Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization

Aleksander Kołcz
AOL, Inc.,
44900 Prentice Drive
Dulles, VA 20166, USA
a.kolcz@ieee.org

Abdur Chowdhury
AOL, Inc.,
44900 Prentice Drive
Dulles, VA 20166, USA
cabdur@aol.com

Joshua Alspector
AOL, Inc.,
44900 Prentice Drive
Dulles, VA 20166, USA
jalspector@aol.com

ABSTRACT

Detection of near duplicate documents is an important problem in many data mining and information filtering applications. When faced with massive quantities of data, traditional duplicate detection techniques relying on direct inter-document similarity computation (e.g., using the cosine measure) are often not feasible given the time and memory performance constraints. On the other hand, fingerprint-based methods, such as I-Match, are very attractive computationally but may be brittle with respect to small changes to document content. We focus on approaches to near-replica detection that are based upon large-collection statistics and present a general technique of increasing their robustness via multiple lexicon randomization. In experiments with large web-page and spam-email datasets the proposed method is shown to consistently outperform traditional I-Match, with the relative improvement in duplicate-document recall reaching as high as 40-60%. The large gains in detection accuracy are offset by only small increases in computational requirements.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining

Keywords

deduplication, data cleaning, spam filtering, web mining

1. INTRODUCTION

In recent years, large dynamic document repositories have become commonplace, owing to the Internet phenomenon and to rapid advances in the storage technology. Due to a variety of factors (e.g., redundancy/mirroring, spam, plagiarism), such repositories may contain more than one copy of some documents, where sometimes the multiple copies are not exactly identical, but are similar enough to be considered as near duplicates. Near-duplicate proliferation is often undesirable, with potential problems including increased storage requirements, decrease in the quality of search engine performance and spam. Large concentrations of duplicate documents may also skew the content distribution statistics with potentially harmful consequences to machine learning applications [19].

A number of duplicate-detection schemes have been proposed in the literature (e.g., [3][2][6]). Their focus varies from providing high detection rates to minimizing the computational and storage resources needed by the detection process. With massive document repositories, run-time performance tends to be critical, which makes relatively simple single-hash techniques such as I-Match [6], particularly attractive. Unfortunately, document signatures produced by such techniques are potentially unstable in the presence of even small changes to document content. In applications such as spam filtering, where the adversary often purposely randomizes the content of individual messages to avoid detection [12], such instability is clearly undesirable.

In this work we propose an extension of the I-Match technique (but also applicable to other single-signature schemes) that significantly increases its robustness to small document changes. Our approach is based on randomization of the I-Match lexicon. The improvements in detection accuracy come at the cost of increased signature size, with an easily controllable trade-off. In a number of experiments, we show consistent superiority of our approach over the original scheme and demonstrate its attractiveness for the target applications of information retrieval and spam filtering.

The paper is organized as follows: Section 2 provides an overview of prior work in the area of near-duplicate detection. Section 3 focuses on the I-Match algorithm and suggests a modification improving its reliability for long documents. Section 4 introduces the randomized lexicon technique. In Section 5 we discuss applications of near-duplicate detection in web search in spam filtering. In Section 6 the experimental setup is outlined, with the results presented in Section 7. The paper is concluded in Section 8.

2. NEAR-DUPLICATE DETECTION: PRIOR AND RELATED WORK

The problem of finding duplicate, albeit non-identical, documents has been the subject of research in the text-retrieval and web-search communities, with application focus ranging from plagiarism detection in web publishing to redundancy reduction in web search and database storage.

KDD'04 August 22–25, 2004, Seattle, Washington, USA

Generally, one distinguishes between the problems of assessing the resemblance and containment of documents [3], although the techniques used to address them are often closely related [3][21]. A number of approaches to near-duplicate detection have been proposed, which can roughly be classified as [17]: *similarity-based* and *signature-based*.

Similarity-based techniques can be further sub-categorized according to the document representation chosen. On one end of the spectrum, the standard information-retrieval bag-of-words representation is used [5][20][17]. Alternatively, document decomposition into units larger than words leads to partial retention of positional information. In particular, shingle-oriented techniques such as COPS [2], KOALA [16], and DSC [4][3] view a document as a stream of tokens, which is broken into overlapping or non-overlapping segments referred to as shingles.

In principle, similarity-based techniques require that all pairs of documents are compared, i.e., each document is compared to every other document and similarity is calculated. For massive data sets, brute-force implementations can be computationally prohibitive, with the theoretical $O(d^2)$ runtime, where d is the number of documents. Several optimization techniques were proposed to reduce the number of comparisons made. In [8] only documents whose sizes are sufficiently close are compared against each other. In the fingerprinting context, frequently occurring shingles may be eliminated [16]. Also, instead of performing similarity computation over the complete sets of shingles, random sampling retains only a small subset of shingles per document [3]. Such simplifications, however, do hinder the accuracy.

Even with computational shortcuts, however, similarity based approaches to duplicate detection in large datasets tend to be computationally expensive. For example, with the performance-improving implementation of the DSC algorithm is still $O(s \cdot d)$, where s is the average number of shingles per document and d is the number of documents [4]. A more efficient alternative, DSC-SS, uses super shingles (i.e., concatenations of several shingles). Here, instead of measuring resemblance as a ratio of matching shingles, resemblance is defined as matching a single super shingle in two documents, which is much more efficient because it no longer requires the calculation of shingle overlap. A similar approach is used in [13], under the name of Locality Sensitive Hashing (LSH). Here, instead of shingles, a number of k-tuples of words are randomly selected from each document and two documents having at least one such tuple in common they are considered to be near duplicates.

3. I-MATCH AND ITS EXTENSIONS

Similarity-based duplicate detection approaches inherently map each document to one or more clusters of possible duplicates, depending on the choice of the similarity threshold. While that is appropriate when detecting the similarity of documents or detecting plagiarism, those techniques produce high overhead when large collections are evaluated. The I-Match [6] approach produces a single hash representation of a document, thus guaranteeing that a single document will map to one and only one cluster, while still providing fuzziness of non-exact matching. An I-Match signature is determined by the set of unique terms shared by a document and the I-Match lexicon. The signature generation process can be described as follows:

1. The collection statistics of a large document corpus are used to define an I-Match lexicon, L, to be used in signature generation.

2. For each document, d, the set of unique terms U contained in d is identified.

3. I-Match signature is defined as a hashed representation of the intersection $S = (L \cap U)$, where the signature is rejected if is $|S|$ below a user-defined threshold.

The effectiveness of I-Match relies on the appropriate choice of lexicon L. Experimental data suggest that one effective strategy of lexicon selection is to impose an upper and lower limit on the inverted document frequency (*idf*) for words in the document collection, since terms with mid-range *idf* values tend to be more useful in duplicate detection[6]. Note that high-*idf* terms may be very effective in pinpointing a particular document, but they also capture misspelled words and other spurious strings, which reduces their value in identifying *near* rather than *exact* duplicates.

I-Match may result in false-positive matches if a large document has a very small intersection with L. In other words, I-Match signature of a document may become unreliable when $\frac{|S|}{|U|}$ becomes too small. Here we propose an extension of the I-Match technique that addresses the small-intersection problem. Note that the collection statistics define an ordering over the set of all possible lexicon terms, with term *idf* used to determine the sorting order. Assuming that the primary lexicon, L, corresponds to a range of *idf* values, we reject all terms with lower *idf* values and define the secondary lexicon, B, as one containing the remaining terms ranked according to their increasing *idf* values.

In the modified I-Match procedure, whenever the primary lexicon fails to intersect with sufficiently many terms in U, the secondary lexicon is used to supply extra terms, until the number of elements in S exceeds a user-defined threshold.

4. DECREASING THE FRAGILITY OF I-MATCH SIGNATURES

Ideally, the signature of a document should be insensitive to small changes in document content. For example, in the context of the spam-filtering application, these include changing the order of words in the document, as well as inserting or removing a small number of words. Unlike signature-generation algorithms relying on positional information of words, I-Match is inherently insensitive to changes in the word order, but inserting or deleting a word from the active lexicon will change the value of the signature. Signature brittleness is particularly undesirable given the adversarial nature of spam filtering, where an attacker might attempt to guess the composition of the lexicon and purposely randomize messages with respect to the lexicon's vocabulary.

Let us reverse the roles of the document and the lexicon, however. We can reasonably expect that if a lexicon is modified by small number of additions/deletions, this is unlikely to significantly change the stability of I-Match signatures with respect to the modified lexicon. Moreover, similar levels of duplicate detection accuracy can often be obtained by largely non-overlapping lexicons [6]. A small modification to document content may thus change an I-Match signature due to a particular lexicon but, at the same time, there may

exist a number of alternative lexicons (for which I-Match performs with equivalent accuracy) for which the signatures may be unaffected by such a change.

The latter observation suggests the benefits of creating multiple signatures per document, which seems to require the presence of multiple different lexicons (selections of which could be non-trivial). We note, however, that such lexicons can be related to one another. In particular, let us consider a setup where a suitable lexicon is chosen and then K different copies of the original lexicon are derived by randomly eliminating a fraction p of terms in the original (i.e., the K extra lexicons are proper subsets of the original). Assuming that p is small, we expect the quality of signatures due to the additional lexicons to be similar to the original. Using the arguments presented above, an extended I-Match signature of a document in the randomized lexicon scheme is defined as a $(K+1)$ tuple, consisting of I-Match signatures due to the original lexicon and its K perturbations. Any two documents are considered to be near duplicates if their extended signatures overlap on at least one of the $K+1$ coordinates.

Let us take a document and modify it by randomly removing or adding a word from the original lexicon, with n such changes in total (note that changes involving vocabulary outside of the original lexicon cannot affect the extended I-Match signature). Each such change will necessarily change the signature according to the original lexicon, whereas the probability that at least one of the K additional signatures will be unaffected by such a change can be estimated as:

$$1 - (1 - p^n)^K \tag{1}$$

This is derived as follows: For a particular perturbation of the original lexicon, a change to the document contents will not affect the signature as long as the change occurs within the subset of the original lexicon that is missing in the perturbation, which occurs with the probability of p. Assuming that n is much smaller than the size of the missing subset, the probability that n such changes will preserve the signature can be approximated as p^n. Since the K additional lexicons were generated independently from one another, the process in which a number of the signatures is changed in response to modifications to the document can be modeled as K Bernoulli trials. Accordingly, the probability that all K signatures will change is equal to $(1-p^n)^K$ and, conversely, the probability that at least one of them will be unaffected is given by (1).

Eq. (1) can be seen as the stability of the extended I-Match signature to changes that are *guaranteed* to affect the I-Match signature according to the original lexicon alone. As illustrated in Figure 1, at the cost of using a few extra lexicons, the stability of I-Match signatures can be increased significantly.

Our approach is related to the supershingling/megashingling [3][10] and the locality sensitive hashing (LSH) [11][13]. As in those techniques, a number of signatures is generated per document and if a pair of documents has at least one of them in common, the documents are considered to be near-duplicates. The differences lie in how the individual signatures are generated. Most importantly, the proposed scheme is insensitive to word permutations and does not suffer from dependencies on document length.

5. APPLICATIONS

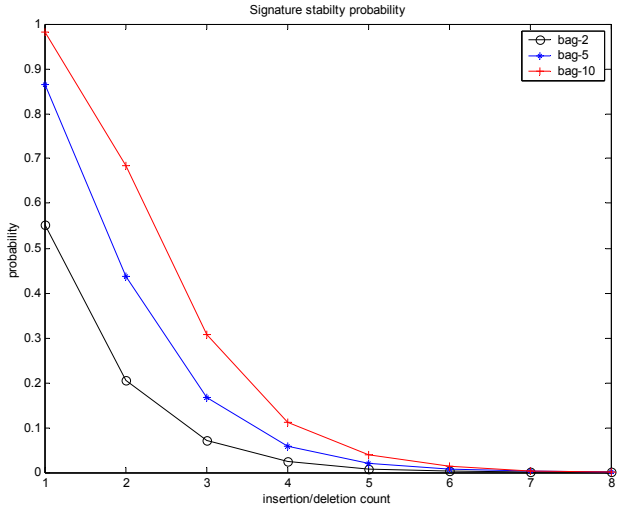

Figure 1: Stability of bagged I-Match signatures under random insertion/deletion of words in the original document for the case of $p = 0.33$. The y-axis corresponds to the probability that the extended I-Match signature will not be affected by a change to the document contents. Bag-n signifies that n randomized lexicons were used.

5.1 Web search

From the usability perspective, it is important that results of web search queries do not contain multiple references to the same information source. Unfortunately, due to mirroring and free copying, it is fairly common that the same web-page content may be available in multiple locations and under different names. Such documents often differ slightly, but the changes tend to be irrelevant. Another example is given by news stories, where often the same news article with minor modifications is posted multiple times and in different locations. Apart from contributing to information clutter, duplicates and near-duplicates increase the size of the document index, with negative consequences to indexing and retrieval performance [4][10]. Finding near-replicas of webpages has been one of the key motivations for researching scalable de-duplication techniques [4][13][6][18] and, given the exponential growth of the web, it continues to be an important application.

5.2 Spam filtering

Recently, there has been growing interest in applications of machine learning and data mining techniques to the problem of filtering spam [9]. Nevertheless, many practical systems (e.g., DCC[1] or Vipul's razor[2]) try to exploit one key characteristic of spam, i.e., its tendency to be sent in high volume. Duplicate detection systems often operate in a batch mode, where the objective is to find all duplicates in a large existing collection. In the cotext of spam filtering, given a collection of prototypes (of spam) and a stream of documents, one may wish to filter all documents in the

[1] http://www.rhyolite.com/anti-spam/dcc/
[2] http://razor.sourceforge.net/

stream that can be considered as near duplicates of some elements in the collection. The near rather than exact duplicate detection is critical since spam messages are rarely identical, precisely to avoid template-based detection schemes [12].

6. EXPERIMENTAL SETUP

In the following we evaluate the near-duplicate detection accuracy of the modified and extended I-Match using webpage and email document collections, where detection using single and multiple randomized signatures is compared.

6.1 Web page data

The WT10G [14][15] dataset from NIST was used for our web page similarity experiments. While this 10GB 1.7 million document collection is synthetic in nature, it was developed to possess characteristics of the larger web for text retrieval effectiveness research [1]. Additionally, the WT10G corpus has been examined for similarity of the collection to the web as a whole and was found to be representative [22]. These factors make it attractive for use in duplicate similarity experiments in which web pages are being examined.

6.2 Email data

- The Legitimate email collection consisted of $18,555$ messages collected from $4,607$ volunteers as examples of non-spam and was primarily to assess if near-duplicate detection of spam may lead to any false-positives among legitimate emails.
- The Honeypot-Spam collection consisted of $10,039$ messages collected by accounts set up to attract spam (i.e., they should not be receiving any email at all). These data were known to contain many highly similar messages.
- The Cluster-Spam collection consisted of $8,703$ spam messages grouped in 28 clusters. These data were obtained by interactively querying a large database of spam messages, where a cluster contained related messages extracted via queries employing different combinations of keywords.

6.3 Document preprocessing

After removing HTML markup, each document was mapped onto the set of unique words (defined as sequences of alphanumerics delimited by white space). In the case of email, only the text contained in the subject line and the message body was considered. Words were converted to lower case and the ones containing more than one digit as well as those having fewer than four characters were removed. Additionally, messages with fewer than 5 unique words were ignored. For the Honeypot and Cluster spam datasets, removal of trivial duplicates resulted in a reduction in the number of documents from $10,039$ to $5,328$ and from $8,703$ to $6,389$, respectively.

6.4 The evaluation process

The exact point at which two documents cease to be near-duplicates and become just highly similar is difficult to define. To avoid the ambiguity in the near-duplicate judgments, we chose the traditional cosine similarity measure as a benchmark metric against which the accuracy of the signature-based techniques was compared. For documents i and j, their cosine similarity is defined as

$$cosine(i,j) = \frac{|\text{common unique features}(i,j)|}{\sqrt{d(i)d(j)}}$$

where $d(j)$ is the number of unique features in document j. Our experience suggested that two documents can safely be considered as near-duplicates if their cosine similarity is greater than 0.9. In the presence of severe randomization, this does not guarantee that all duplicates of a particular template document will be recovered, but it is desired that a good duplicate-detection technique identifies a large fraction of the same documents as the cosine-similarity approach.

Given a query i, and a document collection, we define the recall of a signature-based detection technique as the ratio of the number of documents flagged as duplicates of i to the corresponding number identified by the cosine measure, when using i as a template,

$$recall\,(i) = \frac{|\text{duplicates found for } i|}{|\text{documents } j \text{ such that } cosine\,(i,j) \geq 0.9|} \quad (2)$$

6.5 I-Match signature algorithm settings

In applying I-Match and its extensions, one important question is the choice of the I-Match lexicon. In previous studies, the collection statistics of a large document set were used to find near-duplicates within that same collection, but it was also suggested that a large diverse *training* collection could be effective in detecting duplicates in a *different* collection. This is of particular importance, since the content distribution may be constantly changing and one often does not have a large enough target document collection to derive a stable lexicon. As recently shown in [7], there might be disadvantages to constantly updating the collection statistics to track the target distribution of content since it tends to reduce the time-validity of signatures while adding little in terms of deduplication accuracy. To evaluate the effect a lexicon choice might have, in the experiments with the web-page collection we considered two lexicons: one derived from the WT10G dataset itself and one derived from a large collection of news stories (here referred to as SGML), which was also used in [6]. In the case of the email data, just the SGML-based lexicon was applied. The SGML dataset corresponded to TREC disks 4-5[3], which is a compilation news collections.

6.6 WT10G lexicon

The WG10G collection contained 1,679,076 documents and 5.8 million unique terms, out of which $411,000$ terms were selected by retaining those words for which $nidf \in [0.2, 0.8]$. The choice of this particular interval was motivated by our past experience with I-Match applied to web data and the results reported in [6].

6.7 SGML lexicon

The SGML collection contained 556,000 documents , with an average length of 662 terms and a total numner of unique terms of 488,000. The *nidf* interval $[0.2, 0.8]$ contained the lexicon terms used in our experiments with the web data. For experiments with the email data, cross-validation suggested a different choice of the *nidf* range: $[0.2, 0.3]$.

[3]http://trec.nist.gov/data/qa/T8_QAdata/disks4_5.html

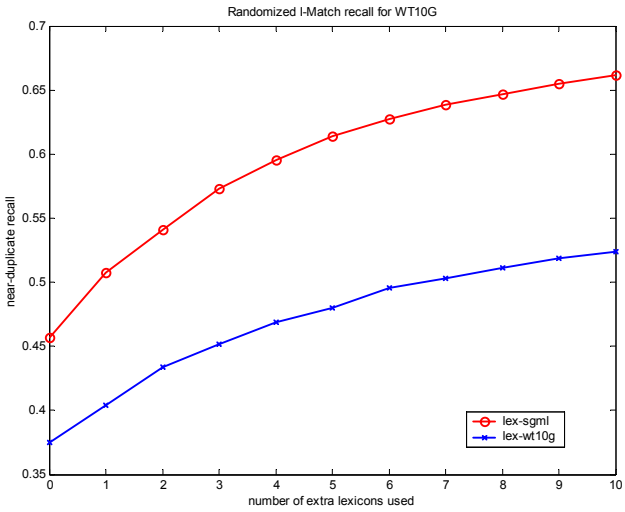

Figure 2: Near-duplicate web-page recall of I-Match as a function of the number of randomized lexicons used. Usage of multiple randomized lexicons is gernerally beneficial, but I-Match appears to perform best with lexicons derived from the SGML corpus.

6.8 Lexicon bagging

In experiments with lexicon randomization, the original copy of the lexicon was augmented with K copies (with K in $\{1, ..., 10\}$) obtained by bootstrap sampling from the original and ensuring that each term was selected at most once. Thus a randomized lexicon shared approximately 67% of terms with the original.

7. RESULTS

7.1 Web-page data

We used a random sample of 115,342 documents (approx. 7% of the total) as queries against the full WT10G collection. For each query, a set of documents for which the cosine similarity exceeded the 0.9 threshold was identified to be used for measuring the recall of hash-based techniques. Only 29,568 of the queries had a cosine neighborhood containing documents other than themselves and only those were used in the evaluation of I-Match. Note that computation of cosine similarity is quite expensive and would generally be unsuitable in operational settings.

I-Match with the WT10G and SGML lexicon choices was then applied to the document corpus and for the near-duplicate clusters containing the documents from the query set, the recall of the I-Match technique was measured. The results are shown in Figure 2. The use of lexicon randomization clearly provides a dramatic increase in the near-duplicate recall, although there is a saturation effect and beyond a certain point there is little benefit of introducing additional perturbed lexicons.

Interestingly, the SGML lexicon led to higher detection accuracy than the lexicon derived from the target collection. We suspect this is because the SGML collection was much "cleaner" and thus more representative of the content on which we usually wish to focus when deduplicating. Conversely, the WT10G collection contained both proper and misspelled versions of words, as well as various types of formatting noise, which would contribute to signature brittleness.

This seems to strengthen the arguments advanced in [7] that deriving a lexicon from large stable document collection may be preferred. Of course, in practice, this might depend on other factors, such as the language in which the target documents are written.

Table 1: Duplicate detection accuracy (reported as recall and relative increse in recall) in the *Honeypot-spam* and *Cluster-spam* experiments. *N*-bag signifies that *N* auxiliary lexicons were used.

	Honeypot		Cluster	
lexicon cnt	Recall	Rel. Increase	Recall	Rel. Increase
1	0.66	0%	0.40	0%
1+2-bag	0.72	9%	0.49	23%
1+5-bag	0.76	15%	0.55	38%
1+10-bag	0.80	21%	0.61	52%

7.2 Honeypot-spam and Cluster-spam vs. legitimate email

In these experimens, a random 10% of the honeypot/cluster-spam data was used as queries against the honeypot/cluster-spam and the legitimate-email datasets. The resulting average values of the recall are given in Table 1, which additionally shows relative increase in recall (see also Figure 3) due to signature randomization. None of the near-duplicate detection configurations produced any false-positive matches against the legitimate email collection. Lexicon randomization provided a clear benefit, both in terms of duplicate detection and spam detection metrics.

7.3 Discussion

Given that the SGML lexicon was derived using a large collection of news articles, it is interesting to observe its good generalization performance in the application considered, since web-page and email documents are generally different from news stories. This supports the claim that once a large diverse document collection is used, little in terms of copy-detection accuracy can be gained by tracking the changes to content distribution to fine tune the algorithm to the collection to which it is actually applied.

The results shown in Figures 2 and 3, and in Table 1 indicate that by using even a few extra randomized lexicons, the recall can be improved significantly. Given the rather small storage and computational consequences of using lexicon randomization (linear in the number of lexicons used), this should make the proposed method attractive in practical applications of I-Match.

8. CONCLUSIONS

We considered the problem of improving the stability of I-Match signatures with respect to small modifications to document content. The proposed solution involves the use of several I-Match signatures, rather than just one, all derived from randomized versions of the original lexicon. Despite utilizing multiple fingerprints, the proposed scheme does not involve direct computation of signature overlap,

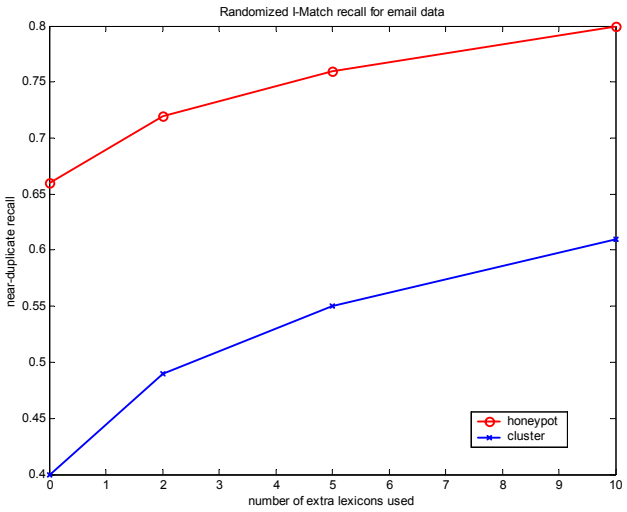

Figure 3: Near-duplicate spam recall of I-Match as a function of the number of randomized lexicons used. The absolute values of recall depend on the dataset (i.e., true amount of duplication present), but the benefits of using multiple lexicons are clear.

which makes signature comparison only marginally slower than in the case of single-valued fingerprints. Additionally, clear improvements in signature stability can be seen when adding just one extra signature component, with more gains to be made as more are added.

The original I-Match algorithm was modified to improve its reliability for very long documents. Also, we demonstrated that lexicons for I-Match can be successfully derived from a collection different from the target one, which in fact may be preferable if the target collection is noisy.

The proposed extended I-Match signature scheme does indeed provide greater robustness to term additions and deletions. Its effectiveness as a countermeasure to word substitutions is less, however, since a substitution is equivalent to an addition-deletion combination. In future work we intend to investigate ways to improve signature stability in the presence of term substitutions.

9. REFERENCES

[1] P. Bailey, N. Craswell, and D. Hawking. Engineering a multi-purpose test collection for web retrieval experiments. *Inf. Process. Management*, 39:853–871, 2003.

[2] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *Proceeding of SIGMOD*, pages 398–409, 1995.

[3] A. Broder. On the resemblance and containment of documents. *SEQS: Sequences '97*, 1998.

[4] A. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic clustering of the web. In *Proceedings of the Sixth International World Wide Web Conference*, 1997.

[5] C. Buckley, C. Cardie, S. Mardisa, M. Mitra, D. Pierce, K. Wagstaff, and J. Walz. The smart/empire tipster ir system. In *TIPSTER Phase III Proceedings*. Morgan Kaufmann, 2000.

[6] A. Chowdhury, O. Frieder, D. A. Grossman, and M. C. McCabe. Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems*, 20(2):171–191, 2002.

[7] J. Conrad, X. Guo, and C. Schriber. Online duplicate document detection: signature reliability in a dynamic retrieval environment. In *CIKM*, pages 443–452, 2003.

[8] J. Cooper, A. Coden, and E. Brown. A novel method for detecting similar documents. In *Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.

[9] T. Fawcett. "In vivo" spam filtering: A challenge problem for data mining. *KDD Explorations*, 5(2):203–231, 2003.

[10] D. Fetterly, M. Manasse, and M. Najork. On the evolution of clusters of near-duplicate web pages. In *Proceedings of the 1st Latin American Web Congress*, pages 37–45, 2003.

[11] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Databases (VLDB)*, 199.

[12] J. Graham-Cummings. The spammers' compendium. In *Proceedings of the Spam Conference*, 2003.

[13] T. Haveliwala, A. Gionis, and P. Indyk. Scalable techniques for clustering the web. In *Proceedings of WebDB 2000*, 2000.

[14] D. Hawking. Overview of the TREC-9 web track. In *TREC-9 NIST*, 2000.

[15] D. Hawking and N. Craswell. Overview of the trec-2001 web track. In *TREC-10 NIST*, 2001.

[16] N. Heintze. Scalable document fingerprinting. In *1996 USENIX Workshop on Electronic Commerce*, November 1996.

[17] T. C. Hoad and J. Zobel. Methods for identifying versioned and plagiarised documents. *Journal of the American Society for Information Science and Technology*, 2002.

[18] S. Ilyinsky, M. Kuzmin, A. Melkov, and I. Segalovich. An efficient method to detect duplicates of web documents with the use of inverted index. In *Proceedings of the Eleventh International World Wide Web Conference*, 2002.

[19] A. Kołcz, , A. Chowdhury, and J. Alspector. Data duplication: An imbalance problem ? In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets (II)*, 2003.

[20] M. Sanderson. Duplicate detection in the Reuters collection. Technical Report TR-1997-5, Department of Computing Science, University of Glasgow, 1997.

[21] Shivakumar and Garcia-Molina. Finding near-replicas of documents on the web. In *WEBDB: International Workshop on the World Wide Web and Databases, WebDB*. LNCS, 1999.

[22] I. Soboroff. Does wt10g look like the web? In *SIGIR 2002*, pages 423–424, 2002.