

Autonomous Personal Filtering Improves Global Spam Filter Performance

Gordon V. Cormack and Mona Mojdeh
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
{gvcormac,mmojdeh}@uwaterloo.ca

ABSTRACT

Using two email streams, we show that a personal filter trained exclusively on user feedback substantially outperforms ($p \approx 0.000$) three industry-leading global spam filters not using feedback. We show that autonomous personal filters, trained on the output from a global spam filter rather than user feedback, substantially outperform ($p \approx 0.000$) the global filter, if by a somewhat smaller factor than user-feedback-trained personal filters. To our knowledge, no controlled quantitative study addressing these questions has previously been reported.

1. THE QUESTION

On-line personal spam filters [7], while remarkably effective at identifying spam, must be trained in real time by labeling some or all of the messages directed to particular users. The task of labeling, which typically falls on the recipient, may be too difficult or too burdensome for many users, occasioning the use of systemwide filters that demand no user input. Systemwide filters typically employ some combination of pattern-based rules, global whitelists and blacklists, non-user-specific training examples or collaborative filtering [2].

There is some controversy as to whether personal or systemwide filters are more effective, but few controlled studies have addressed the issue. The CEAS 2008 Live Spam Challenge¹ is one of the few that has compared personal and systemwide filters. However, feedback labels were available to the filters, and it is not known in general whether or not these labels were used. We do know that personal filters were among the best performing, and that an industry-leading collaborative filter, which did not use feedback, was among the poorest. In this paper, we address the question, “is it possible to improve global filter performance without user feedback?”

2. THEORY

An on-line personal spam filter classifies messages one at a time, in sequence, as they are delivered to the user’s inbox. In the course of reading the messages, the user provides feedback, labeling some of them as either spam or ham. Very low

error rates – less than 1% overall – are achieved by learning methods that use no information other than the message content and user feedback. The best results are achieved with *ideal* feedback, in which each message is labeled, and labeled accurately, by the user. In practice, a user cannot be expected to label every message, or even most messages. Typically, the user interface solicits feedback only for misclassified messages, or for a small subset of hard-to-classify messages. The rest are assumed to be classified correctly.

Even among the messages actually labeled by users, error rates of 5% or more are typical [10, 21]. Noise-tolerant learning methods have been shown to achieve classification error rates many times smaller than the that of the feedback [3, 17].

Global spam filters use information from a variety of sources to classify messages, including hard-coded rules, blacklists, and feedback from experts or a community of users. None of these sources are specific to any particular user and, as a consequence, global filters exhibit higher error rates than personal filters – of the order of 5% for the industry-leading filters we used.

These observations led us to form the hypothesis that a global filter could be used as a surrogate user, providing simulated “feedback” labels to a noise-tolerant personal filter. We call such an approach an *autonomous personal filter* because it is tailored to the user’s email, but requires no human input.

Our rationale is based on the observation that the global filter error rate is comparable to the feedback noise level for which noise-tolerant personal filters work well. If the errors appear as noise to the personal filter, the personal filter will improve on the global filter. If the errors do not appear as noise, the personal filter will learn to reproduce the errors, yielding no improvement.

The extent to which the errors appear as noise depends on the nature of both the global and personal filter. Filters that rely on independent sources of information and employ different learning methods are more amenable. To this end, global filters that use blacklists and near-duplicate detection are likely better than those that employ pattern-based rules, as a personal filter can learn the patterns much more easily than the blacklists or hashed examples.

For the same reason, weaker personal filters may work better than stronger ones, because they are less likely to learn the non-random nature of the global filter errors. The learning methods employed by personal filters may be broadly characterized as *generative* or *discriminative*. A generative method models the characteristics of ham and spam

¹www.ceas.cc/challenge

independently, and classifies a message according to which model fits best. Naive Bayes (NB) is the best-known generative method [14]; sequential compression methods like Prediction by Partial Matching (PPM) and Dynamic Markov Compression (DMC) [1] are also generative. A discriminative method models only the characteristics that distinguish ham from spam, ignoring the rest. Common discriminative methods are logistic regression (LR) [8] and support vector machines (SVM) [19].

For ideal feedback, the best-performing methods are discriminative, with LR and SVM showing comparable effectiveness [4, 20]. But they perform very poorly with noisy feedback, unless parameters such as learning rate and regularization are adjusted [18]. Of the generative methods, DMC works best for ideal feedback. Unlike the other methods discussed here, DMC treats each message as a sequence of bits rather than a “bag of words” or other predefined tokens. This means, for example, that the method can learn word fragments or sequences of words or characters that might otherwise be lost in tokenization. DMC is less sensitive to training noise than LR or SVM, but has no obvious parameters that can be tweaked. Naive Bayes [14] is a somewhat weaker method, but is hardly affected by 5% training noise.

It is worth noting that the term “Bayesian” has come to be used as a generic term for any learning spam filter, largely due to the influence of Graham [9] and Robinson [13]. Their method, which is nearly ubiquitous in open-source filters, is not strictly a naive Bayes classifier, due to its use of per-message feature selection and normalization. The method we label NB is a simplification of the Graham-Robinson method. A further development of the open-source community is the use of “train on error” and “train until no error” strategies. When trained in this manner, a “Bayesian” filter becomes discriminative rather than generative.

Ensemble methods have been shown to work well for ideal and noisy feedback [11, 3]. In almost every study, the fusion of separate spam filter results yields a better filter than any individual filter. Very simple methods like voting work well; methods that combine filter scores work somewhat better, provided the scores are amenable. If the filters making up the ensemble make independent errors, and have similar error rates, the average of their results will necessarily have a lower error rate. In practice, even if the error rates are vastly different, the fusion filter still improves on the individual filters.

3. DATASETS AND FILTERS

Using two distinct email streams, we tested autonomous combinations of three global filters and four personal filters. For comparison we also tested a state-of-the-art personal filter with real and simulated user feedback.

Datasets. We used two distinct email streams: the CEAS 2008 Live Spam Challenge Laboratory Corpus², and a private corpus which we dub MrX-5. The CEAS corpus consists of exactly the messages used in the CEAS 2008 Live Spam Challenge, for both real-time and laboratory experiments. For the real-time experiment, we configured and deployed an industry-leading desktop plugin spam filter. Using IMAP, we fetched the test messages in real-time, and the

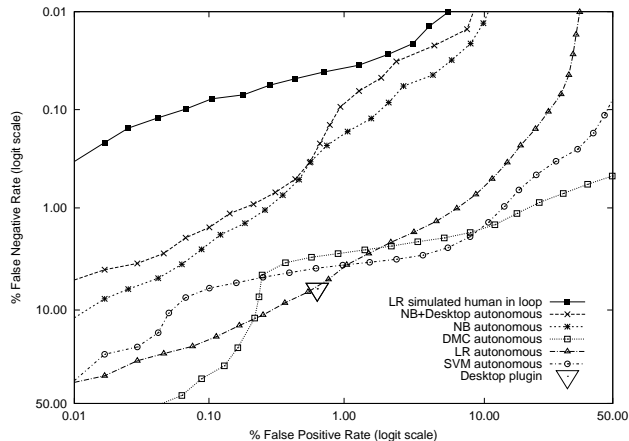


Figure 1: CEAS 2008 Corpus results, with commercial desktop filter baseline. The large triangle represents the performance of a commercial collaborative filter plugin applied to the messages in the course of the CEAS 2008 Live Spam Challenge. The top line represents the performance of a personal filter, trained using on-line active learning on 0.8% of the messages. The other lines represent the performance of autonomous personal filters, trained exclusively on the output from the baseline filter. Among the autonomous filters, the best performance is achieved by summing the results of naive Bayes and baseline filters.

Filter	% fpr	% fnr	fnif
LR simulated human	0.63	0.04	149.8
NB+Desktop	0.63	0.25	25.6
NB autonomous	0.63	0.29	21.9
DMC+Desktop	0.63	0.63	10.1
DMC autonomous	0.63	3.07	2.1
SVM autonomous	0.63	4.00	1.6
LR autonomous	0.63	5.98	1.1
Desktop plugin	0.63	6.64	1.0

Table 1: False negative improvement factor relative to desktop plugin filter on CEAS 2008 Corpus. All factors > 1 are significant ($p \approx 0.000$).

²plg.uwaterloo.ca/~gvcormac/ceascorpus/

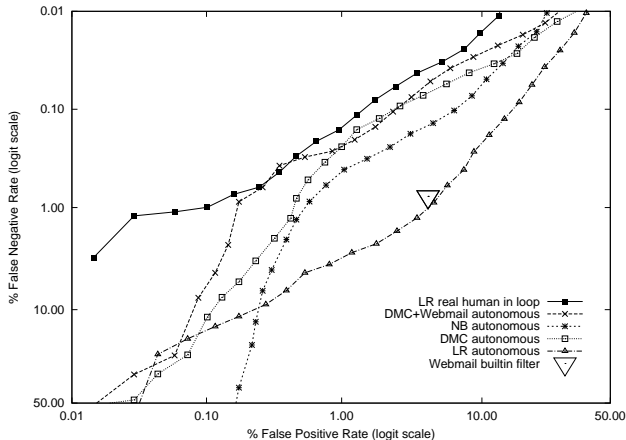


Figure 2: MrX-5 Corpus results, with commercial webmail filter baseline. The large triangle represents the performance of a commercial webmail system when forwarded a stream of personal email in real time. The top line represents the performance of a personal filter, trained by the actual message recipient using on-line active learning on 1.4% of the messages. The other lines represent the performance of autonomous personal filters, trained exclusively on the output from the baseline filter. Among the autonomous filters, the best performance is achieved by summing the results of DMC and baseline filters.

plugin moved those it classified as spam to a junk folder. Due to technical issues, about 10% of the messages were never delivered to the filter; these messages were not considered in our evaluation. The net result was a corpus of 123,100 messages – 23,844 ham and 99,256 spam – with labels indicating the filter result and also the gold standard as adjudicated by CEAS.

For comparison, we also captured the results of the best-performing personal filter (LR) in the CEAS on-line active learning experiment [15]. Each filter was allowed to solicit user feedback for at most 1000 of the messages (about 0.8%). Active learning directly emulates the user interface scenario in which user feedback is solicited from time to time for “hard to call” messages [12]. It also aptly models one-sided feedback, in which a small number of spam messages are deliberately delivered to the inbox so as to solicit user error reports [16]. Although our primary goal is to evaluate autonomous filters, we include the result of this non-autonomous filter for comparison.

Our second email stream, MrX-5, consists of 266,424 messages – 6908 ham and 259,516 spam – delivered to a particular user from July 2008 through March 2009. While this dataset is private, the authors undertake to test other researchers’ filters with it, on request. The same user’s email has been used for the MrX, MrX-2 and MrX-3 datasets reported elsewhere [7, 2]. The messages in the stream were filtered contemporaneously using LR in exactly the same active learning configuration as for CEAS, but with real rather than simulated user feedback. In addition, the messages (prior to filtering) were forwarded to an industry leading webmail system, with an embedded spam filter. While the

Filter	% fpr	% fnr	fnif
LR human in loop	4.24	0.04	20.2
DMC+Webmail	4.24	0.05	14.6
DMC	4.24	0.07	11.1
NB+Webmail	4.24	0.11	6.9
NB autonomous	4.24	0.14	5.3
Webmail builtin	4.24	0.81	1.0
LR autonomous	4.24	1.03	0.8

Table 2: False negative improvement factor relative to webmail builtin filter on MrX-5 Corpus. All factors > 1 are significant ($p \approx 0.000$).

webmail system has a (somewhat cumbersome) interface for user feedback, it was not used. A list of messages delivered to the inbox was captured from the system, and used to create labels indicating the webmail filter result. Finally, the email stream was filtered using the IT department’s SpamAssassin server. The result – a numerical “spamminess” score for each message – was captured.

Global filters. For the CEAS dataset, the global filter was a commercial desktop plugin implementation of a collaborative spam filter based on duplicate detection. The filter maintains a hashed digest of each reported spam message and a near-duplicate detection method determines whether each message has previously been reported as spam. Filter errors arise from two sources: errors in near-duplicate detection, and errors or omissions in the database of previously reported spam. Neither of these sources of error is obviously correlated with the content-based features employed by a personal filter. The error rates (0.6% false positives, 6.3% false negatives) are within the assumed range.

For MrX-5, two global filters were used. The webmail filter uses an undisclosed method; we expect that it relies heavily on network authentication, blacklists, and the like. The filter also supplies a user interface to mark email as spam or not, and is advertised as learning from this feedback. However, this interface was not used. The resulting error rates (4.2% false positives, 0.77% false negatives) are within the assumed range. The SpamAssassin filter (SA) uses a wide variety of information sources, including blacklists and near-duplicate detection, and also content-based patterns. SpamAssassin includes a “Bayesian” filter, which was not used. SpamAssassin’s error rates (with the default threshold setting of 5) of 0.25% fpr, 17% fnr are higher than those of the other filters, but still amenable for noise-tolerant methods.

User-trained personal filter. For both datasets, we used exactly the **wat1** filter that yielded best or near-best results at TREC 2007 [5]. This method examines only the first 3500 bytes of each message, treating each overlapping character 4-gram as a token. Each message is classified using logistic regression, and a single gradient descent step with learning rate 0.004 is taken for each feedback message. The filter was trained in an active learning scenario. For the CEAS dataset, the filter was allowed to request feedback on up to 1000 messages, but no more. For MrX-5, the filter first delivered messages classified as ham to the inbox, and then also delivered messages whose classification was uncertain to a different folder. The user perused this folder occasionally, identifying messages as ham or spam. 3843 messages (1.4%) were placed in the folder.

Filter	% fpr	% fnr	fnif
LR human in loop	0.25	0.65	25.8
NB+SpamAssassin	0.25	9.13	1.8
DMC+SpamAssassin	0.25	12.59	1.3
SpamAssassin	0.25	16.70	1.0
NB autonomous	0.25	18.86	0.9
DMC autonomous	0.25	23.20	0.7
LR autonomous	0.25	31.36	0.5

Table 3: False negative improvement factor relative to SpamAssassin global filter on MrX-5 Corpus. All factors > 1 are significant ($p \approx 0.000$).

Autonomous personal filters. We tested two discriminative and two generative filters previously shown to be noise tolerant [3, 17]. The **wat1** logistic regression filter was used, with learning rate reduced to 0.0005. A relaxed on-line support vector machine (RO SVM) [19] was used. This filter is derived from **tftS1F**, whose performance at TREC 2007 was on par with **wat1**. The only change was to decrease the regularization parameter C from 100 to 0.5, a modification known to increase noise tolerance.

The generative filters were Dynamic Markov Compression and our own adaptation of Graham and Robinson’s Naive Bayes method. The DMC implementation was identical to **wat2** which showed strong third-place results (after **wat1** and **tftS1F**) at TREC 2007.

The details of our naive Bayes method have not previously been reported, and are therefore presented here. We implemented it and tuned it some time ago in an effort to discover the essence of Graham and Robinson’s methods. Our naive Bayes implementation is in fact an ensemble formed by summing the results of two NB filters using different tokenization. The first member of the ensemble, NB-4G, uses overlapping 4-grams from the first 3500 bytes of the message, including headers. The second member, NB-W, uses the first 3000 “words”, where a word is defined as any sequence of alphabetic and numeric characters. For each message, the filter computes the log-odds-ratio for each token t , based on the number of times t has appeared in feedback messages labeled as spam ($t \in \text{spam}$) or in feedback messages labeled as ham ($t \in \text{ham}$):

$$\log OR_t = \log \left(\frac{|\{t \in \text{spam}\}|}{|\{t \notin \text{spam}\}|} \cdot \frac{|\{t \notin \text{ham}\}|}{|\{t \in \text{ham}\}|} \right).$$

The score for NB-4G is the mean of the largest and smallest k values of $\log OR_t$, where $k = 30$. Note that the k largest values are typically positive, indicating spam, while the k smallest are typically negative, indicating ham. The score for NB-W is derived by the same method with $k = 15$. The overall score for NB is the sum of these two scores.

Global-autonomous ensemble filter. Given prior results showing the effectiveness of spam filter fusion, we predicted that the fusion of global and autonomous personal filter results might well exceed the performance of either. To this end, it was necessary to derive a score from the global filter in order to combine it with the score from the autonomous personal filter. For the desktop and webmail filters, messages classified as spam were given a score of 1, while messages classified as ham were given a score of 0. For SpamAssassin, we added the score reported by SpamAssassin to the score from the autonomous personal filter.

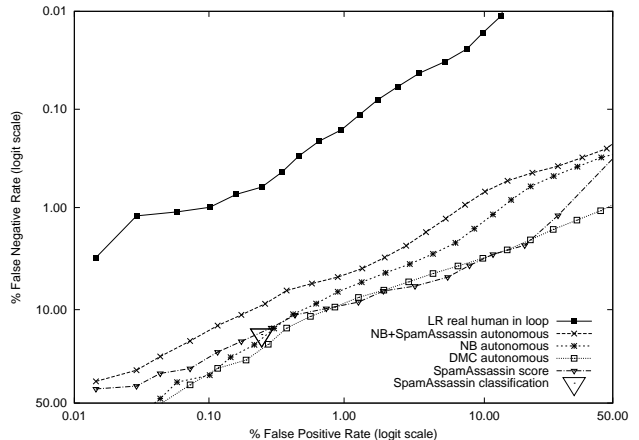


Figure 3: MrX-5 Corpus results, with SpamAssassin filter baseline. The large triangle represents the performance of SpamAssassin as configured and run by the recipient’s IT department. The curve with small triangles represents the score returned by SpamAssassin. The top line represents the performance of a personal filter, reproduced from figure 2. The other lines represent the performance of autonomous personal filters, trained exclusively on the output from the baseline filter. Although the individual autonomous filters do not appear to improve on the baseline, ensemble filters do.

4. EVALUATION

False positive rate (fpr) is the fraction of ham messages that are misclassified by the filter. *False negative rate* (fnr) is the fraction of spam messages that are misclassified. All the personal filters evaluated here return a score s rather than a categorical result; such a categorical result is easily derived by comparing s to some fixed threshold t . For each possible value of t the same filter yields a different pair (fnr_t, fpr_t) of false positive and false negative rates. Determining the “best” choice of t involves a somewhat imponderable tradeoff between fnr and fpr. Instead of evaluating with respect to some arbitrary t , we use receiver operating characteristic analysis to evaluate filter effectiveness. A receiver operating characteristic curve (ROC) is simply the set of all (fnr_t, fpr_t) achievable for some t . From the curve one may determine, for example, what fnr_t would be if t were set so that $fpr_t = x$, where x is some specific false positive rate. Or one may determine what fpr_t would be if t were set so that $fnr_t = y$.

The desktop and webmail filters report only a categorical result. Their effectiveness is therefore characterized by a single (fpr, fnr) pair which may be juxtaposed with the ROC curves for comparison. In general, a curve that lies above (fpr, fnr) indicates superior effectiveness. For quantitative comparison, we set t such that $fpr_t = fpr$ and compute the *false negative improvement factor* $fnif = \frac{fnr}{fnr_t}$. Informally, a filter with $fnif = k$ is “ k times better.” For $fnif > 1$ we estimate statistical significance by applying a sign test to the categorical result derived using t .

We could equally well have used *false positive improvement factor* $fpiif = \frac{fpr}{fpr_t}$, which fixes $fnr_t = fnr$. The use of one or the other does not imply that the value of fpr or

fnr is particularly apt; it is merely the value yielded by the global filter and hence the only available common frame of reference. fni_f and fpi_f are simply two measures to indicate the degree to which the ROC curve for the learning filter is superior to the point for the global filter. A superior curve will have both $fni_f > 1$ and $fpi_f > 1$; for our purpose the choice of measures is not critical, and we follow the convention of considering the y -axis (fnr_t) to be the dependent variable.

SpamAssassin returns both a categorical result and a score. The point representing the categorical result lies on the ROC curve representing the score. For consistency with the other comparisons, we report fni_f relative to this point.

5. RESULTS

Our first experiment used the CEAS corpus, with the commercial desktop filter as a baseline. As seen in figure 1 and table 1, the baseline filter achieves $fpr = 0.63\%$ and $fnr = 6.64\%$. While these error rates are perhaps higher than advertised, they are competitive with the best global filter results we have observed for the MrX corpora. The personal filter, with 1000 feedback messages, achieves (with suitable choice of t) the same fpr and $fnr_t = 0.04$, an improvement factor of $fni_f = 150$.

For our primary hypothesis, the filters of interest are those labeled “autonomous.” The ROC curves for all autonomous filters except LR are clearly superior to the baseline, though naive Bayes (NB) and the ensemble of naive Bayes and the baseline (NB+Desktop) are substantially better. For all autonomous filters $fni_f > 1$ ($p \approx 0.000$). However, it would be misleading to conclude that the LR result ($fni_f = 1.1$) represents a substantive improvement over the baseline. In fact, the LR curve very nearly contains the baseline point, indicating that it learned the nature of the baseline filter errors.

SVM ($fni_f = 1.6$) and DMC ($fni_f = 2.1$) show substantive improvements over the baseline, but this improvement is eclipsed by that of NB ($fni_f = 22$). The ensemble method NB+Desktop improves NB somewhat ($fni_f = 26$). The ensemble method DMC+Desktop (see table 1) dramatically improves DMC ($fni_f = 10$), but not to nearly the effectiveness of NB.

Our second experiment used the MrX-5 corpus, with the builtin webmail filter as a baseline. As seen in figure 2 and table 2, the baseline filter achieves $fpr = 4.2\%$ and $fnr = 0.8\%$. We were surprised by the high false positive rate, and hand-verified for a one month period that every false positive (some 30 messages) was in fact delivered to the spam folder. The personal filter, which was trained by the user in the normal course of reading email, achieved $fnr_t = 0.04$ ($fni_f = 20$). DMC ($fni_f = 11$) and NB ($fni_f = 5.3$) both improved on the baseline, but in this case DMC is superior. The ensemble method improves on both DMC ($fni_f = 15$) and NB ($fni_f = 6.9$). The LR curve once again falls near, but in this case slightly below, the baseline ($fni_f = 0.8$). We were unable to run SVM due to the size of the dataset.

Our third experiment also used the MrX-5 corpus, but with SpamAssassin as the baseline. As seen in figure 3 and table 3, SpamAssassin achieves $fpr = 0.25\%$ and $fnr = 17\%$. The same personal filter results from our second experiment yield $fnr_t = 0.65\%$ ($fni_f = 26$) relative to the SpamAssassin baseline. Performance of the autonomous fil-

ters is worse than for the previous two experiments, with all individual filters showing $fni_f < 1$. But the ensemble methods both improve on the baseline: NB+SpamAssassin ($fni_f = 1.8$), DMC+SpamAssassin ($fni_f = 1.3$). These improvements are significant $p \approx 0.000$ and substantive, though not as dramatic as for the previous experiments.

6. DISCUSSION

The results of our three experiments support the hypothesis that autonomous personal filters can improve on global systemwide filters. Generative filters like naive Bayes and DMC work well for this purpose, particularly in ensemble with the systemwide filter. Discriminative filters like logistic regression and SVM do not work so well, even with noise-tolerant parameter settings. This result is perhaps not surprising, as these filters should be able to isolate features that are highly correlated with the global filter’s errors, and therefore reproduce the errors. For example, the IP address of a blacklisted server would appear in the message header.

The results were least dramatic with respect to the SpamAssassin baseline filter. This result was predicted, as SpamAssassin relies heavily on content-based patterns which the autonomous filter can learn. But SpamAssassin also uses blacklists and collaborative filtering, which may be the source of apparent noise. SpamAssassin includes a Graham-Robinson “Bayes” filter which is (optionally) deployed in a manner similar to that described here. The filter is trained on the result of the SpamAssassin rules, but only when those rules yield an extreme score, and only when the rules disagree with the Bayes filter. We have previously evaluated the effect of the Bayes filter [6, 7] in this setting and found it to offer a small improvement. A topic for future research is to see if the approach presented here would yield a more substantial improvement. Our reasons for believing that it might are:

- Training on messages with extreme scores is the opposite of what is known to work well for personal filter training. In general, messages with intermediate scores yield the most information as training examples.
- The train-on-error regimen tends to make the Bayes filter discriminative rather than generative. Our results indicate that generative filters work better.
- The global filter that SpamAssassin uses includes content-based patterns as well as blacklists and collaborative filter. We posit that excluding the content-based patterns would yield better results.

It is impossible to avoid the conclusion that, notwithstanding the improvements in autonomous filtering demonstrated here, a filter harnessing user feedback vastly outperforms those that don’t. It may be that some other autonomous filter can do as well as one harnessing feedback, but to our knowledge none has been demonstrated in a controlled experiment. The amount of feedback involved in training the LR filter used here is not particularly onerous, and it may be argued that much larger effort and risk are associated with forgoing improvement factors of 20 or more. While the results presented here mitigate the difference between filters that harness feedback and those that do not, those that harness feedback remain the standard to beat.

7. REFERENCES

- [1] BRATKO, A., CORMACK, G. V., FILIPIČ, B., LYNAM, T. R., AND ZUPAN, B. Spam filtering using statistical data compression models. *Journal of Machine Learning Research* 7, Dec (2006), 2673–2698.
- [2] CORMACK, G. *Email Spam Filtering: A Systematic Review*. NOW Publishers, 2008.
- [3] CORMACK, G., AND KOLCZ, A. Spam filter evaluation with imprecise ground truth. In *32nd ACM SIGIR Conference on Research and Development on Information Retrieval* (Boston, 2009).
- [4] CORMACK, G. V. TREC 2007 Spam Track Overview. In *Sixteenth Text REtrieval Conference (TREC-2007)* (Gaithersburg, MD, 2007), NIST.
- [5] CORMACK, G. V. University of waterloo participation in the trec 2007 spam track. In *Sixteenth Text REtrieval Conference (TREC-2007)* (Gaithersburg, MD, 2007), NIST.
- [6] CORMACK, G. V., AND LYNAM, T. R. TREC 2005 Spam Track overview. In *Proc. 14th Text REtrieval Conference (TREC 2005)* (Gaithersburg, MD, November 2005).
- [7] CORMACK, G. V., AND LYNAM, T. R. On-line supervised spam filter evaluation. *ACM Transactions on Information Systems* 25, 3 (2007).
- [8] GOODMAN, J., AND TAU YIH, W. Online discriminative spam filter training. In *The Third Conference on Email and Anti-Spam* (Mountain View, CA, 2006).
- [9] GRAHAM, P. Better bayesian filtering. <http://www.paulgraham.com/better.html>, 2004.
- [10] GRAHAM-CUMMING, J. SpamOrHam. *Virus Bulletin* (2006-06-01).
- [11] LYNAM, T. R., AND CORMACK, G. V. On-line spam filter fusion. In *29th ACM SIGIR Conference on Research and Development on Information Retrieval* (Seattle, 2006).
- [12] MEYER, T. A. A TREC along the spam track with SpamBayes. In *Proc. 14th Text REtrieval Conference (TREC 2005)* (Gaithersburg, MD, November 2005).
- [13] ROBINSON, G. A statistical approach to the spam problem. *Linux Journal* 107 (March 2003), 3.
- [14] SAHAMI, M., DUMAIS, S., HECKERMAN, D., AND HORVITZ, E. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop* (Madison, Wisconsin, 1998), AAAI Technical Report WS-98-05.
- [15] SCULLEY, D. Online active learning methods for fast label-efficient spam filtering. In *Proc. CEAS 2007 – Fourth Conference on Email and Anti-Spam* (Mountain View, CA, 2007).
- [16] SCULLEY, D. Practical learning from one-sided feedback. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (2007), ACM New York, NY, USA, pp. 609–618.
- [17] SCULLEY, D., AND CORMACK, G. V. Filtering spam in the presence of noisy user feedback. *Tufts University* (2008).
- [18] SCULLEY, D., AND CORMACK, G. V. Filtering spam in the presence of noisy user feedback. In *Proceedings of the 5th Conference on Email and Anti-Spam (CEAS 2008)* (2008).
- [19] SCULLEY, D., AND WACHMAN, G. M. Relaxed online support vector machines for spam filtering. In *30th ACM SIGIR Conference on Research and Development on Information Retrieval* (Amsterdam, 2007).
- [20] SCULLEY, D., AND WACHMAN, G. M. Relaxed online SVMs in the TREC Spam Filtering Track. In *Sixteenth Text REtrieval Conference (TREC-2007)* (Gaithersburg, MD, 2007), NIST.
- [21] TAU YIH, W., MCCANN, R., AND KOLCZ, A. Improving spam filtering by detecting gray mail. In *Proc. CEAS 2007 – Fourth Conference on Email and Anti-Spam* (Mountain View, CA, 2007).