Hardening Fingerprinting by Context

Aleksander Kołcz
Microsoft Live Labs
Redmond, WA
USA

Abdur Chowdhury
Illinois Institute of Technology
Chicago, IL
USA

ABSTRACT

Near-duplicate detection is not only an important pre and post processing task in Information Retrieval but also an effective spam-detection technique. Among different approaches to near-replica detection methods based on document signatures are particularly attractive due to their scalability to massive document collections and their ability to handle high throughput rates. Their weakness lies in the potential brittleness of signatures to small changes in content, which makes them vulnerable to various types of noise. In the important spam-filtering application, this vulnerability can also be exploited by dedicated attackers aiming to maximally fragment signatures corresponding to the same email campaign. We focus on the I-Match algorithm and present a method of strengthening it by considering the usage context when deciding which portions of a document should affect signature generation. This substantially (almost 100-fold in some cases) increases the difficulty of dedicated attacks and provides effective protection against document noise in non-adversarial settings. Our analysis is supported by experiments using a real email collection.

1. INTRODUCTION

I-Match [2] and other signature based spam-filtering techniques are vulnerable to attacks aiming to increase signature fragmentation. Traditionally, signature based schemes have been attacked by inserting into the content so called "hash-busters", i.e., random and/or meaningless strings. Such random string attacks are not likely to be successful to thwart lexicon-based signature techniques, such as I-Match, however. This is because, to affect the signature, changes in document content have to intersect with the lexicon (which is kept secret). On the other hand, "good word" attacks that target primarily content-based spam filters have the potential of affecting a signature-dependent system as well. In fact, signature-based schemes can be more vulnerable to document modification attacks since inserted/removed/altered words do not have to be "good" in order to affect the signature. Also, changing just one word may be enough to alter the signature, while in order to change the decision of a content-based spam filter a more extensive content alteration is typically necessary.

In this work we present an extension to the I-Match algorithm [2] that validates the context within which lexicon terms are being used. The method is based on using a learning corpus to not only identify the lexicon terms, but also their proper usage context, which is then validated at the time of filtering (i.e., signature generation). We demonstrate that this simple "language model" technique can reduce signature fragmentation as much as 100 fold. While no signature based technique is immune to a dedicated attacker, we are able to show that circumventing the context-enhanced version of I-Match is much harder than in the case of the regular I-Match.

The paper is organized as follows. In Section 2 we review the signature-based near-duplicate detection and outline the I-Match algorithm. In Section 3 we analyze the reasons for fragility of I-Match signatures and propose a context enhanced extension of the algorithm. The difficulty of overcoming the original and context-sensitive variants of I-Match are examined in Section 4. Section 5 describes the evaluation framework and presents the experimental results. In Section 6 we overview related work and the paper is concluded in Section 7.

2. SIGNATURE-BASED NEAR DUPLICATE DETECTION

Near duplicate document detection is an important problem in several application domains. In Information Retrieval, and Web search in particular, it is essential for reducing the size of an inverted index (close to 30% of all Web pages are near replicas), as well as in post processing the results of a search query (e.g., detecting threads or multiple updates to the same story) [1][4]. Other applications include plagiarism detection [7], where one is interested not only in duplicate detection on the document level but also in identifying if significant portions of one document are "re-used" in another.

An important recent application domain is that of spam-detection [8][14][12]. One of the key features of spam can be described as *highly similar content in high volume*, since a spam campaign is often sent to many different recipients with only minor alterations to the message itself. Here near-duplicate detection can be used both to detect high-volume campaigns (some of which may be legitimate) and to filter messages belonging to already identified spam campaigns.

Approaches to near-duplicate detection can be roughly divided into those based on suitably-defined document overlap or similarity and those based on comparison of document signatures or fingerprints. The latter class carries a substantial scalability advantage since determination of a near-duplicate cluster membership can be established via a

CEAS 2007 - Fourth Conference on Email and Anti-Spam, August 2-3, 2007, Mountain View, California USA

single hash-table lookup (i.e., either the signature matches a prototype or it does not). On the other hand, they tend to be more fragile since sometimes insignificant changes to a document alter the signature and can break up a true underlying near-duplicate cluster into many smaller components. The types of document changes to which signature generation is particularly sensitive depend on the actual algorithm involved. We focus our discussion on the I-Match technique, which will be described next.

2.1 I-Match: generating signatures with a lexicon

I-Match [2] relies on the collection statistics of a document corpus to identify a set of terms, called a *lexicon*, which tend to correlate with the gist of a document without being too general (and thus too frequent) and without being too specific (which might equate to noise). Often a lexicon corresponds to a range in the Zipfian ranking of terms induced by the document collection, but this is not strictly necessary and alternative choices of a lexicon can yield comparable levels of performance [8].

In its basic form an I-Match signature is derived from the intersection of the set of unique terms contained in a document and the I-Match lexicon. The process can be described as follows:

1. The collection statistics of a large document corpus are used to define an I-Match lexicon, $\mathcal{L}$, to be used in signature generation.
2. For each document, d, the set of unique terms $\mathcal{U}$ contained in d is identified.
3. I-Match signature is defined as a hashed representation of the intersection $\mathcal{S} = (\mathcal{L} \cap \mathcal{U})$, where the signature is rejected if $|\mathcal{S}|$ falls below a user-defined threshold.

The signatures can be unreliable of the overlap set $\mathcal{S}$ is too small. One can resort to not generating a signature in such cases. Alternatively, as proposed in [9], a larger secondary lexicon is maintained and its intersection with $\mathcal{U}$ is used to enrich the overlap set such that a reliable signature can be generated. The terms constituting the secondary lexicon are typically less frequent than the ones comprising the primary one.

3. FRAGILITY OF I-MATCH

From the definition of I-Match provided in Section 2.1 it is clear that any modification of the original document affecting its intersection with the lexicon will alter the resulting signature. Such document alterations may be due to intentional or accidental misspellings, parsing imperfections (e.g., mixing of content and markup), formatting noise, as well as intentional attacks. One can limit the extent to which I-Match is affected by a careful choice of the lexicon, word-level distributional clustering, or by utilizing several alternative I-Match lexicons at the same time [8]. Here we consider an alternative approach, noting that I-Match lexicon selection favors non-function gist words that often tend to occur in specific usage patterns or in certain contexts. We therefore propose to not only discover the lexicon terms but also to restrict their valid context. While, due to the sparsity of textual data and the flexibility of natural language, discovery of all "valid" usage of any particular word may be impossible, given that a document typically intersects with several lexicon words we can expect that at least some of them will re-occur in a context they had been observed in the past. At the same time, restricting the valid context allows one to account for the fact that near-duplicate detection can be applied to documents representing a particular domain, e.g., one in which word usage is naturally restricted.

3.1 Hardening I-Match by context

Let us define a document d of length N as a sequence of terms:

$$d = [t_1 t_2 ... t_N]$$

With each term t_i we associate two bigrams: $b_i = [t_{i-1} t_i]$ and $a_i = [t_i t_{i+1}]$, corresponding to the preceding and following context with which t_i is found in d. In cases where $i = 1$ or $i = N$, the preceding and following terms are defined by special symbols, corresponding to the beginning or the end of the document, respectively. We say that t_i appears within the *expected* or *valid* context if

$$P(b_i) \geq \theta \quad \text{or} \quad P(a_i) \geq \theta \tag{1}$$

where P denotes a probability estimated over the training corpus and θ is a user-defined threshold. The requirement could be made more complex by insisting that both $P(a_i)$ and $P(b_i)$ exceed the threshold and/or by allowing different threshold values depending on the type of context considered.

A content-enhanced I-Match follows the same steps that were described in Section 2.1, except that the intersection set $\mathcal{S}$ is post-processed to retain only lexicon terms that were found in document d in their expected context. Because of the effect of data sparsity on context estimation, the size of a lexicon used in conjunction with the context-enhanced I-Match may have to be higher when compared with regular I-Match so that null intersections with the lexicon can be avoided. The steps involved in the context-enhanced I-Match are thus as follows:

1. The collection statistics of a large document corpus are used to define an I-Match lexicon, $\mathcal{L}$, to be used in signature generation.
2. For each lexicon term, the set of valid contexts is found using the same document corpus used to derive $\mathcal{L}$, or alternatively another tuning collection.
3. For each document, d, the set of unique terms $\mathcal{U}$ contained in d is identified.
4. The intersection $\mathcal{S} = (\mathcal{L} \cap \mathcal{U})$ is filtered to retain terms found within their expected context (i.e., the context in which their appear within d should correspond to one of the valid contexts). The pruned intersection is denoted by $\mathcal{S}_c$.
5. I-Match signature is defined as a hashed representation of the intersection $\mathcal{S}_c$, where the signature is rejected if $|\mathcal{S}_c|$ falls below a user-defined threshold.

One of the questions regarding context-enhanced I-Match relates to the amount of resources needed for maintaining the context. On the one hand, if the valid context is broad (e.g., hundreds of context words per each lexicon word) the I-Match system will be taxed by the need to maintain large

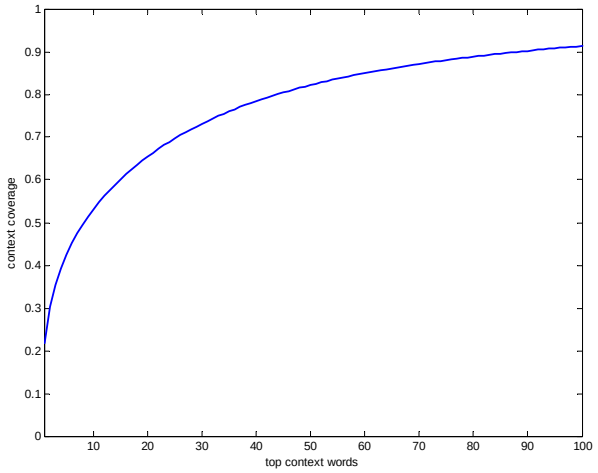

Figure 1: Mean coverage of context for 15K words constituting an example I-Match lexicon as a function of the top most frequent context words considered.

amounts of data for reference and lookup during signature generation. On the other hand, if context tends to be narrow (e.g., a handful of dominant context words capturing the vast majority of valid usage of a lexicon word), it may be too trivial to discover by the attacker and hardening will bring only a small improvement when compared to standard I-Match. Fortunately, in practice context tends to be well behaved. Figure 1 provides a concrete example for an I-Match lexicon of 15,000 words, whose context was estimated with a corpus of 27,518 documents. For each lexicon word the context words were ranked according to their frequency of occurrence and the fraction of context captured by considering just the top N context words was measured. Figure 1 shows mean of this fraction across the whole lexicon as a function of N. It can be seen that including just a few top-context words captures only a small fraction of the valid usage. At the same time, accounting for a moderate number of context words (e.g., around 80) captures a significant majority of the context while being unlikely to exhaust the resources of the I-Match system.

4. ASSESSING THE COMPLEXITY OF I-MATCH ATTACKS

4.1 Objectives of the attacker

Word attacks against signature-based systems differ from the ones targeting content-based spam filters. In the case of the latter one hopes to identify a set of "good" words that collectively outweigh the evidence of spam contained in a message, conditioned on the original message. Once discovered, the combination can be re-used more or less unchanged till the filter adapts to this new form of spam. One can also extend it to discover a set of words that are "good" as far as the filter is concerned, with the view of using them to identify the set of "good" words sufficient to push any particular spam message over the decision threshold.

When breaking a signature based system, the attacker attempts, at the minimum, to identify a modification to the content of a particular message sufficient to alter its signature. The nature and extent of the change are dependent on the signature algorithm that is being compromised. For example, in our earlier work a method of increasing signature robustness through the use of multiple signatures was described [8]. In such a scheme, the attacker would have to break all of the signatures generated in order to foil the defences.

In this work we focus our attention on a setup where only one signature per document is generated. Additionally, we concentrate on the I-Match algorithm where signature generation is equivalent to computing a set intersection between words contained in a document and the contents of the I-Match lexicon. Due to this feature of I-Match, a signature can be altered by finding just one word that is not present in the document but belongs to the lexicon and adding it to the document. It can also be altered by removing or modifying one of the lexicon words already in the document. We will initially focus on the former, assuming that the attacker/spammer starts with a "payload" that needs to be delivered intact and surrounds it with "noise" content, possibly crafted on a per-message basis, with the goal to maximally fragment the signatures associated with messages carrying the same payload. Ideally, each such message should be sufficiently different from any other, so as to make all of the signatures distinct. The approach of utilizing words already present in the document will be discussed in Section 4.4.

4.2 Setting up an I-Match attack

We will assume that the attacker knows that the underlying system is using I-Match and is aware of the algorithm mechanics. The system being attacked utilizes a lexicon $\mathcal{L}$ containing L elements (words or terms). The attacker starts with a pure payload document that should not be altered and attempts to extend it by inserting words. We will assume that the attacker has a way of determining if document alteration leads to a change in signature. E.g., if the payload represents a spam message already blocked by the system, a successful message alteration will cause the message to go through (assuming this is the only filtering technology operating on the message at that time). The attacker attempts to identify a potential vocabulary $\mathcal{V}$ of N elements that is likely to contain the I-Match lexicon. There is an inherent uncertainty involved in such a process, but for the sake of an argument we will assume that $N >> L$ and that $\mathcal{L} \subset \mathcal{V}$. In practice, the attacker cannot be expected to be so lucky and the intersection between $\mathcal{V}$ and $\mathcal{L}$ may not fully contain $\mathcal{L}$ and it is likely that only a partial overlap with the lexicon is achieved.

Given a signature of the payload document as the test medium, the goal of the attacker is to identify the vocabulary of the lexicon (minus the part of the vocabulary that participated in generation of the signature for the template). The minimum useful result is to identify at least one extra word (which would allow one to split the campaign into two different variants) with the ideal goal of identifying all of the vocabulary words if possible.

We will concentrate on the complexity of satisfying the minimum requirement. With the finite reservoir of vocabulary to try, in a sampling without replacement model (gov-

erned by the hypergeometric distribution), the probability of a randomly chosen word to intersect with the lexicon (i.e., success at first trial) is $p = \frac{L}{N}$ and the probability that the first success will be achieved on the kth ($k > 1$) trial is

$$\frac{L}{N-k+1}\prod_{i=1}^{k-1}\left(1-\frac{L}{N-(i-1)}\right) \quad (2)$$

The quantity of interest, however, is the expected number of events needed to find a word intersecting with the lexicon. The number of unsuccessful trials before a lexicon intersection takes place is controlled by the negative hypergeometric distribution with the expectation of:

$$U = \frac{N-L}{L+1} \quad (3)$$

Therefore, the expected number of sampling events till (and including) the first success is equal to

$$T = U + 1 = \frac{N+1}{L+1} \quad (4)$$

The value of T depends thus on the precision of bracketing the original vocabulary by $\mathcal{V}$. This will depend on the knowledge the attacker possesses of how the original lexicon was selected and whether or not they have access to the same document collection. To provide an example, let us assume that the attacker has access to the same document collection used in deriving the I-Match lexicon. It is then reasonable to expect that the I-Match lexicon was created by ignoring all words that occurred just once and ignoring top frequency words. For a large document collection this is still likely to leave on the order of 10^6 words from which the actual lexicon was selected. The lexicon itself is likely to contain on the order of $10^4 - 10^5$ words. In this example the value of T would thus be around 10-100. The attacker might be more accurate in identifying the candidate set, however, especially having experimented with the algorithm themselves. A more pessimistic estimate would thus place T in the range 1-10.

4.3 Attacking context enhanced I-Match

Let us now assume that the attacker suspects that context-enhanced I-Match is being used by the system under attack. In addition to the vocabulary reservoir containing the lexicon, the attacker will also attempt to estimate the context for each candidate lexicon word. This will result in extra C words/terms on average per each element of $\mathcal{V}$. At the very minimum, the context set should be comprehensive enough so that valid context for at least one lexicon term is covered. A more reasonable goal is to aim at covering valid context for each lexicon word, with the ideal goal of covering full context for all of the lexicon words.

Let us focus on the moderate goal of discovering at least one valid context for each lexicon word and let us assume that the attacker is successful in covering the valid context. Since it is possible that out-of-context terms are covered as well (e.g., due to the differences between corpora used by the attacker and defender and due to the uncertainty about the cut-off thresholds in (1)), there will be a probability $\frac{1}{C} \leq p_h \leq 1$ of selecting a valid context from the context candidates for a lexicon term. Using the geometric distribution to estimate the number of trials till first success, the attacker will have to attempt $1/p_h$ context words on average before finding valid context, assuming the context is sought for a valid lexicon word. On the other hand, if the context is explored for a word not present in the I-Match lexicon, the attacker will exhaust all C possibilities, on average, without achieving success. Thus, the expected number of attack attempts till discovering a valid context for a lexicon word will be

$$T_{context} = (T-1)\cdot C + \frac{1}{p_h} \quad (5)$$

When $p_h \gtrapprox \frac{1}{C}$ the expected increase in the the cost of the attack compared to attacking regular I-Match will be

$$gain = \frac{T_{context}}{T} = C \quad (6)$$

If we can assume $T \gg 1$ then we should expect to have $gain \approx C$ for other values of p_h as well.

This analysis suggests that the benefit of enhancing I-Match by context rests on the difficulty of estimating context for the lexicon terms. If the terms (and the general class of terms they belong to) are chosen such that they go together with a handful of dominant context words, the attacker should have little difficulty of capturing the dominant contexts, which will keep the value of C small (say 1 or 2). On the other hand, if usage context does not exhibit a clear mode and, moreover, if the valid context is at least somewhat dependent on the particular document collection used in deriving the lexicon, the attacker may be forced to consider quite a few possibilities in order to ensure success, thus pushing the value of C higher. As illustrated in Figure 1, realistic values of C can be expected to be more than just a handful (e.g., at least 50 in this particular example) which makes the impact of context-enhanced I-Match quite significant.

4.4 Lexicon discovery with multiple payloads

The attacker may also consider reconstructing the I-Match lexicon based on the contents of several different payloads. As mentioned in Section 4.1, a strategy alternative to adding extra content to a fixed payload is to remove some of the payload's content so that the signature is altered. With a single payload this would allow one to recover portions of the lexicon that could not be used to fragment the signatures of messages carrying the same payload. With multiple payloads, however, portions of the lexicon discovered with one payload could be used to attack the system to deliver different payloads.

The main advantage of the payload-centric attack is that the payload is likely to contain relatively few words compared to the vocabulary set $\mathcal{V}$ containing the complete lexicon. If the goal is just to discover lexicon words overlapping with a given document, the task is relatively straightforward since one can simply try removing words or word pairs and monitor any signature changes. On the other hand, the discovery of the complete lexicon using such methodology may take much longer since it is strongly dependent on the content and the number of different payloads used in the process. In fact, the complexity of reconstructing the full lexicon is the same as when using $\mathcal{V}$ directly, since one has the insure that the payloads cover $\mathcal{L}$, and given that the attacker's knowledge of $\mathcal{L}$ is expressed as via the vocabulary set $\mathcal{V} \supset \mathcal{L}$, the attacker has to examine sufficiently diverse payloads in order to cover $\mathcal{L}$. If the attacker follows the strategy of examining each candidate word in turn in conjunction with its pre-estimated context (i.e., C options on

average per word) then also in this case the attack against context enhanced I-Match can be expected to have the level of difficulty increased by a factor of C (6). Note that if the context provided for a lexicon word in the payload matches one of the valid contexts, the discovery is instantaneous, but there is no guarantee thereof. The attacker may insist on using certain words in only certain contexts and if these do not pass the validity criteria, some of the lexicon words may take much longer to be discovered or may be not discovered at all.

Note that in the case of context-enhanced I-Match, if the attacker operates using word pairs as the basic building block, even if a particular insertion or removal is successful, there may still be ambiguity as to which of the pair is a lexicon word and which only provides its context. Also, a context for a lexicon word can also be a lexicon word in its own right.

4.5 Improving the system defences

In practice the attacker may have to face a more formidable task. It can be difficult to verify if modifications to the contents lead to a change in signature (which does not have to be exposed directly) and even in the case of spam filtering, if the attacker can observe whether a particular document modification is or is not successful, there is an inherent uncertainty as to which of the many detection mechanisms is in fact responsible (there are typically more than one). In the case of regular I-Match with a single lexicon, the defender can practice lexicon rotation, since comparable performance can be realized with alternative lexicon choices. In such a setup, reconstructing the contents of any particular lexicon is likely to pay off only for a limited amount of time. For I-Match systems operating with several alternative lexicons concurrently the attacker has to effectively guess their union. For a K lexicon system this may be less than K times the effort of guessing a single one due to the possibility of partial overlap between the lexicons, e.g., as suggested in [9].

Additional difficulty for an attacker lies in the uncertainty with regards to the distribution of content used to generate the lexicon and its context. While freely available corpora may provide a reasonable approximation, the defender can exploit any potential differences to their advantage. For example, in spam filtering one might consider using lexicons corresponding to primarily spam or non-spam vocabulary. Also, the definition of context can be more involved than the one discussed here and the defender may apply different definitions of context at different times. Even with a single fixed lexicon, this could lead to a significant increase in difficulty for the attacker.

5. EXPERIMENTAL RESULTS

We collected a dataset of email spam complaints that have been reliably pre-clustered on a daily basis throughout the year 2005 into individual campaigns, with the possibility of a campaign spanning multiple days. Regular I-Match with a $15,000$ word lexicon was applied the campaign data and only campaigns exhibiting fragmentation level exceeding 4 distinct signatures per campaign were retained. The I-Match lexicon was derived using the Mutual Information criterion as described in [8], based on a collection of $18,555$ legitimate and $18,461$ spam emails (same data as in [8]). A corpus of $27,518$ legitimate emails was used to estimate the usage for each of the lexicon words, which identified $2,327,645$ word

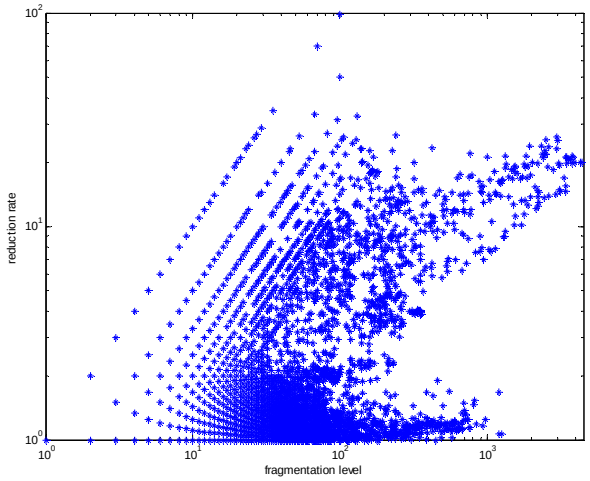

Figure 2: Scatter of fragmentation reduction rate vs. the original fragmentation level measured by the number of distinct I-Match signatures per email campaign. The benefit of content-enhanced I-Match is particularly apparent for high level of signature fragmentation.

bigrams with a lexicon term appearing as the first or the second component of the pair. In context validation only those bigrams were considered whose frequency exceeded or equaled 15.

The context-enhanced I-Match was applied to the email collection and for each campaign the reduction of signature fragmentation level was measured. The reduction rate was defined as:

$$frag\_reduction\_rate = \frac{\text{original signature count}}{\text{reduced signature count}}$$

The scatter plot of signature reduction rate vs. the original fragmentation level is shown in Figure 2. It can be seen for many highly fragmented campaigns (100 or more original signatures) the reduction rate is also very high (10-20 fold), which indicates that the reason for high fragmentation can be attributed to out-of-context word usage. The best overall reduction rate was as high as 99.

Figure 3 shows the dependence of the fraction of campaigns on the fragmentation reduction rate. A greater than 1 reduction rate is achieved for 55% of the campaigns with an apparent power-law like relationship, whereby a very large reduction is a achieved for few campaigns with the majority receiving a small-to-moderate level of reduction in fragmentation. For the purpose of generating Figure 3 the data were smoothed such that reduction rates smaller than 2 were binned with the precision of 0.2, while reduction rates greater than 2 were rounded to the nearest integer.

Although context-enhanced I-Match makes a big difference for highly fragmented campaigns it is also apparent that it makes little difference for campaigns with low fragmentation levels. It is possible that this is due to the fact that in those cases the spam is sent in a few significantly different variants of the payload and additional randomization in the form of random text insertion or corruption of the pay-

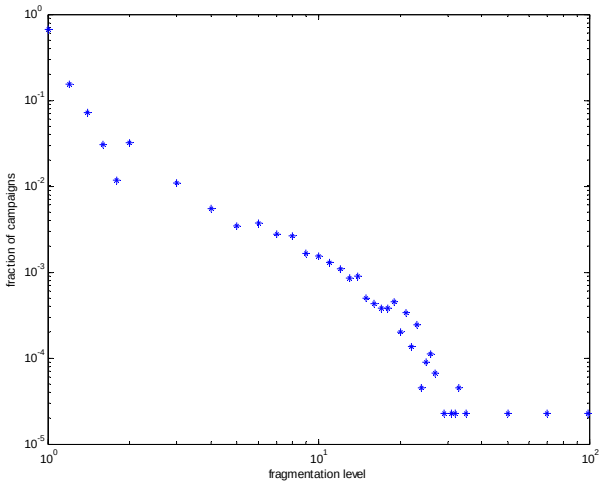

Figure 3: Fraction of campaigns affected by a particular fragmentation reduction rate. A power-law like dependence indicates that while in most cases the gains are moderate, there are significant number of campaigns where the benefits of context-enhanced I-Match can be very high.

load. Figure 4 illustrates a case where contenxt-enhanced I-Match is particularly effective. Here the noise words might be considered "good" by some filters when taken individually. However, their usage is unusual enough for the context-enhanced I-Match to effectively filter out the noise blocks in the majority of cases.

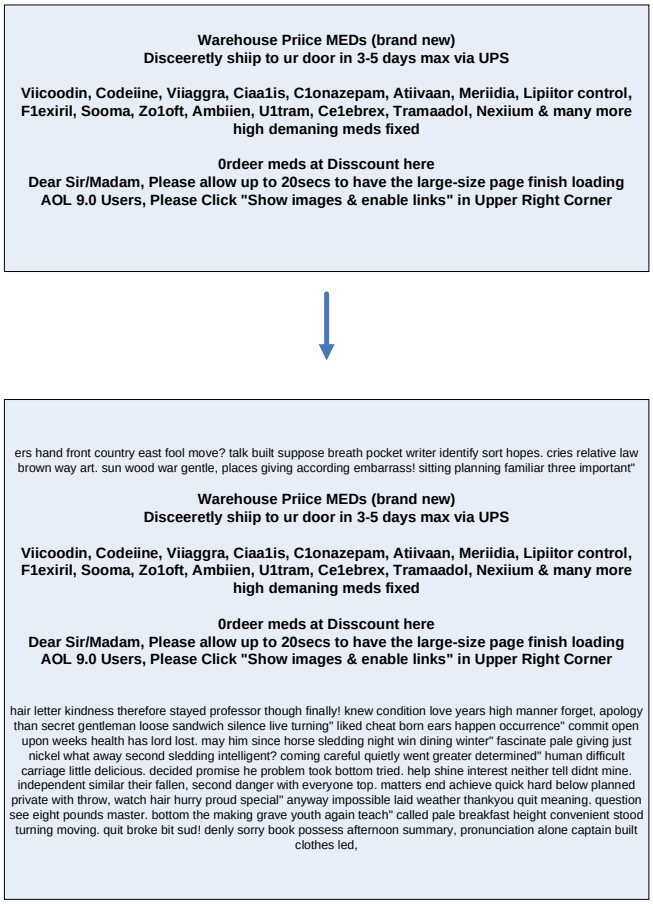

Figure 4: Example of a spam campaign against which context-enhanced I-Match appears to be particularly effective. The top square shows the actual payload of the campaign with the bottom one providing an instance of the payload surrounded by noise content. When applied by the spammer in the real-world, the noise content is often made invisible to the recipient by blending it with the color of the background.

6. RELATED WORK

Although spam filtering has been analyzed from the machine learning perspective for quite some time, explicit accounting for the adversarial aspects of this task is fairly recent [3]. Attacks can range from those targeting message encoding and feature extraction to those targeting the distribution of content. The latter ones have been studied most extensively[5][11] and, in particular, attacks trying to outweigh the evidence of spaminess with evidence of non-spaminess of content received much attention due to their practical importance. Published results indicate that vulnerabilities of content-based statistical spam filters to such attacks are quite real, but they can be mitigated by frequent adaptation to changing content[11]. In [13] it was advocated that spam and non-spam words typically occur in close proximity to other spam or non-spam words, which suggests a document preprocessing methodology where words placed in an out-of-class context are not taken into consideration when classifying a message. This is related to the ideas discussed in this work, but differs in its conditioning of context on the existence of a classification model.

Susceptibility of signature-based spam-detection systems to message alteration has been studied to a lesser extent [6], although such systems have found widespread use in practice (e.g., [12]). Careful feature selection and use of multiple alternative feature sets have been advocated [8], but assessing the vulnerabilities of many practical algorithms is difficult due to their proprietary nature. The use of feature selection to guard against noise has long been studied in the areas of text categorization and Information Retrieval and recent results indicate that selecting features in the context of a particular document (while also accounting for training data statistics) can have particularly beneficial effects on classification performance [10].

7. CONCLUSIONS

By introducing context filtering of lexicon words within a document we were able to substantially decrease the level of I-Match signature fragmentation for heavily randomized spam campaigns while also stabilizing signature generation for the less heavily randomized ones. Over the dataset considered in our experiments the reduction was as high as 99 fold for some of the campaigns, although the reduction level is clearly data dependent. Whereas it is still possible to attack the context-enhanced I-Match by attempting to esti-

mate the valid context, we showed that this represents a task significantly more difficult than simply trying to guess the content of a lexicon for regular I-Match. The enhancements were achieved without adversely affecting the complexity of signature generation.

Aside from making direct attacks more difficult there are also potential non-adversarial benefits of relying on context when computing I-Match signatures. In many cases signature fragmentation is caused by imperfections of parsing and feature extraction. In such situations context validation is likely to help excluding words appearing in meta-tags, formatting artefacts, decorations, etc. Assessment of the impact of context-enhanced I-Match on the performance of near-duplicate detection in non-adversarial applications will be the subject of future work.

8. REFERENCES

[1] 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.

[2] 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.

[3] N. Dalvi, P. Domingos, M. Sanghai, and D. Verma. Adversarial classification. In *Proceedings of The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, pages 99–108, 2004.

[4] 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.

[5] J. Graham-Cumming. How to beat an adaptive spam filter. In *MIT Spam Conference*, 2004.

[6] R. J. Hall. A countermeasure to duplicate-detecting anti-spam techniques. Technical Report 99.9.1, AT&T Labs Research, 1999.

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

[8] A. Kołcz, , A. Chowdhury, and J. Alspector. The impact of feature selection on signature-driven spam detection. In *Proceedings of The First Conference on Email and Anti-Spam (CEAS-2004)*, 2004.

[9] A. Kołcz, , A. Chowdhury, and J. Alspector. Improved robustness of signature-based near-replica detection via lexicon randomization. In *Proceedings of The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, pages 605–610, 2004.

[10] A. Kołcz. Local sparsity control for Naive Bayes with extreme misclassification costs. In *Proceedings of The Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2005)*, pages 128–137, 2005.

[11] D. Lowd and C. Meek. Good word attacks on statistical spam filters. In *Proceedings of The Second Conference on Email and Anti-Spam (CEAS-2005)*, 2005.

[12] V. Prakash and A. O'Donnell. Fighting spam with reputation systems. *ACM Queue*, 3(9), 2005.

[13] J. Zdziarski. Bayesian noise reduction. In *MIT Spam Conference*, 2005.

[14] F. Zhou, L. Zhuang, B. Zhao, L. Huang, A. Joseph, and J. Kubiatowicz. Approximate object location and spam filtering on peer-to-peer systems. In *Proceedings of ACM/IFIP/USENIX International Middleware Conference (Middleware 2003)*, 2003.