rank of Ferret_C is always similar to that of Ferret_M, since most of the documents in Stu04Rpt are mainly composed of Chinese characters mixed with just a few English words.

The documents in Gu05 are reports on applying MATLAB to some mathematical problems. They have a very similar format. Since the problems are the same type in principle, the algorithm and program codes in the reports have similar contents to some extent. That leads to the greater Ferret score on Gu05. We notice again that Ferret_T has different distribution from Ferret_C and Ferret_M, and that is discussed below in section 5.

The Stu04Rpt is collected from individual students' coursework. Unfortunately, some students have done their homework together, and copied each other. Furthermore, their report topics are limited to a few areas, such as knowledge representation, expert system, computer vision, pattern recognition, and game theory. Many students consulted the same references and websites so that identical contents are quoted in their reports.

Inspection of random samples of pairs indicated that there was significant plagiarism where the score was greater than 0.7, and no

plagiarism where the score was less than 0.01. As in English, we found that all texts had a sprinkling of matching trigrams, even when they were independently written. In the following sections we investigate these limits further.

4.3. The optimum threshold

In the section above, we described how Ferret can find unknown plagiarised documents. We can be sure that two documents are very similar when the Ferret score is very high (greater than 0.9). But in practice many plagiarised documents copy part of their contents from others, not the whole paper, so that their Ferret scores are not so high. Furthermore, long documents with short copied passages will have lower resemblance scores so that Ferret needs a lower threshold to detect them. The optimum threshold for Ferret has to be fixed empirically. In the second set of experiments, we try to find a typical optimum threshold for Ferret based on our Chinese corpora, in which document length is in the range of 100 to 25,000 characters, corresponding to typical coursework bounds, as shown in Table I. The optimum threshold means the best threshold applied to our data; results on other corpora are likely to vary. We explain how customised thresholds can be set below.

In order to detect plagiarised documents efficiently, a series of artificially constructed corpora Stu04Rpt_Pn are used initially to determine parameters of Ferret. These have known copied passages. We do not take into our calculations the other naturally occurring plagiarism.

We compute three measures to determine the performance of Ferret: precision (P), recall (R) and F1. The precision is the proportion of plagiarised pairs detected by Ferret which are indeed plagiarised. The recall is the proportion of the plagiarised pairs which Ferret detects. F1 is a standard metric commonly used to take into account both precision and recall, which may have opposing tendencies. Specifically, if p represents precision and r recall, then

$$F1 = \frac{2 \times p \times r}{p + r}$$

We can make Ferret categorise the pair as containing copied passages by setting a threshold value, θ. Any pair of documents whose resemblance score exceeds that threshold will be said to contain copied material. The optimum value for the threshold will be determined as that which leads to the greatest F1 value.

Table IV shows the greatest F1 value of Ferret on the corpora. Table V shows the trends of Ferret precision, recall, and F1 for different

Table IV. The maximum F1 values for corpora with different amounts of copied material. (F1 is the F1 score, P precision, R recall, and θ the threshold.)

Corpus	Ferret_T			
	F1	P	R	θ
Stu04Rpt_P50	0.59	0.98	0.42	0.01
Stu04Rpt_P100	0.85	0.97	0.76	0.01
Stu04Rpt_P200	0.95	0.96	0.93	0.01
Stu04Rpt_P300	0.97	0.95	0.99	0.01
Stu04Rpt_P400	0.97	0.94	1.00	0.01
Stu04Rpt_P500	0.98	0.99	0.97	0.02
	Ferret_C			
Stu04Rpt_P50	0.30	0.66	0.20	0.05
Stu04Rpt_P100	0.51	0.53	0.49	0.04
Stu04Rpt_P200	0.73	0.84	0.65	0.05
Stu04Rpt_P300	0.83	0.87	0.80	0.05
Stu04Rpt_P400	0.87	0.87	0.88	0.05
Stu04Rpt_P500	0.90	0.89	0.92	0.05
	Ferret_M			
Stu04Rpt_P50	0.30	0.41	0.24	0.04
Stu04Rpt_P100	0.52	0.57	0.48	0.04
Stu04Rpt_P200	0.74	0.88	0.64	0.05
Stu04Rpt_P300	0.85	0.90	0.80	0.05
Stu04Rpt_P400	0.88	0.90	0.87	0.05
Stu04Rpt_P500	0.92	0.91	0.92	0.05

thresholds on the Stu04Rpt_P500, which are very similar to the trends on other corpora so that we do not list them all in this paper.

From Table V, we can see that as expected the precision increases and recall declines as the threshold increases, until precision reaches 1. The F1 value of Ferret_T reaches a maximum around 0.01 to 0.02 as shown in Table IV. Ferret_C and Ferret_M reach a peak around 0.04 to 0.05. It demonstrates that Ferret can find copied material with both high precision and recall around that threshold. The threshold can be increased to improve precision at the price of reducing recall.

Based on these data, the typical optimum threshold of Ferret_T on Chinese document is around 0.01-0.02, that of Ferret_C and Ferret_M are around 0.04-0.05. We see that the F1 score for Ferret_T is higher than the others, particularly for smaller amounts of copied text. With

Table V. Plagiarism detection for different thresholds on Stu04Rpt_P500. (F1 is the F1 score, P precision, and R recall.)

Threshold	Ferret_T			Ferret_C			Ferret_M		
θ	P	R	F1	P	R	F1	P	R	F1
0.01	0.96	0.99	0.98	0.10	1.00	0.18	0.10	1.00	0.19
0.02	**0.99**	**0.97**	**0.98**	0.14	1.00	0.25	0.15	1.00	0.26
0.03	1.00	0.87	0.93	0.33	0.98	0.49	0.37	0.99	0.53
0.04	1.00	0.72	0.84	0.65	0.97	0.78	0.69	0.973	0.81
0.05	1.00	0.62	0.76	**0.89**	**0.92**	**0.90**	**0.91**	**0.92**	**0.92**
0.06	1.00	0.55	0.71	0.98	0.83	0.90	0.99	0.83	0.90
0.07	1.00	0.49	0.66	1.00	0.72	0.84	1.00	0.72	0.83
0.08	1.00	0.44	0.61	1.00	0.63	0.77	1.00	0.62	0.76
0.09	1.00	0.38	0.55	1.00	0.54	0.70	1.00	0.53	0.69

the shorter tokens used in Ferret_C and Ferret_M there will be some naturally occurring matches in non-copied text, whereas there is much less likely to be a match with the longer token in Ferret_T, so the threshold can be lower. This suggests that the longer segments using the naive strategy may be more useful, but in practice it may not be the case. When there is an attempt to deceive there may be a number of minor changes that undermine the use of the longer token. See further comments in Section 5.

4.4. Setting thresholds

Tables VI, VII, and VIII illustrate the consistency of the Ferret optimum threshold across different sized document sets. This shows that customised thresholds can be set by analysing a small sample of a large set of documents. These tables list the optimum threshold of Ferret on different size corpora, consisting of randomly selected documents from each Stu04Rpt_Pn corpus. The experiments are repeated 10 times, each time with a different random selection of documents to find the threshold that leads to the greatest F1 value on that corpus. The tables report the average maximum value of F1. We see that the smaller the size of a corpus, the wider the range of the optimum threshold. The bigger the corpus, the more stable the optimum threshold is. The optimum threshold of Ferret_T is around 0.01-0.02, and that of Ferret_C and Ferret_M are around 0.04-0.05.

Another trend is that the greatest F1 value on each Stu04Rpt_Pn increases while n increases: the more copied material there is, the easier it is to detect. Ferret performs less well when the copied unit is small.

Table VI. The optimum threshold of Ferret_T on different size of corpora.

Corpus	20 documents		50 documents		100 documents		156 documents	
Stu04Rpt_	θ	F1	θ	F1	θ	F1	θ	F1
P50	0.01	0.60	0.01	0.59	0.01	0.57	0.01	0.59
P100	0.015	0.84	0.01	0.85	0.01	0.85	0.01	0.85
P200	0.02	0.96	0.01	0.94	0.01	0.94	0.01	0.95
P300	0.015	0.98	0.015	0.97	0.01	0.97	0.01	0.97
P400	0.02	0.99	0.015	0.98	0.015	0.97	0.01	0.97
P500	0.015	0.99	0.015	0.98	0.02	0.98	0.02	0.98

Table VII. The optimum threshold of Ferret_C on different size of corpora.

Corpus	20 documents		50 documents		100 documents		156 documents	
Stu04Rpt_	θ	F1	θ	F1	θ	F1	θ	F1
P50	0.05	0.38	0.035	0.29	0.056	0.30	0.05	0.30
P100	0.055	0.61	0.045	0.52	0.045	0.51	0.04	0.51
P200	0.05	0.77	0.045	0.71	0.05	0.74	0.05	0.73
P300	0.05	0.83	0.055	0.83	0.055	0.84	0.05	0.83
P400	0.06	0.89	0.055	0.89	0.055	0.87	0.05	0.87
P500	0.055	0.93	0.055	0.90	0.055	0.91	0.05	0.90

According to the data listed in Table IV, the best F1 value of Ferret_T is lower than 0.6 when the copied unit is about 50 characters, and that of Ferret_C and Ferret_M are lower than 0.6 when the copied unit is about 100 characters. More experiments to discover limits are described in the next section.

Table VIII. The optimum threshold of Ferret_M on different size of corpora.

Corpus	20 documents		50 documents		100 documents		156 documents	
Stu04Rpt_	θ	F1	θ	F1	θ	F1	θ	F1
P50	0.05	0.40	0.035	0.30	0.05	0.30	0.04	0.30
P100	0.055	0.62	0.05	0.53	0.045	0.52	0.04	0.52
P200	0.05	0.80	0.05	0.72	0.045	0.74	0.05	0.74
P300	0.05	0.83	0.055	0.84	0.05	0.84	0.05	0.85
P400	0.055	0.91	0.055	0.89	0.055	0.87	0.05	0.88
P500	0.055	0.95	0.055	0.91	0.05	0.92	0.05	0.92

4.5. The lower limit for detection

It is natural that small pieces of copying are hard to detect. In this section we try to find the lower limit for detecting copied passages in Chinese. First we count the total number of tokens in copied units in each document pair. Recall that each pseudo-plagiarised file has one or more copied units. Table IX lists the distribution of how many known plagiarised document pairs there are in each corpus in terms of the number of copied tokens in a pair. We set the Ferret threshold around the optimum threshold mentioned above, and count how many known plagiarised document pairs can be found in different intervals. Tables X and XI show the level of recall on document sets with different amounts of plagiarism (the results for Ferret_C are very similar to those for Ferret_M). This is calculated as the number of found plagiarised document pairs whose score is equal to or higher than the threshold, divided by the number of known plagiarised document pairs in the interval. Note the 'N/A' values indicate that there was no applicable dataset for the given size.

Ferret finds nearly all of the plagiarised document pairs that contain more than 1000 copied tokens, and most of them when the number of the copied tokens is between 500 and 1000. When the number of copied tokens is between 300 and 500, Ferret_T is still able to find most of them, but Ferret_C and Ferret_M fail to find nearly half of them, which fall below the threshold. When the number is less than 300, it is hard for Ferret to find most of them. It seems that 500 tokens is the lower limit for Ferret_C and Ferret_M on these data at the optimum threshold around 0.05, which account for about 10% tokens of a document (i.e. 5% of a document pair) in our corpora. Ferret_T has a slightly lower limit at the optimum threshold around 0.01, which is about 300 tokens. (This contrasts with the level at which copying is detected in English, which is typically about 3-4% of words (Lyon et al., 2003, Section 5.3), in documents 10,000 words long.) Thus Ferret can detect plagiarised documents with a high probability as long as the size of the copied content in them is greater than the lower limit, but it is likely to miss them if the size is less than the lower limit.

Since Ferret score may vary with document size. Copy ratio, which is defined as the ratio of the size of copied content to the size of the whole document pair, indicates the lower limit in another point of view. Table XII lists the distribution of plagiarized document pairs among different copy ratios in each corpus. Table XIII, XIV, and XV list the recall of Ferret on this distributions around the optimum threshold. These data illustrate that Ferret can find nearly all plagiarized document pairs when the copy ratio is greater than 0.1, and Ferret can still find most

Table IX. Distribution of pseudo-plagiarised document pairs (described in Section 4.1.1) among different copy sizes in a corpus.

Num of copied chars in a pair	Stu04Rpt_ P50	P100	P200	P300	P400	P500
<100	323	0	3	0	0	0
[100, 300)	542	532	382	0	2	3
[300, 500)	10	304	169	401	415	0
[500, 1000)	0	91	308	354	173	433
≥ 1000	156	156	242	398	556	752
total	1031	1083	1104	1153	1146	1188

Table X. Recall of Ferret_T on different amounts of copied material around the optimum threshold.

Stu04Rpt_	Threshold	<100	[100, 300)	[300, 500)	[500, 1000)	≥ 1000
P50	0.01	0.16	0.40	N/A	N/A	1.00
	0.02	0.10	0.07	N/A	N/A	1.00
P100	0.01	N/A	0.53	0.97	1.00	1.00
	0.02	N/A	0.29	0.64	0.93	1.00
P200	0.01	N/A	0.82	0.99	1.00	1.00
	0.02	N/A	0.50	0.83	0.98	1.00
P300	0.01	N/A	N/A	0.97	1.00	1.00
	0.02	N/A	N/A	0.67	0.99	1.00
P400	0.01	N/A	N/A	1.00	1.00	1.00
	0.02	N/A	N/A	0.84	1.00	1.00
P500	0.01	N/A	N/A	N/A	0.97	1.00
	0.02	N/A	N/A	N/A	0.94	1.00

plagiarized document pairs whose copy ratio is between 0.05 and 0.1. When the document pair's copy ratio is less than 0.05, Ferret may miss it. Since the average size of a document in our copora is about 5000 tokens, copy ratio 0.05 implies that there are about 500 tokens copied between the document pair.

We checked all of the document pairs that contain more than 1000 copied tokens but fail to be detected by Ferret, and found that they are all related to 4 documents (i.e. R124.txt, R34.txt, R176.txt, and

Table XI. Recall of Ferret_M on different amounts of copied material around the optimum threshold

Stu04Rpt_	Threshold	< 100	$[100, 300)$	$[300, 500)$	$[500, 1000)$	≥ 1000
P50	0.04	0.14	0.09	N/A	N/A	1.00
	0.05	0.09	0.02	N/A	N/A	1.00
	0.06	0.03	0.01	N/A	N/A	1.00
P100	0.04	N/A	0.25	0.50	0.86	1.00
	0.05	N/A	0.17	0.28	0.68	1.00
	0.06	N/A	0.09	0.18	0.36	1.00
P200	0.04	N/A	0.46	0.76	0.96	1.00
	0.05	N/A	0.29	0.51	0.89	0.99
	0.06	N/A	0.17	0.34	0.72	0.98
P300	0.04	N/A	N/A	0.72	0.96	1.00
	0.05	N/A	N/A	0.53	0.88	1.00
	0.06	N/A	N/A	0.35	0.76	0.99
P400	0.04	N/A	N/A	0.89	0.97	1.00
	0.05	N/A	N/A	0.66	0.96	0.99
	0.06	N/A	N/A	0.41	0.88	0.98
P500	0.04	N/A	N/A	N/A	0.92	0.99
	0.05	N/A	N/A	N/A	0.81	0.99
	0.06	N/A	N/A	N/A	0.57	0.98

R215.txt in Stu04Rpt) which lead to detection failure on the document pairs whose copy ratio is greater than 0.1 as well. All these 4 documents contain large segments of C-style source code in them. Ferret_C ignores any non-Chinese character so that it cannot detect the copied code in the plagiarised documents. The copied tokens that Ferret_C can see make up a small proportion so that Ferret_C gets a small R-score, which causes failure. However, a section of 500 code characters or English text contributes fewer tokens and tuples than Chinese characters do. Though the Ferret_T and Ferret_M take into account all characters, code tends to account for a smaller proportion than Chinese text in the whole set of tuples in a document. Since Ferret_M considers each Chinese character as a token the size of a document's tuple set is much larger than that of Ferret_T. If the copied section consists mainly of code, then Ferret_M gets a small R-score, which causes its failure. However, the smaller size of the tuple set does not produce such a low

Table XII. Distribution of plagiarized document pairs among different copy ratios in a corpus

Copy ratio	Stu04Rpt_ P50	P100	P200	P300	P400	P500
<0.05	752	586	342	263	185	180
[0.05, 0, 1)	119	315	435	449	411	382
[0.1, 0.2)	6	28	161	256	338	385
≥ 0.2	154	154	166	185	212	241
total	1031	1083	1104	1153	1146	1188

Table XIII. Recall of Ferret_T on different ratio of plagiarism around the optimum threshold

Corpus	Threshold	<0.05	[0.05, 0.1)	[0.1, 0.2)	≥ 0.2
Stu04Rpt_P50	0.01	0.28	0.55	0.83	1.00
	0.02	0.04	0.34	0.67	1.00
Stu04Rpt_P100	0.01	0.60	0.93	1.00	1.00
	0.02	0.24	0.85	0.93	1.00
Stu04Rpt_P200	0.01	0.80	0.99	1.00	1.00
	0.02	0.40	0.95	0.99	1.00
Stu04Rpt_P300	0.01	0.95	1.00	1.00	1.00
	0.02	0.52	0.98	1.00	1.00
Stu04Rpt_P400	0.01	1.00	1.00	1.00	1.00
	0.02	0.68	0.98	1.00	1.00
Stu04Rpt_P500	0.01	0.98	0.98	1.00	1.00
	0.02	0.88	0.98	1.00	1.00

R-score for Ferret_T so it detects the copied code, and seldom misses plagiarised documents in the corpora.

5. Discussion and Conclusions

In this article, we adapted the Ferret copy detection system to handle Chinese corpora, comparing three definitions of document type: typeText, typeChinese, and typeMix. The three document types use

Table XIV. Recall of Ferret_C on different ratio of plagiarism around the optimum threshold

Corpus	Threshold	<0.05	[0.05, 0.1)	[0.1, 0.2)	≥ 0.2
Stu04Rpt_P50	0.04	0.05	0.50	0.50	1.00
	0.05	0.01	0.34	0.50	1.00
	0.06	0.00	0.14	0.50	1.00
Stu04Rpt_P100	0.04	0.18	0.79	0.86	1.00
	0.05	0.04	0.65	0.79	1.00
	0.06	0.00	0.40	0.79	1.00
Stu04Rpt_P200	0.04	0.40	0.91	0.94	0.99
	0.05	0.17	0.79	0.93	0.99
	0.06	0.03	0.62	0.89	0.99
Stu04Rpt_P300	0.04	0.68	0.94	0.98	0.99
	0.05	0.38	0.88	0.97	0.98
	0.06	0.17	0.77	0.96	0.98
Stu04Rpt_P400	0.04	0.85	0.95	0.99	1.00
	0.05	0.53	0.90	0.98	1.00
	0.06	0.17	0.76	0.97	0.99
Stu04Rpt_P500	0.04	0.89	0.96	0.99	1.00
	0.05	0.71	0.93	0.98	0.99
	0.06	0.26	0.86	0.97	0.98

different strategies to represent the characters and other possible components of a Chinese sentence. Our purpose was to see to what extent plagiarised or copied sections could be detected.

The experiments described above show that typeText has different detection results from typeChinese and typeMix, but typeChinese has similar results to typeMix. This is the result of applying different representational strategies, in which very different tokens are extracted from the same Chinese sentence, as illustrated in Figure 1. The naive strategy makes a long token, and may even take a whole Chinese sentence as a single token.

According to our results, it seems that typeText performs better than typeChinese and typeMix because the F1 value of typeText is greater than that of the other two in most experiments. The long token is helpful in increasing precision because a long identical string is more powerful evidence of copying than a short one.

Table XV. Recall of Ferret_M on different ratio of plagiarism around the optimum threshold

Corpus	Threshold	<0.05	[0.05, 0.1)	[0.1, 0.2)	≥ 0.2
Stu04Rpt_P50	0.04	0.04	0.50	0.50	1.00
	0.05	0.01	0.31	0.50	1.00
	0.06	0.00	0.13	0.50	1.00
Stu04Rpt_P100	0.04	0.17	0.77	0.86	1.00
	0.05	0.02	0.64	0.79	1.00
	0.06	0.00	0.36	0.71	1.00
Stu04Rpt_P200	0.04	0.37	0.91	0.96	0.99
	0.05	0.15	0.78	0.94	0.99
	0.06	0.03	0.59	0.91	0.99
Stu04Rpt_P300	0.04	0.64	0.94	0.98	0.99
	0.05	0.36	0.88	0.97	0.99
	0.06	0.14	0.75	0.97	0.98
Stu04Rpt_P400	0.04	0.83	0.94	1.00	1.00
	0.05	0.43	0.89	0.99	1.00
	0.06	0.14	0.74	0.98	1.00
Stu04Rpt_P500	0.04	0.88	0.96	0.99	1.00
	0.05	0.68	0.93	0.99	1.00
	0.06	0.22	0.86	0.98	0.99

In some situations typeText will be the most appropriate approach, for instance in comparing different versions of regularly revised reports, where there is no intention to deceive.

However, a different approach is needed when an intention to deceive is anticipated. In many plagiarised documents in the real world there are minor alterations and rewordings in an attempt to avoid detection. The pseudo-plagiarised documents described in this article consist of directly copied paragraphs without any rewording, so it is very easy to find long identical sentences in them. But they do not typically reflect the nuances of real world plagiarism. For example, if a copied sentence is converted from active voice to passive voice, tuples composed of the longer tokens will differ, whereas the majority of the tuples composed of single character tokens will still match. This is also the case when texts are altered by the insertion, deletion or substitution of a single word or short phrase.

The basic detection unit of Ferret is a tuple made of three tokens. In English a token is a word, and a tuple is only a small part of a sentence. The longer a tuple is, the frailer the detection mechanism becomes. In a Chinese document, typeText tends to assemble Chinese sentence(s) into a tuple, which may make it susceptible to this problem.

When typeText detects copying, we can be confident it exists: it is a sufficient condition. However, it is not a necessary condition: there may be copied text that it will miss which the finer-grained, single character strategy can find. The naive strategy of typeText is fit for detecting the simplest straightforward copying, but may be less effective at finding real world plagiarism.

In situations where there is a deliberate attempt to deceive, typeChinese and typeMix will be more robust than typeText. The single character strategy works for Ferret, and it is good enough to detect copied material up to the limits discussed above. Based on the experiments, Ferret works well on Chinese, and we can draw some conclusions.

1. The single character strategy works well on Chinese documents for detecting real plagiarism. A typical optimum threshold of Ferret is round 0.04 to 0.05 for this data, when Chinese documents are treated as typeChinese or typeMix.

2. Where there is no attempt to deceive, or with pseudo-plagiarised documents, typeText is an effective strategy. A typical optimum threshold is round 0.01 to 0.02

3. The optimum threshold for any particular corpus can be found by analysing a small sample of document pairs.

4. A higher threshold can increase precision but lose some potential plagiarised documents. The level of recall depends on the amount of copied material, and small amounts may not be detected. Put another way, the typical lower limit of Ferret's detection ability is about 0.05 copy ratio. If the copied content is greater than this, then Ferret will have a high probability of finding it.

By taking Chinese characters as tokens we depart from any semantic representation. A character will often be a part of a word and a trigram of characters may be devoid of meaning. It is in this sense that we use a sub-symbolic representation, and observe the contrast between machine based engineering approaches and human based cognitive processing.

In this article, we have demonstrated that the principle of extracting tokens from text, which is so successful when applied to English texts, also applies to Chinese texts. Although we have conducted this research

in relation to the Ferret copy-detection system, many of the issues relate to the representation of written Chinese and have a wider applicability in the analysis of the Chinese language.

Acknowledgements

Dr. JunPeng Bao carried out this work at the University of Hertfordshire, UK, sponsored by the Royal Society as a Visiting International Fellow.

References

Bao, J. P., J. Y. Shen, X. D. Liu, and H. Y. Liu: 2006, 'A fast document copy detection model'. *Soft Computing* **10**, 41–46.

Bao, J. P., J. Y. Shen, X. D. Liu, H. Y. Liu, and X. D. Zhang: 2004a, 'Finding plagiarism based on common semantic sequence model'. In: *Proceedings of the 5th International Conference on Advances in Web-Age Information Management*, pp. 640–645.

Bao, J. P., J. Y. Shen, X. D. Liu, H. Y. Liu, and X. D. Zhang: 2004b, 'Semantic sequence kin: A method of document copy detection'. In: *Proceedings of the Advances in Knowledge Discovery and Data Mining*, pp. 529–538.

Broder, A. Z.: 1998, 'On the resemblance and containment of documents'. In: *Proceedings of Compression and Complexity of Sequences*, pp. 21–29.

Gao, J., M. Li, A. Wu, and C. N. Hang: 2006, 'Chinese word segmentation and named entity recognition: A pragmatic approach'. *Computational Linguistics* **31**, 531–573.

Lane, P. C. R., C. Lyon, and J. A. Malcolm: 2006, 'Demonstration of the Ferret plagiarism dectector'. In: *Proceedings of the 2nd International Plagiarism Conference*.

Lyon, C., R. Barrett, and J. A. Malcolm: 2003, 'Experiments in plagiarism detection'. Technical report 388: School of Computer Science, University of Hertfordshire.

Lyon, C., R. Barrett, and J. A. Malcolm: 2004, 'A theoretical basis to the automated detection of copying between texts, and its practical implementation in the Ferret plagiarism and collusion detector'. In: *JISC (UK) Conference on Plagiarism: Prevention, Practice and Policies Conference*.

Lyon, C., J. A. Malcolm, and R. G. Dickerson: 2001, 'Detecting short passages of similar text in large document collections'. In: *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Lyon, C., R. Barrett, and J. A. Malcolm: 2006, 'Plagiarism is easy, but also easy to detect'. *Plagiary* **1**, 1–10

Malpohl, G.: 2006 *JPlag: Detecting Software Plagiarism* `http://wwwipd.ira.uka.de:2222/`

Manning, C. D. and H. Schütze: 2001, *Foundations of statistical natural language processing*. Cambridge, MA: The MIT Press.

Turnitin: 2006 *Plagiairism Prevention* `http://www.turnitin.com/static/plagiarism.html`