SCAM: A Copy Detection Mechanism for Digital Documents*

Narayanan Shivakumar, Hector Garcia-Molina
Department of Computer Science
Stanford University
Stanford, CA 94305-2140
{*shiva, hector*}@*cs.stanford.edu*

ABSTRACT

Copy detection in Digital Libraries may provide the necessary guarantees for publishers and newsfeed services to offer valuable on-line data. We consider the case for a registration server that maintains registered documents against which new documents can be checked for overlap. In this paper we present a new scheme for detecting copies based on comparing the word frequency occurrences of the new document against those of registered documents. We also report on an experimental comparison between our proposed scheme and COPS [6], a detection scheme based on sentence overlap. The tests involve over a million comparisons of netnews articles and show that in general the new scheme performs better in detecting documents that have partial overlap.

Keywords: Copy Detection, Plagiarism, Registration Server, Databases.

INTRODUCTION

A Digital Library provides users with on-line access to digitized news articles, books, and other information. In this environment, a user may easily redistribute this digital information on bulletin boards and mailing lists. Unless this problem is "solved," few publishers or authors will place valuable information in these Digital Libraries.

Most existing techniques that address this problem fall into two categories, those of copy prevention and copy detection. Copy prevention schemes include physical isolation of the information (e.g., by placing it on a stand-alone CD-ROM system), use of special-purpose hardware for authorization [18], and *active* documents that are essentially documents encapsulated by programs [10]. We believe that prevention techniques may be cumbersome, may get in the way of the honest user [6], and may make it difficult to share information. Furthermore, prevention schemes are not always bulletproof since documents may be *recorded* by using software emulators [6].

The other approach is not to place restrictions on the distribution of documents, but to *detect* illegal copies. Detection schemes fall into two categories, signature based and registration based. In signature based schemes, a "signature" is added to the document, and this signature can be used to trace the origins of the document. For example, one popular approach is to incorporate *watermarks* such as word spacings and checksum into documents [5, 4, 22, 7, 3].

Signature schemes have two weaknesses: (a) the signatures often can be removed automatically, leading to untraceable documents, and (b) they are not useful for detecting partial overlap. For these reasons we advocate registration based copy detection schemes. With these schemes original documents are registered and stored in a repository [17, 2]. Subsequent documents that are produced are compared against the pre-registered documents for partial or complete overlap. This check can be initiated by a person, e.g., a program committee member checking if a conference submission overlaps significantly with previous papers, or automatically by a program, e.g., a bulletin-boards or electronic mail gateway checking messages going through to see if they include copies of copyrighted articles. The repository of registered documents can be compacted in a variety of ways [6] and periodically distributed to mail gateways and bulletin boards so that checks can be done locally. Another application of registration copy detection is for filtering duplicate messages often found in newsgroups and mailing lists [25].

There are a number of ways to detect duplication with registered documents. In COPS [6], registered documents are broken up into sentences or sequences of sentences, and are stored in the registration server. Subsequent query documents are broken up in the same way and are compared against the registered documents. If a query document shares more than a given threshold of matching sentences (or sequences of sentences) with a registered document, the user is notified. Another scheme is presented in [14], where the problem of finding "similar" files is addressed. The mechanism works by selecting a few words as *anchors* and computing checksums of a following window of characters for comparison. It is mainly intended for file management applications and the detection of files that are very similar, but not for detection of small text overlaps.

Registration schemes can also be broken. For example, with COPS, a user can modify a large number of sentences, e.g., by adding or changing a word, rendering the new document untraceable to the original. However, this requires substantial manual work, and for this reason we believe registration based copy detection is superior to signature based schemes.

*This material is based upon work supported by the National Science Foundation under Cooperative Agreement IRI-9411306. Funding for this cooperative agreement is also provided by ARPA, NASA, and the industrial partners of the Stanford Digital Libraries Project. Any opinions, finding, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the other sponsors. This work was supported by an equipment grant from Digital Equipment Corporation.

Although COPS has been shown to work well [6], it does have some problems. In particular, it has some difficulties in detecting sentences. Often equations, figures, and abbreviations confuse it. Also, checking for overlap involves many random probes into the registration database, and is expensive. For these reasons, we have explored alternative schemes.

In this paper we present a comparison scheme based on the word occurrence frequencies of documents. Conceptually, we compute a vector that gives the frequency with which each possible word occurs in the new document. Then we compare this vector against "similar" vectors in the database of registered documents. This is very similar to how Information Retrieval (IR) systems compute document similarities [20], except that we use a new similarity measure that more accurately characterizes copy overlap, while traditional IR systems look for semantic similarity. Several schemes have been proposed to enhance IR schemes, such as use of signature files [8], lexical analysis [1], stoplists [13, 9], stemming algorithms [12, 15], thesaurus [21] and ranking algorithms [19]. Since our approach is based on IR, such schemes are orthogonal to our model, and one or more of these schemes could be used to enhance our document comparison mechanism.

Our scheme is based on words, which are easier to detect than sentences, and hence may be more accurate, especially for informal documents. We also believe that word access patterns have more locality than sentence access patterns and this may lead to improved performance in some cases. However, our main motivation in choosing words is that sentence based mechanisms such as COPS, cannot detect partial sentence overlaps. Hence we believe that word based schemes may be superior to sentence based mechanisms in detecting plagiarism in documents. To support our claims, we present results comparing COPS against our prototype SCAM (Stanford Copy Analysis Mechanism) on 1233 $\times$ 1233 netnews article pairs, and show that in general SCAM performs better than COPS in detecting instances of plagiarism. However, we also note that SCAM reports more *false positives* than COPS, where false positives are pairs of documents that are reported to be possible instances of plagiarism, but are not. We also compare SCAM against a traditional vector-based IR scheme on the same 1233 netnews articles, and show that SCAM again performs better in detecting document overlaps.

COPY DETECTION PRELIMINARIES

In this section, we present the architecture of a generic copy detection server and introduce relevant terminology. We also give a brief summary of some issues that need to be considered while building a copy detection server such as data structures, and the textual units used for comparison.

Copy Detection Server Architecture

In Figure 1, we see the architecture of a generic copy detection server with a repository of registered documents. (The repository is shown to be centralized, but in practice may be distributed.) We define *chunking* of a document to be the process of breaking up a document into more primitive units such as sentences, words or overlapping sentences. Documents that are to be registered are chunked and inserted into the repository. New documents that arrive are chunked into the same units and are compared against the pre-registered documents for overlap. In subsequent sections we will consider different chunking units and document similarity measures.

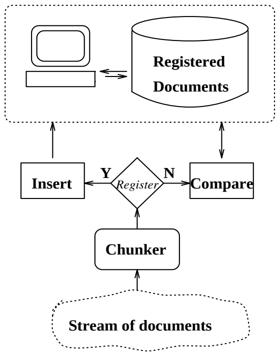

Figure 1: A Generic Copy Detection Server

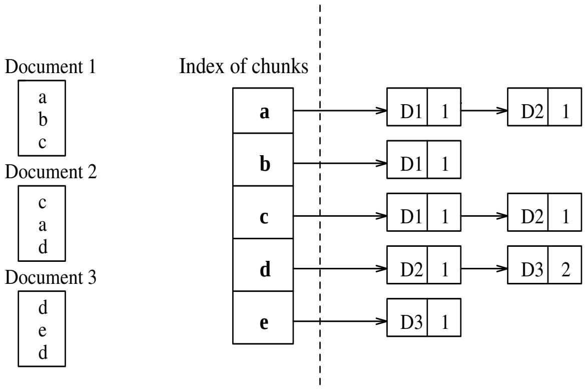

Figure 2: Inverted index storage mechanism.

Let W represent the *vocabulary* of the chunks, that is the set of occurrences of all distinct chunks in the registered documents. Let w_i refer to the i^{th} chunk in the vocabulary. Let the size of the vocabulary (number of distinct chunks) be N.

Inverted Index Storage

We propose using an inverted index structure (as in traditional IR systems [20]) for storing chunks of the registered documents. An index of the chunks in the vocabulary is constructed and maintained at registration time. Each entry for a chunk points to a set of *postings* that indicate the documents where the chunk occurs. Every posting for a given chunk w_i has two attributes (*docnum*, *frequency*), where *docnum* is a unique identifier of a registered document, and *frequency* is the number of occurrences of w_i in document with ID *docnum*. In Figure 2, we illustrate the index structure with three registered documents. The letters "a" through "e", which represent the chunks in the documents, constitute the vocabulary with $N = 5$. For instance, chunk "d" has two postings representing that it occurs once in document $D2$ and twice in document $D3$.

When a document D is to be compared against the pre-registered documents, the chunks of D are looked up in the registered document index. This means that only the documents that overlap at the chunk level will be considered using this index mechanism. Hence the total number of lookups on the index is the number of distinct chunks that occur in the document D.

Units of Chunking

As defined earlier, chunking involves breaking up a document into more primitive units such as paragraphs, sentences, words or overlapping sentences. The unit of chunking chosen for copy detection is critical since it shapes the subsequent overlap search cost and storage cost as outlined below.

- **Similarity level:** The bigger the chunking unit the lower the probability of matching unrelated documents. For instance, two unrelated documents may both have a sentence like "This research was funded by NSF" as part of a paragraph. If the chunking unit is a paragraph, the two documents will probably not be detected as an overlap, while they will be detected if the chunking unit is a sentence. On the other hand, the bigger the chunking unit, the higher the probability of missing actual overlaps. For instance, consider two paragraphs that share 5 out of 6 identical sentences. With paragraph chunking, no match will be detected, while with sentence chunking, 5 out of the possible 6 units will be detected as matching.
- **Search cost:** The larger the chunk, the higher the potential number of distinct chunks that will be stored. For instance, as the collection of documents grows, we expect the number of distinct sentences that will be stored to be higher than the number of distinct words. This is because beyond a certain point the number of new words introduced into the vocabulary will be low as opposed to the near-linear growth of sentences/ paragraphs. Hence we see that the potential size of the chunk index is higher when the chunking unit chosen is larger. Of course the number of postings per chunk is larger when the chunking unit is small (as in words).

However, we see one advantage for small chunking units. A small chunking unit increases *locality*. That is most documents will have a relatively small *working set* of words rather than sentences. Consider the frequency distribution of N words to follow Zipf's Law [26, 23, 11]. If the words are *ranked* in non-increasing order of frequencies, then the probability that a word w of rank r occurs is

$$P(w) = \frac{1}{r * \sum_{v=1}^{N} 1/v}$$

If we assume a vocabulary of about 1.8 million words [23], about 40,000 (about 2% of 1.8 million) words constitute nearly 75% of the actual occurrences of words thereby increasing the effects of cacheing. That is, by retaining the most popular words and their associated postings in main memory, we may be able to reduce the number of accesses to the disk-resident portion of the index. With sentence chunking, we expect the access pattern to be more random due to the very large size of the sentence vocabulary.

In this paper, we will investigate the use of words as the unit of chunking, as opposed to sentence chunking used by COPS. As we have argued, word chunking may lead to more locality during comparisons. In addition, word chunking has the potential to detect finer (e.g., partial sentence) overlap, which may be especially important with informal documents that may not have a clear sentence structure. As discussed earlier, COPS sometimes has problems detecting sentence boundaries.

However, before we can use word chunking, we need to determine a good scheme for comparing documents. Recall that for sentence chunking, comparison was straightforward: if X of the Y sentences in document D_1 appear in D_2 then the overlap is $X * 100/Y$ [6]. Unfortunately, this simple scheme breaks down for words: the fact that D_2 has many of the words of D_1 does not necessarily mean they overlap. In the next section, we propose one scheme based on relative frequency of words that we have empirically found to be effective.

OVERLAP MEASURES

In traditional IR schemes [20], when queries arrive from users, the query is "compared" against the documents, and some measure of relevance between the document and the query is obtained. Similarly, we need to establish a metric that measures the overlap between an incoming document and a pre-registered document. In this section, we consider a popular model used in IR systems termed the Vector Space Model (VSM) [20] that relates documents to queries, and see why it is not directly applicable for copy detection. We then propose the Relative Frequency Model (RFM) that presents a better framework for detecting overlaps.

For our discussion, let D refer to a generic document (registered or new). We define the *occurrence vector* $O(D)$ to be a listing of the chunks in D. Let $F(D)$ (size N) be the *frequency vector*, where $F_i(D)$ is the number of occurrences of chunk w_i in D. Let $sim(D_1, D_2)$ denote the similarity measure between documents D_1 and D_2, as computed below.

To illustrate the similarity computations, consider a registered document R and a new document S_j that is to be compared to R. Let $O(R) = < a, b, c >$, $O(S_1) = < a, b >$, $O(S_2) = < a, b, c >$, and $O(S_3) = < a^k >$, where $k \geq 1$ denotes the number of occurrences of chunk a in document S_3 (that is $F_a(S_3) = k$). Also let $O(S_4) = < a, b, c, d, e, f, g, h >$ Assume that the vocabulary is $W = \{a, b, c, d, e, f, g, h\}$. We would expect a good copy detection scheme to report R and S_1 to be "quite" similar, R and S_2 to be exact replicas, R and S_3 to be somewhat similar at low k values but not very similar for high k, and S_4 to have significant overlap with R.

Vector Space Model

A popular model in the IR domain [20], is the VSM model. Given a query with its corresponding weights, a dot product of the weighted occurrence vector of the query with a stored document is computed: if the dot product value exceeds a certain threshold, the document is flagged to match the query. A common weighting scheme used is the normalized frequency count. If we were to apply this measure to our copy detection example, the normalized frequency vectors would be $V(R) =< 1/3, 1/3, 1/3, 0, \ldots >$, and $V(S_1) =< 1/2, 1/2, 0, 0, \ldots >$ and so on. The similarity between R and S_1 would be $sim(R, S_1) = 1/3 * 1/2 + 1/3 * 1/2 = 1/3$. Similarly $sim(R, S_2) = 1/3$, $sim(R, S_3) = 1/3$ and $sim(R, S_4) = 1/8$. Since the overlap of R with S_2 and S_4 is more significant than that with S_1 or S_3, the overlap values reported are not acceptable.

Another popular weighting measure used is the *cosine similarity* measure [20] which defines relevance of document R to query Q to be

$$sim(R,Q) = \frac{\sum_{i=1}^{N} \alpha_i^2 * F_i(R) * F_i(Q)}{\sqrt{\sum_{i=1}^{N} \alpha_i^2 * F_i^2(R) * \sum_{i=1}^{N} \alpha_i^2 F_i^2(Q)}}$$

where α_i is the weight associated with the occurence of the i^{th} chunk. Intuitively, the higher the frequency of a word, the less the word contributes towards matching similarities. If we use this measure for our example and assume uniform weights for words ($\alpha = 1$), we find that $sim(R, S_1) = (1*1+1*1)/\sqrt{3*2} = 0.82$ and $sim(R, S_2) = 1$. (In our computation we used un-normalized frequency vectors, but the result is the same if they are normalized.) In this case, the metric appears to work well. Unfortunately, $sim(R, S_3) = k/(k*\sqrt{3}) = 0.58$. This means that the cosine measure is independent of the actual number of occurrences of a in document S_3 which is again unacceptable for copy detection. We need a metric that gives us a decreasing similarity measure as k increases. Also $sim(R, S_4) = 0.61$ is a fairly low value considering that the entire registered document is plagiarized. Therefore, we also need a metric that should detect *subset* overlaps that are instances of plagiarism.

Relative Frequency Model

From the earlier examples, we see that the cosine similarity measure works well when word frequencies are of similar magnitudes, but it does not when the magnitudes differ significantly. To use the good properties of the cosine similarity measure and remove its insensitivity to word frequency magnitudes, we define a measure that uses relative frequencies of words as indicators of similar word usage and combines it with the *cosine similarity* measure.

We first define the *closeness set* $c(R, S)$ to contain those words w_i that have similar number of occurrences in the two documents. That is, a word w_i is in $c(R, S)$ if it satisfies the following condition

$$\epsilon - \left(\frac{F_i(R)}{F_i(S)} + \frac{F_i(S)}{F_i(R)}\right) > 0$$

where $\epsilon = (2^+, \infty)$ is a user-tunable parameter. If either $F_i(R)$ or $F_i(S)$ are zero, we say that the closeness condition is not satisfied. Notice that if a word occurs the same number of times in two documents, then it occurs in the closeness set irrespective of the value of ϵ. If for example $F_i(R) = 3$ and $F_i(S) = 2$, then this measure computes $\epsilon - (3/2 + 2/3) = \epsilon - 13/6$. If $\epsilon > 13/6$ then w_i will be considered to be "close enough" to be in the closeness set. Intuitively, the closeness set determines the set of words used to the same extent in two documents, and forms the common "signature" for the document pair: ϵ is a tolerance factor while computing this set.

Next, we define the subset measure of document D_1 to be a subset of document D_2 to be

$$subset(D_1, D_2) = \frac{\sum_{w_i \in c(D_1, D_2)} \alpha_i^2 * F_i(D_1) * F_i(D_2)}{\sum_{i=1}^{N} \alpha_i^2 F_i^2(D_1)}$$

This expression computes the *assymetric* subset measure for the document pair, while only considering the close words. Note that this measure is very similar to the cosine similarity measure. The main motivation for this measure is that the values reported in the symmetric cosine measure are low even though a document may be a subset of another document (for instance $sim(R, S_4)$). The subset measure avoids this problem by normalizing the numerator of the regular cosine measure only with respect to the first document.

We then define the similarity measure between two documents to be

$$sim(R, S) = \max\{subset(R, S), subset(S, R)\}$$

If $sim(R, S) \geq 1$, we set $sim(R, S)$ to be 1. This is because no extra information is gathered when $sim(R, S)$ is computed to be greater than 1: the two documents are denoted to be very related anyway. We set the maximum value to be 1 merely to be able to express our similarity value as a range between 0 and 100%.

Intuitively, the components of the new similarity measure fit together as follows. The value of ϵ is the leeway in determining words that are to be used in computing the subset measure of a document pair. The similarity between the two documents is determined to be higher of the pair-wise subset measure (with the maximum similarity value being 1), and hence will have high values for registered documents that are either a subset or a superset of an incoming document. As we mentioned earlier, this feature is desirable for copy detection schemes, but is missing in traditional IR metrics.

In the following examples, we set α to be 1 since our current implementation assumes uniform weighting of words. To illustrate, with an $\epsilon = 2^+$, we get $c(R, S_1) = \{a, b\}$ and $sim(R, S_1) = \max\{(1*1+1*1)/3, (1*1+1*1)/2\} = 1$. Also we get $sim(R, S_2) = 1$, $sim(R, S_3) = 1$ when $k = 1$, $sim(R, S_3) = 0$ when $k > 1$, and $sim(R, S_4) = 1$. Notice that the $sim(R, S_3)$ values decrease as k increases, as desired, something not done by the original cosine measure. Also $sim(R, S_4)$ is a high value indicating an overlap between R and S_4. With an $\epsilon = 3$, $sim(R, S_1)$, $sim(R, S_2)$ and $sim(R, S_4)$ are unchanged. However, $sim(R, S_3)$ is now 1 for $k = 1$, 2/3 for $k = 2$ and 0 for $k \geq 3$. In general, the similarity value drops as k increases, but the difference in the a count needs to be larger before the similarity drops.

In general, a high value of ϵ increases the tolerance level for flagging common words in partially overlapping documents. but increases the chances of matching unrelated documents (false positives). A low value of ϵ (2^+) will decrease false positives but will also decrease the ability to detect minor overlaps. One important question that arises now is how one should choose a "good" value of ϵ that avoids missing partial overlaps while still not reporting several unrelated documents as similar. In the next section we address this question empirically by studying a collection of netnews articles.

EXPERIMENTS

In this section, we first define our comparison tests for documents, and then present some empirical results comparing word chunking (SCAM) with a sentence chunking scheme (COPS). To highlight the outlined disadvantages of the traditional IR measures, we subsequently present results comparing our document similarity model against the cosine similarity measure.

For our experiments we used 1233 netnews articles (with their headers removed) as the document set. We registered the 1233 articles and compared them for overlaps against themselves. There are 1,520,289 (1233*1233) document comparison values, one for each document pair. We also performed our tests on a set of 91 "conventional" text documents (draft versions, conference papers, journal papers) written by our research group. In this paper, we do not report these results for sakes of brevity. In general, we observed similar results as those we report for netnews articles. In our experiments, the value for ϵ in SCAM was set to 2.5 since we found it to work well in practice.

The first question that we consider, is how COPS and SCAM differ in documents they report to have overlap. In Figure 3, we show the overlap values reported by COPS and the corresponding distribution of values reported by SCAM. The table comparing SCAM and IR is similar to this table and is not presented. For the readers' convenience, we have added super-scripts to certain elements in our tables that we shall refer to. For instance, COPS reports 64^a entries to have overlap values between 81 and 90%. For those 64 documents, SCAM reports 39^b entries to be in the 91-99% range, and 9^c to have 100% overlap. From this table, it is clear that COPS and SCAM agree on more than 99.79% (percentage of diagonal elements) of document pairs. However, there are still significant differences in which documents they report to be overlaps. For instance, SCAM believes 2^d document pairs have 21-30% overlap, while COPS reports them to have 81-90% overlap.

We believe that SCAM and COPS were correct if both reported the same overlap value. We sampled a few document pairs manually in this category, and confirmed that this is a reasonable assumption. However, SCAM and COPS differ significantly in their reported overlap in about 0.21% of the document comparisons. Hence we investigate which system is "correct" in these cases. For this we manually examined the document pairs where the schemes differ.

The notion of correctness of a system such as SCAM or COPS depends on what our ultimate goal is. This goal can only be specified as a manual test where a human decides if a pair of documents actually involve plagiarism or substantial overlap, for instance. We call these possible manual tests Document Target Tests (DTTs). The goal of SCAM or COPS is to the "predict" the outcome of these DTTs. In this paper we consider four different DTTs

1. **Plagiarized:** If a document includes some parts of another article, we denote the article pair to have satisfied the Plagiarism test. (This does not imply malicious use of the copied text.)
2. **Subset:** If a document is almost completely included in another document (possibly interspersed with other text), we denote the document pair to have satisfied the Subset test.
3. **Copies:** If two documents appear to be exact copies, we denote the article pair to have satisfied the Copies test.
4. **Related:** If two documents appear to have a common thread relating them, we denoted the article pair to have satisfied the Related test.

The questions above are non-exclusive in that one or more of the tests could be satisfied for an article pair. In general, if one or more of the first three tests were satisfied, the fourth test was also satisfied. There were a few instances in which the fourth test was marked, even though none of the others were marked: these cases happened when an article was judged to be a response to the other, but had not included any part of the first article. In general, different humans may have different responses to the DTTs since they are subjective (Plagiarism and Related more so than others). Hence our results described below should be considered in an illustrative rather than in an "absolute" sense.

In Figures 4 and 5, we present results comparing SCAM and COPS on the differing document pairs (numbering 544). In these and the subsequent tables, we simplified our table structure for sakes of clarity. Document pairs that had similarity values less than 33% were classified under **None**, between 33 and 67% under **Some**, between 66 and 90% under **Lots**, and above 90% under **Full**. In Figure 4, we consider the results of document pairs that satisfied each DTT. The entry 249^e denotes that 249 document pairs were denoted by the human to be instances of plagiarism. The entry 26.91^h denotes that COPS reported 26.91% of the 249 documents to have substantial (lots) overlap. Similarly, 32.53^i denotes that SCAM reported 32.53% of the 249 documents to be almost identical copies. In Figure 5, we present the results of document pairs that did not satisfy the human DTT. The entry 295^j indicates that 295 document pairs were denoted by the human not be instances of plagiarism, and the entry $87.12\%^l$ indicates that SCAM reported 87.12% of the 295 documents to have some overlap.

As stated earlier, the ultimate goal of SCAM or COPS is to be able to "predict" the outcome of the above described human tests (DTTs). If a threshold τ is chosen for the value reported by each scheme, ideally for all the document pairs that satisfy the DTT, the scheme (SCAM or COPS) would report a value greater than τ. For document pairs that do not satisfy the DTT, the scheme would report a value less than τ. In other words, an ideal copy detection system would report overlap values as follows.

1. In the Detection comparison (Figure 4), 100% of the documents will have reported values greater than τ.
2. In the False Positive comparison (Figure 5), 100% of the documents will have reported values less than τ.

For instance, in Figure 4, assume that for the Copies DDT we chose the decision threshold to be between Lots and Full

	COPS	0	1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-	100
0	1,518,037	1,515,333	15	40	6	1518	618	297	121	45	25	2	17
1-10	68	0	3	2	0	19	9	18	4	10	1	2	0
11-20	48	0	0	0	12	2	3	8	6	1	2	2	12
21-30	57	0	0	0	1	20	9	4	2	7	3	5	6
31-40	38	0	0	1	9	1	3	2	6	4	4	3	5
41-50	32	0	0	0	4	3	5	1	3	2	6	2	6
51-60	33	0	0	0	3	4	6	3	2	2	2	1	10
61-70	15	0	0	0	4	0	1	1	0	2	1	0	6
71-80	9	0	0	0	1	1	0	0	0	0	6	0	1
81-90	**64**[a]	0	0	0	2	2	5	1	2	1	3	**39**[b]	**9**[c]
91-99	29	0	1	0	0	0	1	1	1	0	0	10	15
100	1859	0	0	1	0	4	0	0	0	0	0	7	1847
	Total	1,515,333	19	44	42	1574	660	336	147	74	53	73	1934

Figure 3: Distribution of SCAM's results with respect to COPS on netnews articles.

Test	System	# Satisfy(Test)	None	Some	Lots	Full
PLAGIARIZED	COPS	**249**[e]	**43.78**[f]%	26.91%	**26.91**[h]%	2.41%
	SCAM	249	2.01[g]%	38.55%	26.91%	**32.53**[i]%
SUBSET	COPS	120	17.50%	35.83%	45.83%	0.83%
	SCAM	120	0.83%	2.50%	30.00%	66.67%
COPIES	COPS	38	5.26%	2.63%	92.11%	0.00%
	SCAM	38	0.00%	0.00%	0.00%	100.00%
RELATED	COPS	253	44.66%	26.48%	26.48%	2.37%
	SCAM	253	1.98%	39.92%	26.48%	31.62%

Figure 4: Detection test comparison of SCAM and COPS on netnews articles.

Test	System	# ¬Satisfy(Test)	None	Some	Lots	Full
PLAGIARIZED	COPS	**295**[j]	**100.00**[k]%	0.00%	0.00%	0.00%
	SCAM	295	0.00%	**87.12**[l]%	10.17%	2.71%
SUBSET	COPS	424	90.33%	5.66%	2.83%	1.18%
	SCAM	424	0.94%	82.55%	14.39%	2.12%
COPIES	COPS	506	79.45%	13.04%	6.32%	1.19%
	SCAM	506	0.99%	69.76%	19.17%	10.08%
RELATED	COPS	291	100.00%	0.00%	0.00%	0.00%
	SCAM	291	0.00%	86.60%	10.31%	3.09%

Figure 5: False positive test comparison of SCAM and COPS on netnews articles.

Test	System	# Satisfy(Test)	None	Some	Lots	Full
PLAGIARIZED	IR	167	**7.19**[m]%	51.50%	41.32%	0.00%
	SCAM	167	**21.56**[n]%	7.78%	22.16%	48.50%
SUBSET	IR	86	0.00%	23.26%	**76.74**[o]%	**0.00**[q]%
	SCAM	86	0.00%	0.00%	**11.63**[p]%	**88.37**[r]%
COPIES	IR	14	0.00%	0.00%	100.00%	0.00%
	SCAM	14	0.00%	0.00%	0.00%	100.00%
RELATED	IR	170	7.06%	50.59%	42.35%	0.00%
	SCAM	170	21.76%	8.82%	21.76%	47.65%

Figure 6: Detection test comparison of SCAM and IR on netnews articles.

Test	System	# ¬Satisfy(Test)	None	Some	Lots	Full
PLAGIARIZED	IR	483	**19.25**[s]%	75.16%	5.59%	0.00%
	SCAM	483	**74.95**[t]%	19.67%	3.52%	1.86%
SUBSET	IR	564	18.62%	76.06%	5.32%	0.00%
	SCAM	564	70.57%	19.15%	7.80%	2.48%
COPIES	IR	636	16.51%	70.60%	12.89%	0.00%
	SCAM	636	62.58%	16.98%	8.49%	11.95%
RELATED	IR	480	19.38%	75.63%	5.00%	0.00%
	SCAM	480	75.21%	19.38%	3.54%	1.88%

Figure 7: False positive test comparison of SCAM and IR on netnews articles.

(i.e., at 90% overlap). In this case, SCAM would have a 100% success rate for the Detection test (for all cases where the documents are copies, the reported overlap is greater than the threshold). Similarly, SCAM has nearly 90% success rate for the False Positive tests (in 90% of the cases where the documents are not copies, the overlap is less than the threshold). For COPS, this divide is less clear for the Copies DTT: if we choose the Lots-Full boundary, COPS would have 0% success in the Detection test, and nearly 99% success in the False Positive test. If we choose the Some-Lots boundary, COPS would have 92.11% success in the Detection test, but only 92.5% success in the False Positive test. Similarly for the rest of the boundaries and DTTs.

Notice that in general, SCAM performs much better than COPS in detecting document overlaps. For instance, if we set the threshold for the Plagiarism test to be 33% (None-Some), SCAM would make an error only in 2.01[g]% of the documents by not reporting them to be instances of plagiarism, while COPS would not detect up to 43.78[f]% of the documents in the disagreement set. Similarly, with a threshold of 90% (Lots-Full) COPS would not detect exact copies of an article, while SCAM would have a 100% success rate on these copies.

However, this enhanced detection of document overlaps in SCAM has a price: higher percentage of false positives. For example, say we set the threshold for the Plagiarism test to 33% (None-Some). In this case, COPS would correctly report that none of the 295[j] document pairs were instances of Plagiarism (Figure 5). However, SCAM would incorrectly report that all the documents were instances of Plagiarism.

The reader should note that the overall percentage of false positives is much lower since Figures 4 and 5 only consider the "problem" document pairs. For example, we stated above that with a 33% threshold, SCAM would incorrectly diagnose 295 document pairs as Plagiarism. However, these are 295 pairs out of 1,520,289, so the false positive rate is closer to $295/1,520,289 = 0.02\%$. Hence, the fraction of extra (unnecessary) documents reported in a Plagiarism test would be very low. Of course, this fraction could be made even smaller by using a higher threshold, but this would reduce the number of actual Plagiarism cases detected by SCAM.

Incidentally, notice that by only showing four ranges (None, Some, Lots, Full) in our tables, we may be biasing our conclusions. For example, perhaps a threshold of 35% or 40% would give more desirable results, while in our table we have limited our threshold values to four. In spite of this, we only show four ranges because (1) showing more ranges would make it harder to visualize the results, and (2) we believe (after analyzing the raw data) that these four ranges are adequate to roughly distinguish the various cases.

In summary, we can see that COPS is very conservative in denoting document pairs to satisfy the DTT, and hence misses several real instances of overlap. On the other hand, SCAM is liberal in denoting document pairs to satisfy the DTT while they actually are not instances. A liberal scheme like SCAM may be more appropriate if catching most cases of overlap is paramount. In this case, it is useful to have a human check the detected documents to eliminate false positives. On the other hand, if overlap detection is to be completely automated, as for duplicate removal in netnews articles [24], COPS has the advantage.

In Figures 6 and 7, we report tables similar to Figure 4 and 5, comparing our new document comparison measure to the traditional IR cosine measure. If we were to choose the boundary between Lots and Full as our breaking point for the Subset DTT, SCAM would have nearly 88.37% success in the Detection test, and 97.5% success in the False positive test. On the other hand, it is less clear what should be the boundary for IR for the Subset DTT. The boundary Lots-Full would give a success rate of 0% for the Detection test, and 100% for the False Positive test. The boundary Some-Lots would give a success rate of 76.74% for the Detection test, and about 95% success for the False Positive test. Similarly for the other boundaries and DTTs.

In general, we see that SCAM performs better than the traditional IR measures in separating the document pairs by their thresholds. Also the number of false positives is lower in SCAM than in IR. For instance, with a 33% threshold, 74.95[t]% of the document pairs that are not instances of plagiarism can be filtered away in SCAM, as opposed to only 19.25[s]% in IR.

DISCUSSION

We see from Figure 4 that SCAM performs better than COPS in many of the examined cases for detecting overlap values. We found that in general COPS performs worse in detecting overlaps for the following reasons.

COPS has problems with small sentences and cannot handle in its current implementation documents that have multiple copies of the same sentence. Also COPS has the classic *sentence boundary problem* since it is hard to detect when a sentence ends [6]. While there are some fairly sophisticated mechanisms [16] for detecting sentence boundaries, SCAM has the clear advantage since it uses word chunking.

COPS sometimes gets confused due to ".signature" files at the bottom of news messages while SCAM does not (again due to word chunking). We did some experiments with the two systems after stripping the .signature files at the bottom of the files, but this proved to worsen the behavior of both schemes (COPS more so than SCAM). This is because users when responding to an news article included part of the earlier message along with the .signature. Since it is not easy to automatically detect and remove .signatures from the middle of articles, COPS and SCAM had *lesser* associations with the "root" article due to the absence of .signature.

There were several articles that had limited punctuation. Since COPS relies on punctuation marks for its sentence chunking, it had more problems. This was very common in several rec.sport.$*$ columns which included lists of "favorite" players/ teams, and in clari.finance.* groups which listed current gold prices and currency conversion rates with little punctuation.

One of the biggest problems that COPS had was in partial sentence overlaps (as expected). Since several users who respond to articles include only parts of sentences that they respond to (or more typically the line in which some comment occurred), COPS did not detect real overlaps of documents. This highlights one significant problem of sentence chunking.

We see from Figure 5 that SCAM reports more false positives than COPS. Very small documents that had similar vocabulary, had high values of overlap. We expect to overcome the number of false positives at the lower overlap values by introducing a *weighting* scheme for words depending on occurrence frequency in documents. It may also be possible to entirely drop low weight words, improving SCAM performance.

Another weakness of SCAM is that it is not clear how to choose a "good" value of ϵ for a wide variety of documents. We have empirically found a value (2.5) of ϵ that works well for netnews articles and a small set of conventional text documents, but it is not clear if this value will work equally well with other sets of documents. We plan to consider some more document sets, and check if the 2.5 value is universally acceptable for a large class of document sets.

One promising idea that we are exploring is to combine copy detection schemes like COPS and SCAM. A system that is provided with both schemes could use the most appropriate one for the desired overlap test. Another idea is to develop more accurate tests (even if they are more expensive) that could be applied on the small subset of documents identified by COPS or SCAM, in order to reduce the number of false positives.

CONCLUSION

We consider the copy detection problem from the perspective of a registration server that analyzes documents for overlap. We have presented a new similarity measure based solely on word frequencies in documents. We presented results of some initial comparisons of SCAM (Stanford Copy Analysis Mechanism), our current prototype, against COPS [6], another approach based on sentence comparisons. The results demonstrate the general advantages of word chunking for detecting document overlap, while also highlighting that using sentence chunking reduces the number of false positives.

Our comparisons have used relatively small netnews articles due to easy availability of several related document sets. We expect to explore the applicability of word chunking and our document comparison model on more larger conventional text documents in the future (in addition to the set of text documents like draft, conference papers that we have already experimented with.) In the long run, we believe that a hybrid scheme that uses both word and sentence chunking will be the most effective at detecting overlap, although, of course, its cost may be high. We are also planing a detailed performance evaluation of the various overlap schemes.

REFERENCES

1. A. Aho and M. Corasick. Efficient string matching: An aid to bibliographic search. *Communications of the ACM*, 18(6), 1975.
2. C. Anderson. Robocops: Stewart and Feder's mechanized misconduct search. *Nature*, 350(6318):454–455, April 1991.
3. D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. Technical Report 468, Computer Science Department, Princeton University, January 1995.
4. J. Brassil, S. Low, N. Maxemchuk, and L.O'Gorman. Document marking and identification using both line and word shifting. Technical report, AT&T Bell Labratories, 1994. May be obtained from ftp://ftp.research.att.com/dist/brassil/docmark2.ps.
5. J. Brassil, S. Low, N. Maxemchuk, and L.O'Gorman. Electronic marking and identification techniques to discourage document copying. Technical report, AT&T Bell Labratories, 1994.
6. S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *Proceedings of the ACM SIGMOD Annual Conference*, San Francisco, CA, May 1995.
7. A. Choudhury, N. Maxemchuk, S. Paul, and H. Schulzrinne. Copyright protection for electronic publishing over computer networks. Technical report, AT&T Bell Labratories, 1994. Submitted to IEEE Network Magazine June 1994.
8. C. Faloutsos. Description and performance analysis of signature file methods. *ACM Transactions on Office Information Systems*, 2(4), 1984.
9. W. Francis and H. Kucera. *Frequency analysis of English Usage*. Houghton Mifflin, New York, 1982.
10. G. N. Griswold. A method for protecting copyright on networks. In *Joint Harvard MIT Workshop on Technology Strategies for Protecting Intellectual Property in the Networked Multimedia Environment*, April 1993.

11. W. Li. Random texts exhibit zipf's law-like word frequency distribution. *IEEE Transactions on Information Theory*, 38(6):1842, 1992.

12. J.B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1-2), 1968.

13. H.P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4), 1957.

14. U. Manber. Finding similar files in a large file system. In *USENIX*, pages 1–10, San Francisco, CA, January 1994.

15. C. Paice. Another stemmer. *ACM SIGIR Forum*, 24(3), 1990.

16. D.D. Palmer and M.A. Hearst. Adaptive sentence boundary disambiguation. In *Proceedings of ANLP*, Stuttgart, Germany, October 1994.

17. A. Parker and J. O. Hamblen. Computer algorithms for plagiarism detection. *IEEE Trasnactions on Education*, 32(2):94–99, May 1989.

18. G.J. Popek and C.S. Kline. Encryption and secure computer networks. *ACM Computing Surveys*, 11(4):331–356, December 1979.

19. G. Salton and C. Buckley. Term-weighting approaches in authomatic text retrieval. *Information Processing and Management*, 24(5), 1988.

20. G. Salton. The state of retrieval system evaluation. *Information processing & management.*, 28(4):441, 1992.

21. Y-C. Wang, J. Vandendorpe, and M. Evens. Relationship thesauri in information retrieval. *J. American Society of Information Science*, 1985.

22. D. Wheeler. Computer networks are said to offer new opportunities for plagarists. *The Chronicle of Higher Education*, pages 17, 19, June 1993.

23. T. Yan and H. Garcia-Molina. Index structures for selective dissemination of information under the boolean model. *IEEE Transactions on Database Systems*, June 1994.

24. T. Yan and H. Garcia-Molina. Duplicate detection in information dissemination. In *Proceedings of Very Large Databases (VLDB'95) Conference*, Zurich, Switzerland, September 1995.

25. T. Yan and H. Garcia-Molina. Sift – a tool for wide-area information dissemination. In *Proceedings of USENIX*, 1995.

26. G.K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley Press, Cambridge, Massachusetts, 1949.