TREC 2007 Spam Track Overview

Gordon Cormack
University of Waterloo
Waterloo, Ontario, Canada

1 Introduction

TREC's *Spam Track* uses a standard testing framework that presents a set of chronologically ordered email messages a spam filter for classification. In the filtering task, the messages are presented one at at time to the filter, which yields a binary judgement (*spam* or *ham* [i.e. non-spam]) which is compared to a human-adjudicated *gold standard.* The filter also yields a *spamminess* score, intended to reflect the likelihood that the classified message is spam, which is the subject of post-hoc ROC (Receiver Operating Characteristic) analysis. Four different forms of user feedback are modeled: with *immediate feedback* the gold standard for each message is communicated to the filter immediately following classification; with *delayed feedback* the gold standard is communicated to the filter sometime later (or potentially never), so as to model a user reading email from time to time and perhaps not diligently reporting the filter's errors; with *partial feedback* the gold standard for only a subset of email recipients is transmitted to the filter, so as to model the case of some users never reporting filter errors; with *active on-line learning* (suggested by D. Sculley from Tufts University [5]) the filter is allowed to request immediate feedback for a certain quota of messages which is considerably smaller than the total number. Two test corpora – email messages plus gold standard judgements – were used to evaluate subject filters. One *public* corpus (trec07p) was distributed to participants, who ran their filters on the corpora using a track-supplied toolkit implementing the framework and the four kinds of feedback. One private *corporus* (MrX 3) was not distributed to participants; rather, participants submitted filter implementations that were run, using the toolkit, on the private data. Twelve groups participated in the track, each submitting up to four filters for evaluation in each of the four feedback modes (immediate; delayed; partial; active).

Task guidelines and tools may be found on the web at http://plg.uwaterloo.ca/~gvcormac/spam/ .

1.1 Filtering – Immediate Feedback

The immediate feedback filtering task is identical to the TREC 2005 and TREC 2006 (immediate) tasks [1, 3]. A chronological sequence of messages is presented to the filter using a standard interface. The the filter classifies each message in turn as either *spam* or *ham,* also computes a *spamminess score* indicating its confidence that the message is spam. The test setup simulates an ideal user who communicates the correct (gold standard) classification to the filter for each message immediately after the filter classifies it.

Participants were supplied with tools, sample filters, and sample corpora (including the TREC 2005 and TREC 2006 public corpora) for training and development. Filters were evaluated on the two new corpora developed for TREC 2007.

1.2 Filtering – Delayed Feedback

Real user's don't immediately report the correct classification to filters. They read their email some time, typically in batches, some time after it is classified. Last year (TREC 2006) the delayed learning task sought to simulate user behaviour by withholding feedback for some random number of messages after which feedback was given; this delay followed by feedback was repeated in several cycles. This year (TREC 2007) the track seeks instead to measure the effect of delay. To this end, immediate feedback is given for the first several thousand messages (10,000 for trec07p; 20,000 for MrX 3) after which no feedback at all is given. Thus, the majority of the corpus is classified with no feedback and the cumulative effect of delay may be evaluated by examining the learning curve.

Participants trained on the TREC 2006 corpus. While the 2007 guidelines specified that feedback might never be given, they did not specify the exact nature of the task. It was anticipated that the delayed feedback task would be more difficult for the filters, and that filter performance would degrade during the interval for which no feedback was given. It was anticipated that participants might be able to harness information from unlabeled messages (the ones for which feedback was not given) to improve performance.

1.3 Partial Feedback

Partial feedback is a variant on delayed feedback effected with exactly the same tools. As for "delayed feedback" the feedback was in fact either given immediately or not at all. In this case, however, the messages for which feedback was given were those sent to a subset of the recipients in the corpus; that is, the filter was trained on some users' messages but asked to classify every users' messages. Partial feedback was used only for the trec07p corpus, as it contained email addressed to many recipients. It was not applicable to MrX 3, being a single-user corpus.

1.4 The On-line Active Learning Task

For the on-line task, filters were passed an additional parameter – the quota of messages for which feedback could be requested – and were expected to return an additional result – to request or decline feedback for each message classified. Filters that were unaware of these parameters were assumed to request feedback for each message classified until the quota was exhausted; thus the default behaviour was identical to the delayed feedback task. However, filters were able to decline feedback for some messages (presumably those whose classification the filter found unimportant) in order to preserve quota so as to be able to request feedback for later messages.

A naive solution to this problem would be to have the filter make a label request for every message. This would request labels and train normally for the first N messages, where N is the initial quota, and then would not update for the remainder of the run. The testing jig is backward compatible with filters from prior years by making the naive approach the default method if no label request is specified. This allows prior filters to run on this task without modification.

2 Evaluation Measures

We used the same evaluation measures developed for TREC 2005. The tables and figures in this overview report Receiver Operating Characteristic (ROC) Curves, as well as $1 - ROCA(\%)$ – the area above the ROC curve, indicating the probability that a random spam message will receive a lower spamminess score than a random ham message.

The appendix contains detailed summary reports for each participant run, including ROC curves, 1-ROCA%, and the following statistics. The *ham misclassification percentage* (*hm%*) is the fraction of all ham classified as spam; the *spam misclassification percentage* (*sm%*) is the fraction of all spam classified as ham.

There is a natural tension between ham and spam misclassification percentages. A filter may improve one at the expense of the other. Most filters, either internally or externally, compute a spamminess score that reflects the filter's estimate of the likelihood that a message is spam. This score is compared against some fixed threshold t to determine the ham/spam classification. Increasing t reduces *hm%* while increasing *sm%* and vice versa. Given the score for each message, it is possible to compute *sm%* as a function of *hm%* (that is, *sm%* when t is adjusted to as to achieve a specific *hm%*) or vice versa. The graphical representation of this function is a Receiver Operating Characteristic (ROC) curve; alternatively a recall-fallout curve. The area under the ROC curve is a cumulative measure of the effectiveness of the filter over all possible values. ROC area also has a probabilistic interpretation: the probability that a random ham will receive a lower score than a random spam. For consistency with *hm%* and *sm%*, which measure failure rather than effectiveness, spam track reports the area *above* the ROC curve, as a percentage ($(1 - ROCA)\%$). The appendix further reports *sm%* when the threshold is adjusted to achieve several specific levels of *hm%,* and vice versa.

A single quality measure, based only on the filter's binary ham/spam classifications, is nonetheless desirable. To this end, the appendix reports *logistic average misclassification percentage* (*lam%*) defined as $lam\% = logit^{-1}(\frac{logit(hm\%)+logit(sm\%)}{2})$ where $logit(x) = log(\frac{x}{100\%-x})$. That is, *lam%* is the geometric mean of the

odds of ham and spam misclassification, converted back to a proportion[1]. This measure imposes no a priori relative importance on ham or spam misclassification, and rewards equally a fixed-factor improvement in the odds of either.

For each measure and each corpus, the appendix reports 95% confidence limits computed using a bootstrap method [2] under the assumption that the test corpus was randomly selected from some source population with the same characteristics.

3 Spam Filter Evaluation Tool Kit

All filter evaluations were performed using the *TREC Spam Filter Evaluation Toolkit*, developed for this purpose. The toolkit is free software and is readily portable.

Participants were required to provide filter implementations for Linux or Windows implementing five command-line operations mandated by the toolkit:

- **initialize** – creates any files or servers necessary for the operation of the filter
- **classify** *message [quota]* – returns ham/spam classification and spamminess score for *message. [quota]* is used only in active learning feedback.
- **train ham** *message* – informs filter of correct (ham) classification for previously classified *message*
- **train spam** *message* – informs filter of correct (spam) classification for previously classified *message*
- **finalize** – removes any files or servers created by the filter.

Track guidelines prohibited filters from using network resources, and constrained temporary disk storage (1 GB), RAM (1 GB), and run-time (2 sec/message, amortized). These limits were enforced incrementally, so that individual long-running filters were granted more than 2 seconds provided the overall average time was less than 2 second per query plus one minute to facilitate start-up.

The toolkit takes as input a test corpus consisting of a set of email messages, one per file, and an index file indicating the chronological sequence and gold-standard judgements for the messages. It calls on the filter to classify each message in turn, records the result, and at some time later (perhaps immediately, perhaps never, and perhaps only on request of the filter) communicates the gold standard judgement to the filter.

The recorded results are post-processed by an evaluation component supplied with the toolkit. This component computes statistics, confidence intervals, and graphs summarizing the filter's performance.

4 Test Corpora

	Ham	Spam	Total
trec07p	25220	50199	75419
MrX3	8082	153893	161975
Total	33302	204092	237394

Table 1: Corpus Statistics

For TREC 2006, we used one public corpus and one private corpus with a total of 237,394 messages (see table 1).

4.1 Public Corpus – trec07p

The public corpus contains all the messages delivered to a particular server from April 8 through July 6, 2007. The server contains many accounts that have fallen into disuse but continue to receive a lot of spam. To these accounts were added a number of "honeypot" accounts published on the web and used to sign up for

[1]For small values, odds and proportion are essentially equal. Therefore *lam%* shares much with the geometric mean average precision used in the robust track.

a number of services – some legitimate and some not. Several services were canceled and several "opt-out" links from spam messages were clicked. All messages were adjudicated using the methodology developed for previous spam tracks. [4] This corpus is the first TREC public corpus that contains exclusively ham and spam sent to the same server within the same time period. The messages were unaltered except for a few systematic substitutions of names.

4.2 Private Corpus – MrX3

The MrX3 corpus was derived from the same source as the MrX and MrX2 corpora used for TREC 2006 and TREC 2006 respectively. All of X's email from December 2006 through July 11, 2007 was used. The proportion of spam has grown substantially since 2005[2]; Ham volume was insubstantially different.

5 Spam Track Participation

Group	Filter Prefix
Beijing University of Posts and Telecommunications	kid
Fudan University-WIM Lab	fdw
Heilongjiang Institute of Technology	hit
Indiana University	iub
International Institute of Information Technology	III
Jozef Stefan Institute	ijs
Mitsubishi Electric Research Labs	crm
National University of Defense Technology	ndt
Shanghai Jiao Tong University	sjt
South China University of Technology	scu
Tufts University	tft
University of Waterloo	wat

Table 2: Participant filters

Corpus / Task	Filter Suffix
trec07p / immediate feedback	pf
trec07p / delayed feedback	pd
trec07p / partial feedback	pp
trec07p / active feedback	p1000
MrX3 / immediate feedback	x3f
MrX3 /delayed feedback	x3d

Table 3: Run-id suffixes

Twelve groups participated in the TREC 2007 spam track. Each participant submitted up to four filter implementations for evaluation on the private corpora; in addition, each participant ran the same filters on the public corpora, which were made available following filter submission. All test runs are labelled with an identifier whose prefix indicates the group and filter priority and whose suffix indicates the corpus to which the filter is applied. Table 2 shows the identifier prefix for each submitted filter. All test runs have a suffix indicating the corpus and task, detailed in figure 3 .

6 Results

Figures 2 through 6 show the results of the best seven systems for each type of feedback with respect to each corpus. The left panel of each figure shows the ROC curve, while the right panel shows the learning curve: cumulative 1-ROCA% as a function of the number of messages processed. Only the best run for each

[2]Note that the MrX and MrX3 corpora include all email delivered during a particular time period, MrX2 was sampled so as to yield the same ham:spam ratio as MrX.

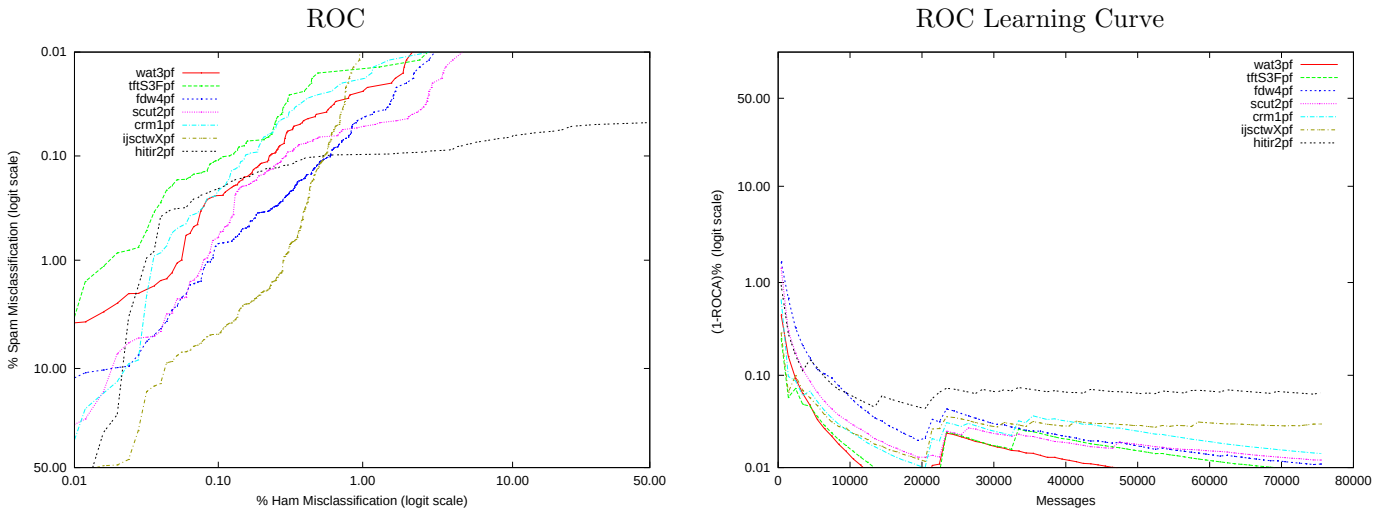

Figure 1: trec07p Public Corpus – Immediate Feedback

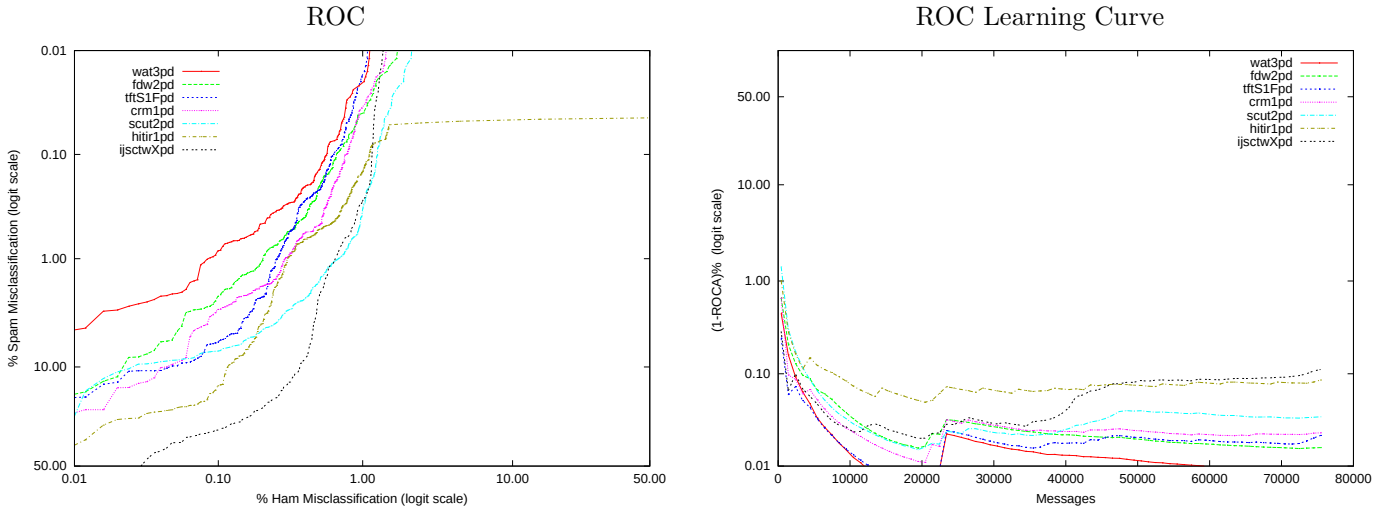

Figure 2: trec07p Public Corpus – Delayed Feedback

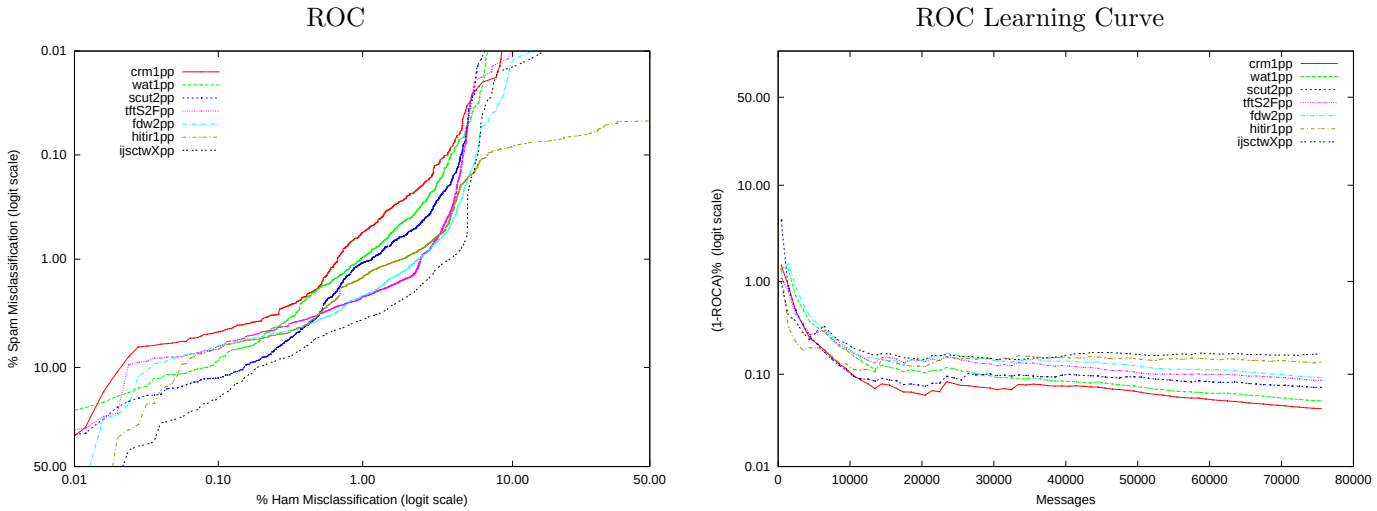

Figure 3: trec07p Public Corpus – Partial Feedback

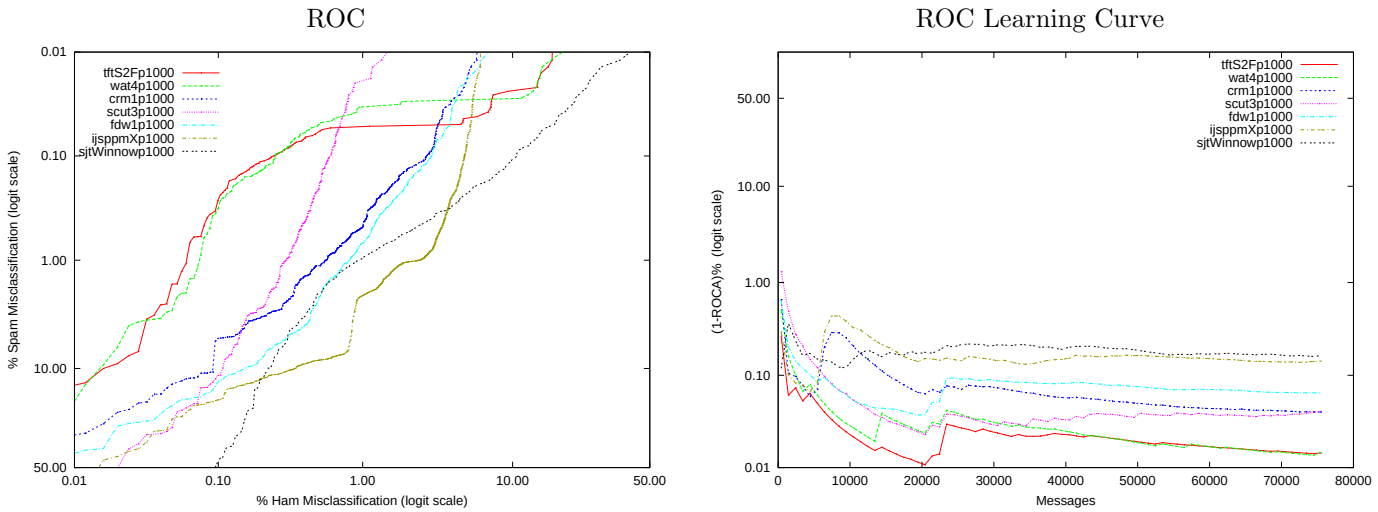

Figure 4: trec07p Public Corpus – Active Learning (quota 1000)

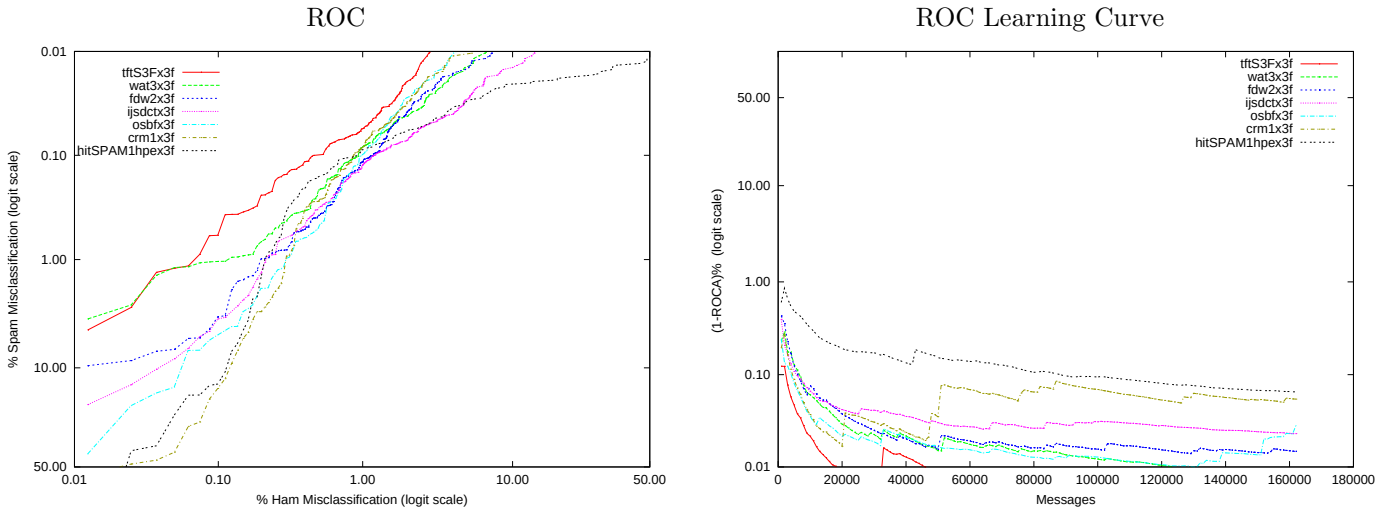

Figure 5: MrX3 Corpus – Immediate Feedback

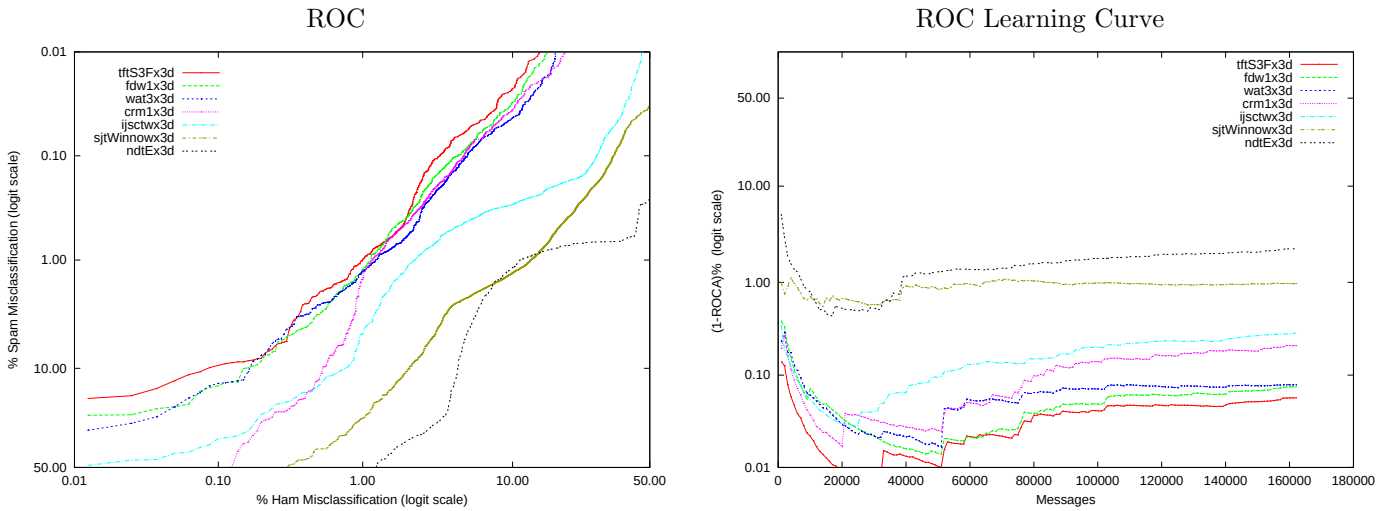

Figure 6: MrX3 Corpus – Delayed Feedback

	Immediate feed.		Delayed feed.		Partial feed.		Active learning	
Rank	run tag	1-ROCA(%)	run tag	1-ROCA(%)	run tag	1-ROCA(%)	run tag	1-ROCA(%)
1	wat3pf	0.0055	wat3pd	0.0086	crm1pp	0.0425	tftS2Fp1000	0.0144
2	wat1pf	0.0057	wat1pd	0.0105	wat1pp	0.0514	wat4p1000	0.0145
3	wat4pf	0.0057	wat4pd	0.0105	wat4pp	0.0514	crm1p1000	0.0401
4	wat2pf	0.0077	fdw2pd	0.0159	wat3pp	0.0516	scut3p1000	0.0406
5	tftS3Fpf	0.0093	wat2pd	0.0207	scut2pp	0.0719	tftS1Fp1000	0.0413
6	tftS1Fpf	0.0099	tftS1Fpd	0.0214	tftS2Fpp	0.0858	tftS3Fp1000	0.0475
7	tftS2Fpf	0.0103	fdw1pd	0.0223	tftS1Fpp	0.0878	scut2p1000	0.0533
8	fdw4pf	0.0109	tftS2Fpd	0.0225	tftS3Fpp	0.0919	fdw1p1000	0.0641
9	scut2pf	0.0121	tftS3Fpd	0.0226	fdw2pp	0.0921	fdw2p1000	0.0881
10	crm1pf	0.0142	crm1pd	0.0229	fdw1pp	0.1066	wat1p1000	0.1193
11	fdw3pf	0.0157	fdw3pd	0.0229	wat2pp	0.1087	wat2p1000	0.1193
12	fdw2pf	0.0195	fdw4pd	0.0229	fdw3pp	0.1109	wat3p1000	0.1215
13	fdw1pf	0.0198	scut2pd	0.0342	fdw4pp	0.1151	ijsppmXp1000	0.1417
14	ijsctwXpf	0.0297	scut3pd	0.0516	hitir1pp	0.1351	ijsctwXp1000	0.1473
15	ijsppmXpf	0.0299	hitir1pd	0.0855	hitir2pp	0.1356	sjtWinnowp1000	0.1626
16	scut1pf	0.0348	hitir2pd	0.0876	scut1pp	0.1534	fdw3p1000	0.1629
17	ijsdcwXpf	0.0371	ijsctwXpd	0.1111	ijsctwXpp	0.1656	scut1p1000	0.1939
18	ijsdctXpf	0.0382	ijsppmXpd	0.1148	ijsppmXpp	0.1724	fdw4p1000	0.2029
19	scut3pf	0.0406	sjtWinnowpd	0.2813	crm4pp	0.1866	hitir2p1000	0.2800
20	crm4pf	0.0457	crm2pd	0.3186	scut3pp	0.1898	crm2p1000	0.3244
21	hitir2pf	0.0644	scut1pd	0.3251	ijsdctXpp	0.1962	hitir1p1000	0.3246
22	hitir1pf	0.0652	crm4pd	0.3354	ijsdcwXpp	0.2477	ndtAp1000	0.7507
23	sjtMulti1pf	0.0709	sjtMulti1pd	0.4250	crm2pp	0.3882	ndtBp1000	1.3037
24	sjtMulti2pf	0.0732	ndtApd	0.4359	sjtMulti1pp	0.4250	sjtMulti1p1000	1.3102
25	IIITHpf	0.1041	ndtBpd	0.5842	sjtMulti2pp	0.4830	ndtCp1000	1.3932
26	crm2pf	0.1289	ndtCpd	0.6547	crm3pp	0.6743	kidult2p1000	1.5239
27	ndtApf	0.1662	crm3pd	0.8844	sjtBayespp	0.6910	kidult3p1000	1.5895
28	ndtBpf	0.1931	kidult3pd	0.9006	ndtApp	0.7910	kidult1p1000	1.6267
29	ndtCpf	0.2164	kidult0pd	1.1703	ndtBpp	0.9366	kidult0p1000	1.9030
30	sjtWinnowpf	0.2209	kidult2pd	1.4355	sjtWinnowpp	1.0133	ndtDp1000	2.3704
31	crm3pf	0.2364	kidult1pd	1.4959	ndtCpp	1.0191	sjtMulti2p1000	2.6864
32	sjtBayespf	0.3155	iube5c5pd	1.5241	kidult3pp	3.1509	sjtBayesp1000	4.0136
33	kidult0pf	0.3599	iube2c3pd	1.5911	kidult1pp	3.1711	iube2c3p1000	10.3933
34	kidult3pf	0.4515	iube2c6pd	1.9411	kidult2pp	3.1940	iube5c5p1000	10.3933
35	kidult2pf	0.4532	ndtDpd	1.9486	kidult0pp	3.5517	iube2c6p1000	12.5153
36	kidult1pf	0.4579	sjtMulti2pd	17.2297	iube5c5pp	4.0446	crm4p1000	50.3043

Table 4: Summary 1-ROCA (%) – trec07p Public Corpus

participant is shown in the figures; tables 4 and 5 show 1-ROCA% for all feedback regimens on trec07p and MrX3 respectively. Full details for all runs are given in the notebook appendix.

7 Conclusions

Once again, the general performance of filters has improved over previous techniques. Support vector machines and logistic regression, specifically engineered for spam filtering, show exceptionally strong performance. Delayed and partial feedback degrade filter performance; at the time of writing we are unaware of any special methods used by participants mitigate this degradation. The learning curves do not show substantial de-learning as delay increases.

The best-performing techniques for active learning use techniques akin to "uncertainty scheduling" in which feedback is requested only for those messages whose score is near the filter's threshold.

	Immediate feed.		Delayed feed.	
Rank	run tag	1-ROCA(%)	run tag	1-ROCA(%)
1	tftS3Fx3f	0.0042	tftS3Fx3d	0.0568
2	tftS2Fx3f	0.0054	tftS2Fx3d	0.0683
3	wat3x3f	0.0076	tftS1Fx3d	0.0685
4	wat1x3f	0.0096	fdw1x3d	0.0747
5	wat4x3f	0.0096	fdw2x3d	0.0751
6	fdw2x3f	0.0147	wat3x3d	0.0787
7	fdw3x3f	0.0154	wat1x3d	0.0896
8	fdw1x3f	0.0155	fdw3x3d	0.1062
9	tftS1Fx3f	0.0166	fdw4x3d	0.1258
10	wat2x3f	0.0219	crm1x3d	0.2079
11	ijsdctx3f	0.0229	wat2x3d	0.2512
12	fdw4x3f	0.0255	ijsctwx3d	0.2830
13	ijsdcwx3f	0.0281	ijsppmx3d	0.3055
14	osbfx3f	0.0281	crm2x3d	0.3811
15	ijsctwx3f	0.0392	ijsdcwx3d	0.5036
16	ijsppmx3f	0.0397	ijsdctx3d	0.5288
17	crm1x3f	0.0543	crm4x3d	0.7589
18	hitSPAM1hpex3f	0.0650	sjtWinnowx3d	0.9674
19	hitSPAM2chix3f	0.1032	ndtEx3d	2.2840
20	crm4x3f	0.1145	crm3x3d	2.5169
21	crm2x3f	0.1296	kid0x3d	2.5383
22	sjtWinnowx3f	0.1666	ndtDx3d	4.6920
23	sjtMulti1x3f	0.3413	sjtMulti1x3d	5.0656
24	crm3x3f	0.9476	ndtAx3d	5.3401
25	IIITx3f	1.0234	sjtBayesx3d	28.7693
26	kidult0x3f	1.0313	IIITx3d	49.9682
27	ndtDx3f	1.3985	-	-
28	sjtBayesx3f	2.0811	-	-
29	ndtAx3f	2.4078	-	-
30	scut2x3f	4.7596	-	-
31	iube5c6x3f	19.0336	-	-
32	hitSPAM3bayx3f	49.9682	-	-

Table 5: Summary 1-ROCA (%) – MrX3 Private Corpus

8 Acknowledgements

The author thanks D. Sculley for suggesting the active feedback task and making the necessary modifications to the spam filter evaluation toolkit.

References

[1] Cormack, G. Trec 2005 spam track overview. In *Proceedings of TREC 2005* (Gaithersburg, MD, 2005).

[2] Cormack, G. Statistical precision of information retrieval evaluation. In *Proceedings of SIGIR 2006* (Seattle, WA, 2006).

[3] Cormack, G. Trec 2006 spam track overview. In *Proceedings of TREC 2006* (Gaithersburg, MD, 2006).

[4] Cormack, G., and Lynam, T. Spam Corpus Creation for TREC. In *Proceedings of the Second Conference on Email and Anti-Spam (CEAS-2005)* (Mountain View, CA, 2005).

[5] Sculley, D. Online active learning methods for fast label-efficient spam filtering.