The Impact of Feature Selection on Signature-Driven Spam Detection

Aleksander Kołcz[1], Abdur Chowdhury[1], and Joshua Alspector[1]

AOL, Inc., 44900 Prentice Drive, Dulles VA 20166, USA.

Abstract. Signature-driven spam detection provides an alternative to machine learning approaches and can be very effective when near-duplicates of essentially the same message are sent in high volume [20]. Unfortunately, signatures can also be brittle to small alterations of message content. In this work we propose a technique for increasing signature robustness, targeting the I-Match algorithm [6], but applicable to other single-signature detection schemes. The proposed method is shown to consistently outperform traditional I-Match in the spam filtering application. As I-Match signature quality and stability depend on vocabulary control, we compare the traditional Zipfian approaches to feature selection with techniques applied typically in text categorization, which are found to provide viable alternatives. In particular, distributional word clustering is demonstrated to be effective in increasing signature robustness.

1 Introduction

While the research focus in spam filtering has been primarily in content classification [1], many effective approaches (e.g., Distributed Checksum Clearinghouse (DCC)[1] or Vipul's razor[2]) tend to emphasize the basic fact that spam very often consists of highly-similar messages sent in high volume. Although content-based personalized spam filters can be very accurate [19], from the viewpoint of an Email Service Provider (ESP), it is often desirable to detect and eliminate spam at the entry-point to their system so that no storage/computational resources are wasted on unsolicited messages. Given the examples of known current spam, it is therefore desired to prevent any messages that "look like these" from being treated as legitimate. Naive exact-matching approaches are of limited value, however, since the spam messages are rarely identical, precisely to avoid template-based detection schemes. Thus, given a set of prototypes, the spam filtering problem can be seen as a special case of near-duplicate document detection [4][3][6], which has been studied extensively in data-mining and information retrieval applications.

The focus of duplicate-detection schemes varies from providing high detection rates to minimizing the computational and storage resources needed. With massive amounts of data, 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 message content. In the spam filtering application, where the adversary often purposefully randomizes the content of individual messages to avoid detection [11], 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 message alterations at the cost of increased signature size. We demonstrate consistent superiority of the proposed approach over the original scheme as well as its attractiveness for the target application of spam filtering. We also investigate the utility of distributional word clustering as a means of further increasing signature invariance.

The paper is organized as follows: Section 2 outlines the near-duplicate approach to spam detection. Section 3 describes the I-Match algorithm, with Section 4 focusing on our contribution to decreasing signature brittleness and enhancing the lexicon selection process. In Section 5 the experimental setup is outlined, with the results presented in Section 6. The paper is concluded in Section 7.

---

[1] http://www.rhyolite.com/anti-spam/dcc/

[2] http://razor.sourceforge.net/

2 Spam filtering by “copy detection”

Machine-learning approaches to spam detection [1] rely on inducing classification models from labeled data provided by the users. In a number of studies, several learners, including Naive Bayes[16], Support Vector Machines [8][15] and AdaBoost [5], have been shown to be quite effective at this task (especially as personal filters). Due to the adversarial nature of spam, classifiers need to constantly adapt to changes in spamming tactics, particularly where (primarily textual) feature extraction and selection are concerned. Also, as pointed out in [9], the prevalence of spam changes over time (and can be difficult to measure), which makes cost-sensitive spam classification quite challenging. Thus the effectiveness of a filtering system is likely to fluctuate and there may be periods of time when one or more filters adapt to the changes in the observed distribution or characteristics of mail.

Although acquisition of non-spam may pose privacy problems, it is usually easy to collect examples of spam that penetrate a system’s defenses. These can be collected, for example, via honeypot mail accounts that should not be receiving any legitimate mail[3], or via direct user feedback, since the user community is usually eager to report or complain about unsolicited mail. While such data can be incorporated into a variety of filter adaptation processes, they provide an immediate source of *spam queries* that can be applied to the email stream [20]. This is particularly attractive in the context of a system serving a large community of users, since incoming messages that can be verified to be near-replicas of known spam can be stopped at the very entry-point to the system. The query-by-example model is also attractive when the unsolicited nature of an email is difficult to determine from the contents of the message alone, which can potentially reduce the effectiveness of content based classifiers.

The near-duplicate approach to spam detection is certainly vulnerable to dedicated spamming attacks (such as frequent content alteration, in extreme cases on a per message basis). Nevertheless, it can be argued that it represents a viable filtering approach, especially as a component of a larger hybrid system.

3 I-Match

The problem of finding duplicate, yet non-identical, documents is not unique to the spam filtering domain and has been the subject of research in the text-retrieval and web-search communities, with the application focus ranging from plagiarism detection in web publishing to redundancy reduction in web search and database storage. A number of solutions have been proposed, ranging from similarity based approaches [17][14] to techniques that trade-off detection accuracy for computational efficiency by decomposing documents into units larger than words (word n-grams, sentences or paragraphs) and performing partial matching over such representations. In particular, shingle or fingerprint based techniques [7][10][12], such as COPS [3], KOALA [13], and DSC [4], have been successfully applied to very large dynamic document repositories.

Similarity-based duplicate detection inherently maps each document to one or more clusters of possible duplicates, depending on the choice of the similarity threshold. The I-Match [6] approach produces a single-hash representation of a document, thus guaranteeing that each message will map to one and only one cluster, while still providing the 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 (see Section 4.1), L, to be used in signature generation.
2. For each message, d, the set of unique terms U contained in d is identified.
3. The 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.

[3] www.brightmail.com

4 Decreasing the fragility of I-Match signatures

Ideally, the signature of a message should be insensitive to small changes in content. In the context of spam-filtering these include changing the order of words, as well as inserting or removing a small number of words. I-Match is inherently insensitive to changes in the word order, but inserting or deleting a word from the I-Match 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 purposefully randomize messages with respect to the lexicon's vocabulary.

To address the issue of signature brittleness we note that experimental data [6] suggest that similar levels of duplicate detection accuracy can often be obtained by largely non-overlapping lexicons. When the set of signatures due to such lexicons is considered, a small modification to message content may change the signature due to any particular lexicon but, at the same time, there may exist a number of alternative lexicons for which the signatures may be unaffected by such a change. Thus a collection of signatures can be expected to be more stable than a single signature.

In practice, obtaining a number of independent lexicons may be non-trivial. It can reasonably be expected, however, that if a lexicon is modified by a small number of additions/deletions, this is unlikely to significantly change the stability of I-Match signatures with respect to the modified lexicon. We therefore suggest a setup where a suitable lexicon is found and then K different perturbations of the original lexicon are derived by randomly eliminating a fraction p of terms from 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. The extended I-Match signature of a message in this 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 (other voting schemes could also be considered).

To examine the sensitivity of the extended signature to content alterations let us consider a message that is modified 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:

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

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.

4.1 I-Match feature selection choices

The effectiveness of I-Match relies on the appropriate choice of the lexicon. Experimental data suggest that one effective strategy is to impose an upper and lower limit on the inverted document frequency (*idf*) for words in the document collection, since terms with very low *idf* and terms with very high *idf* tend to be less useful in duplicate detection than terms with mid-range *idf* values [6]. Although high-*idf* terms may be very effective in pinpointing a particular document, they also capture misspelled words and other spurious strings, which reduces their value in identifying near rather than exact duplicates. This is especially important in the spam filtering application, where much of the randomization is created on purpose, precisely to make copy detection more difficult. Since there are no firm guidelines with regards to the choice *idf* cutoff points, the choice of a lexicon typically involves a degree of trial and error.

An effective spam-fighting system is likely to employ a variety of filtering techniques, where template-based filtering may represent just one component. In particular, inductive classifiers require that a training set of both spam and non-spam examples is collected and used in the training process. Generally, this involves a degree of feature selection to remove "noisy" features and prevent the classifiers from overfitting. A question arises as to whether the set of features selected for their discriminative efficacy might also be effective from

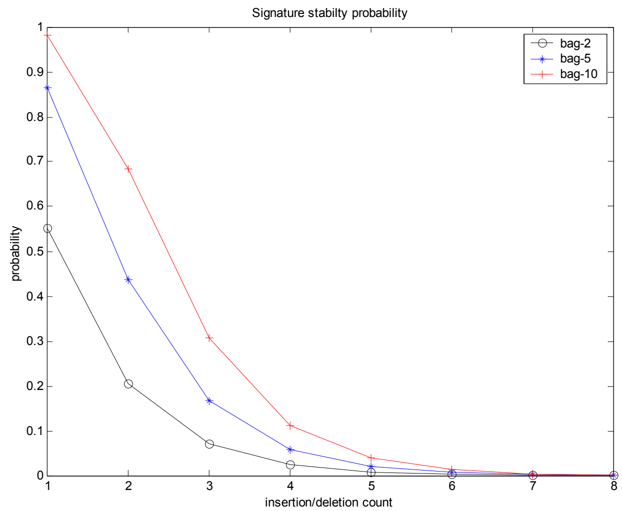

Fig. 1. Stability of extended I-Match signatures under random insertion/deletion of words in the original message for the case of $p = 0.33$. The y-axis corresponds to the probability that the extended I-Match signature will not be affected by a change to the message contents.

the standpoint of identifying similar documents. The standard method of I-Match lexicon selection ignores the very frequent and very infrequent terms, without taking their discriminative power into account. On the other hand, feature selection techniques, such as those based on the popular Mutual-Information (MI) criterion [16] tend to select features that are both fairly frequent and, at the same time, are effective for discrimination between the classes. Note that, similarly to the standard I-Match approach, feature selection techniques are likely to eliminate globally very frequent and globally very infrequent features.

The randomized I-Match signature scheme provides greater robustness to term additions and deletions. Its effectiveness as a countermeasure to word substitutions is less robust, however, since a substitution is equivalent to an addition-deletion combination. If the substitutions were limited to synonyms, one could expect that application of thesauri should be beneficial. Since spam often contains intentional misspellings, spell-checking and "eye-space" normalization (e.g., víägrå→viagra) might also be effective. On the other hand, one can follow a data-driven approach, where equivalence relations between features are found based on a training sample. In this work we considered the distributional feature clustering technique based on the Agglomerative Information Bottleneck (IB) algorithm [18], which have been shown to be effective in the text categorization domain [2]. The IB technique chooses word clusters so as to maximize the Mutual Information between the feature clusters and the class while insuring that relevant properties of the original word distribution are preserved by the new representation. Note that although IB-induced clusters may contain synonyms, they can also group words that are semantically unrelated to one another. IB has been found most effective as an alternative to ranking-based feature selection in the classification domain in cases where relatively many features are important to achieve high accuracy discrimination [2]. Given the multi-faceted nature of spam, this has been our key motivation of applying IB to I-Match lexicon selection.

5 Experimental Setup

In the following we evaluate the near-duplicate detection accuracy of the modified and extended I-Match using an email document collection, where detection using single and multiple randomized signatures is compared. We examine the generalization properties of I-Match based on a lexicon derived from an unrelated large document collection, and compare it with lexicon selection based upon the Mutual Information criterion, with and without IB-based word clustering.

5.1 Datasets

We considered the following real-world email datasets:

- The Legitimate email collection consisted of $18,555$ messages and was used primarily in evaluation to assess if near-duplicate detection of spam produced may lead to any false-positives among legitimate emails. It was also used in conjunction with spam data to derive the Mutual Information ranking of words.
- The Honeypot-Spam collection consisted of $10,039$ messages collected in a number of accounts set up to attract spam. These data were known to contain many highly similar messages and were used in evaluating the effectiveness of the near-duplicate detection approaches considered.
- 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 with the explicit goal of finding near duplicates or highly similar/related messages that showed clear evidence of randomization.
- Additionally, a collection of $18,461$ duplicate-free spam messages (independent from the other two collections) was used in conjunction with the legitimate set to induce MI-based ranking of words.

5.2 Document preprocessing

For each message, the set of features comprised all unique words found in the message body and the subject line (where a word was defined as a sequence of alphanumeric characters delimited by white space), with all punctuation and HTML formatting removed. Header-based features, although generally useful in spam detection [16], were not used in this study. 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 (this was used to reduce the possibility of false-positives due to matching on short "noisy" terms). Within each collection trivial duplicates (i.e., documents having the exact same set of words as another document in the collection) were removed. For the Honeypot and Cluster spam datasets, this resulted in a reduction in the number of documents by 47% and from 27%, respectively. Note the high duplicate rate in the Honeypot dataset.

5.3 The evaluation process

The exact point at which two messages cease to be near-duplicates and become just highly similar is difficult to define and 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 signature-based techniques was compared[4]. Our experience suggested that two emails 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 spam 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 spam, i, and an email collection, we define the recall of a signature-based detection technique as the ratio of the number of emails flagged as duplicates of i to the corresponding number identified by the cosine measure, when using message i as a template,

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

[4] The cosine measure was 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. Note that computation of the cosine metric is very expensive and thus not practical when large amounts of data are involved.

Table 1. Duplicate detection accuracy in the *Honeypot/Cluster spam vs legitimate-email* experiments. The SGML columns correspond to *idf*-based lexicons. In the MI/IB-based columns, the results for IB-generated word clusters are shown are shown in italics. Bag-N signifies that N auxiliary (randomized) lexicons were created.

	MI/*IB* honeypot		SGML honeypot		MI/*IB* cluster		SGML cluster	
lexicon cnt	Recall	Utility	Recall	Utility	Recall	Utility	Recall	Utility
1	0.77/*0.69*	25.2/*19.0*	0.66	30.17	0.39/*0.39*	42.3/*38.2*	0.40	46.34
1+2-bag	0.80/*0.83*	27.8/*38.9*	0.72	36.32	0.48/*0.51*	66.7/*89.4*	0.49	81.25
1+5-bag	0.82/*0.89*	30.4/*43.9*	0.76	43.34	0.55/*0.57*	81.0/*105.1*	0.55	96.24
1+10-bag	0.89/*0.93*	37.5/*56.3*	0.80	56.93	0.63/*0.65*	108.9/*142.8*	0.61	113.38

Note that one does not necessarily care if messages assigned to the same group as a spam template are only "just similar" to the template message as long as none of the legitimate messages gets assigned to the same cluster. To account for the operational performance of a copy-detection technique we define a utility function, such that when using a spam message i as a query the utility is given by

$$utility\,(i) = |spam\,(i)| - cost \cdot |legit\,(i)|$$

where the $|spam\,(i)|$ and $|legit\,(i)|$ are the numbers of spam and non-spam messaged extracted, with *cost* defined as the cost of misclassifying a legitimate message as spam. In our experiments, we used the setting of $cost = 100$. Note that the utility function combines the benefit due to elimination of spam with the cost of false positives.

5.4 Signature algorithm settings

Documents having fewer than 5 unique word features were considered as too short and were ignored. Additionally, for a signature to be valid, it had to be based on at least 5 or 10% of the unique words in the message, whichever was greater (this will be referred to as the cutoff threshold).

The standard I-Match procedure employed a lexicon derived from a large dataset of news stories corresponding to TREC disks 4 and 5[5] (previously used in the I-Match experiments described in [6] for its good quality in capturing the statistics of English), whose collection statistics was used to assign to each term a normalized *idf* score in $[0, 1]$, where the primary lexicon was determined by terms in $[0.2, 0.3]$. Results for this lexicon will be referred to as SGML.

For the MI-driven lexicon selection, a $15,000$ term lexicon was generated by ranking the words in the legitimate and spam collections according to their Mutual Information and retaining the top-ranking ones. In applying the distributional clustering IB algorithm, the MI-based lexicon was iteratively reduced to $5,000$ word-clusters.

In experiments with lexicon randomization, the original lexicon was augmented with K copies (with K in $\{2, 5, 10\}$) obtained by bootstrap sampling from the original and ensuring that each term was selected at most once. Each randomized lexicon shared approximately 67% of terms with the original.

6 Results

In both the honeypot and cluster-spam experiments, a random 10% of the spam data was used as queries against the respective spam collection and the legitimate-email dataset. The resulting average values of the recall and utility metrics are given in Table 1. None of the near-duplicate detection configurations produced any false-positive matches against the legitimate email collection, although the generality of this observation will be investigated in a larger-scale future study. Note that I-Match with MI-based lexicon selection performed better according to the cosine-similarity definition of a near duplicate, while the standard I-Match configurations lead to better spam-filtering utility. For both signature generation variants, lexicon randomization provided a clear benefit, both in terms of duplicate detection and spam detection metrics.

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

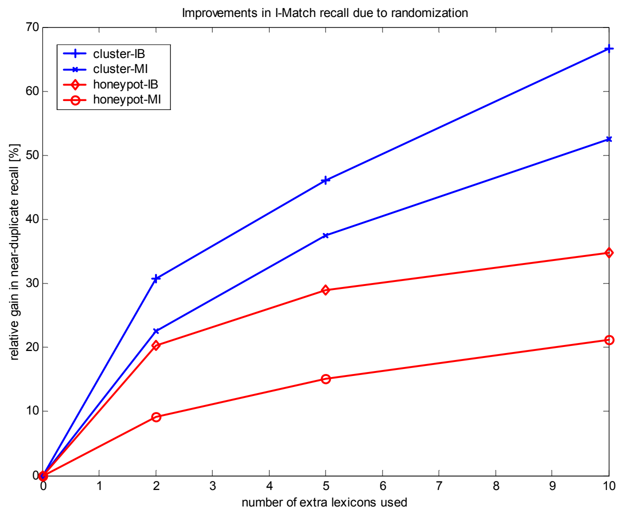

Fig. 2. Relative gain in recall as a function of the number of extra randomized lexicons used for lexicons derived via the MI-driven feature selection process. For features corresponding to IB-based word clusters, improvements due to lexicon randomization are consistently higher.

As indicated by Table 1 and Figure 2, the IB word clustering improved the feature-selection based performance metrics even further. In particular, in most cases the IB approach outperformed the MI/SGML lexicon choices according to both the recall and the utility metrics. Also, as shown in Figure 2, randomization benefits for this type of feature selection were highest.

6.1 Discussion

Given that the I-Match lexicon was derived using a large collection of news articles, it is interesting to observe its good performance in the spam-filtering application, since email documents generally have different characteristics than news. Considering the higher utility values obtained with this choice of the lexicon, we suspect that duplicates detected via this version of I-Match may in fact be "looser" than the ones detected with a collection-specific lexicon. This, however, may be considered beneficial from the spam-filtering perspective, as long as no false-positive legitimate-email matches result from relaxing the notion of a duplicate. We intend to investigate this further in the future.

Our results suggest that in applications where copy-based detection is deployed side-by-side with classification, standard feature selection may offer an attractive way of deriving the I-Match lexicon, especially when a large stable collection of email data is not available. We have also shown that the class distribution of features can be successfully used to cluster words into aggregate features, which can improve the detection accuracy even further. Other forms of clustering and feature normalization might also prove effective.

The results in Table 1 show that by using even a few extra randomized lexicons, both the recall and utility metrics can be improved substantially for all choices of the lexicon. 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. Note that randomization could be re-applied over the course of time, which might in practice be applied as a countermeasure to spammers guessing the composition of the lexicons.

7 Conclusions

We considered the problem of improving the stability of I-Match signatures in the spam filtering application with respect to small modifications to message content. The proposed solution involves the use of multiple I-Match signatures, derived from randomized versions of the original lexicon. Despite utilizing multiple fingerprints, the proposed scheme does not involve direct computation of signature overlap, which makes signature comparison only marginally slower than in the case of single-valued fingerprints.

We also showed that in near-duplicate applications involving document classification, such as spam filtering, term ranking induced by standard feature selection can be used an alternative to traditional *idf*-based method typically employed by I-Match. Notably, the use of word clusters (at least those derived via the Information Bottleneck algorithm) appears to be beneficial in the near-duplicate approach to spam detection. We intend to explore the efficacy of other types of clustering in future work.

References

1. I. Androutsopoulos, G. Paliouras, and E. Michelakis. Learning to filter unsolicited commercial e-mail. Technical Report 2004/2, NCSR Demokritos, 2004.
2. R. Bekkerman, R. El-Yaniv, and N. Tishby. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3:1183–1208, 2003.
3. S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *Proceeding of SIGMOD*, pages 398–409, 1995.
4. A. Broder. On the resemblance and containment of documents. *SEQS: Sequences '97*, 1998.
5. X. Carreras and L. Márquez. Boosting trees for anti-spam email filtering. In *Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, BG, 2001.
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. 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.
8. H. Drucker, D. Wu, and V. Vapnik. Support Vector Machines for Spam Categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.
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. J. Graham-Cummings. The spammers' compendium. In *MIT Spam Conference*, 2003.
12. T. Haveliwala, A. Gionis, and P. Indyk. Scalable techniques for clustering the web. In *Proceedings of WebDB 2000*, 2000.
13. N. Heintze. Scalable document fingerprinting. In *1996 USENIX Workshop on Electronic Commerce*, November 1996.
14. T. C. Hoad and J. Zobel. Methods for identifying versioned and plagiarised documents. *Journal of the American Society for Information Science and Technology*, 2002.
15. A. Kołcz and J. Alspector. SVM-based filtering of e-mail spam with content-specific misclassification costs. In *Proceedings of the Workshop on Text Mining (TextDM'2001)*, 2001.
16. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian Approach to Filtering Junk E-Mail. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, 1998.
17. M. Sanderson. Duplicate detection in the Reuters collection. Technical Report TR-1997-5, Department of Computing Science, University of Glasgow, 1997.
18. N. Slonim and N. Tishby. The power of word clusters for text classification. In *23rd European Colloquium on Information Retrieval Research*, 2001.
19. W. Yerazunis. Sparse binary polynomial hashing and the CRM114 discriminator. In *MIT Spam Conference*, 2003.
20. 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.