TREC 2006 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. Two 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, so as to model a user reading email from time to time in batches. A new task – *active* learning – presents the filter with a large collection of unadjudicated messages, and has the filter request adjudication for a subset of them before classifying a set of future messages. Four test corpora – email messages plus gold standard judgements – were used to evaluate subject filters. Two of the corpora (the *public* corpora, one English and one Chinese) were distributed to participants, who ran their filters on the corpora using a track-supplied toolkit implementing the framework. Two of the corpora (the *private* corpora) were not distributed to participants; rather, participants submitted filter implementations that were run, using the toolkit, on the private data. Nine groups participated in the track, each submitting up to four filters for evaluation in each of the three tasks (filtering with immediate feedback; filtering with delayed feedback; active learning).

2 Spam Track Tasks

Broadly speaking, there were two spam track tasks: filtering, in which participant filters performed on-line classification with simulated user feedback; and active learning, in which participant filters were given a large number of historical email messages and allowed to request adjudication by the user for some of them before classifying a set of new messages. Task guidelines and tools may be found on the web at http://plg.uwaterloo.ca/~gvcormac/spam/ .

2.1 Filtering – Immediate Feedback

The immediate feedback filtering task is identical to the TREC 2005 task[1]. 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 public corpus) for training and development. Filters were evaluated on four new corpora developed for TREC 2006.

2.2 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. The delayed filtering task simulates this delay in the following manner: the filter is asked to classify some number of messages without feedback; after these messages are classified, feedback is given, in the same order the messages are classified, using the same

standard interface as for the filtering task. The only apparent difference to the filter is that each classification request is not immediately followed by training for the classified message.

The exact sequence of classification requests and feedback is determined by a special index file supplied with the corpus. The delay intervals were selected at random using an exponential distribution with a mean corresponding to several day's delay – from 500 to 1000 messages, depending on the corpus. While the intervals were randomly generated, there was no variation in the presentation of feedback to the various filters; each filter saw exactly the same sequence.

Tools for training and development were supplied to participants in advance; index files specifying feedback delay were supplied for the the training corpora. Filters were evaluated on the same four corpora used for immediate feedback, augmented by index files with randomly generated delay.

It was anticipated that the delayed feedback task would be more difficult for the filters, and that filters might be able to harness information from unlabeled messages (one for which feedback had not yet occurred) to improve performance.

The track coordinator considered, in addition, the use of incomplete feedback in which the true classification for some messages was never communicated to the filter. While this scenario more closely models that of real filter deployment, we argue that this situation is aptly modeled by the task as deployed. Since the filter is always trained on past data and asked to classify future data, using incomplete judgements would simply be equivalent to using a smaller corpus of training data. Resource constraints limited the number of corpora we were able to use, and it was decided that the largest possible corpora would yield the highest statistical power.

2.3 The Active Learning Task

The active learning task models the situation in which a spam filter is first deployed. We assume that many historical email messages are available, but that these messages have not been adjudicated as ham or spam. The filter examines these messages as a batch (although it knows their chronology) and asks the user (or administrator) to adjudicate several before being deployed to filter new messages.

For the active learning task each corpus was divided chronologically in the ratio $9:1$. The (chronologically) first 90% of the messages were given to the filter without classification, while the last 90% were held back for testing. For $n = 100, 200, 400, 800, ...$ filters were allowed to query the true classification of n messages selected by the filter. Based on the results of these queries, the filters were then required to classify the test messages in sequence.

The learning task was effected by a *shell* program, written in C++, which read the entire corpus index (including gold standard judgements) and simulated n queries by examining the index. Filter classification and training were effected using the same interface as for the filtering tasks, but this interface was between the shell program and the participant filter, both of which were under control of the participant. A single run of the shell program was used to simulate, in succession, all values of $n \leq m$ where m is the number of messages in the corpus.

A standard shell which selected n messages at random was supplied as a baseline; participants were invited to modify the shell to use a better strategy, while adhering to the constraint that the labels for at most n messages should be used in classification. Training and development was effected on the same training corpora as for the filtering tasks; the same evaluation corpora were used as well.

3 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.

4 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* – returns ham/spam classification and spamminess score for *message*
- **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) communicates the gold standard judgement to the filter.

Participants were supplied as well, with an active learning shell, *active.cpp*, which they modified to effect the active learning task.

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.

5 Test Corpora

For TREC 2006, we used two public corpora, one English and one Chinese, as well as two private corpora derived from the same users' email as the TREC 2005 private corpora.

[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.

5.1 Public English Corpus – trec06p

Public Corpora			
	Ham	Spam	Total
trec06p	12910	24912	37822
trec06c	21766	42854	64620
Total	34677	67766	102442

Private Corpora			
	Ham	Spam	Total
MrX2	9039	40135	49174
SB2	9274	2695	11969
Total	18313	42830	61143

Table 1: Corpus Statistics

14000 ham messages were crawled from the web. Only messages with complete "Received from" headers were selected; the messages were ordered by the time and date on the first such header. These messages were adjudicated by human judges assisted by several spam filters – DMC [3] , Bogofilter and SpamProbe – using the methodology developed for TREC 21005. About 1000 spam messages were discovered in this set; 12910 were ham.

The 14000 crawled messages were augmented by approximately 38000 spam messages collected in May 2006. Each spam message was paired with a ham message. The header of the spam message was altered to make it appear to have been addressed to the same recipient and delivered to the same mail server during the same time frame as the ham message. "To:" and "From:" headers, as well as the message bodies, were altered to substitute names and email addresses consistent with the addressee. SpamProbe and Bogofilter were run on the corpora, and their dictionaries examined, to identify artifacts that might identify these messages. A handful were detected and removed; for example, incorrect uses of daylight saving time, and incorrect versions of server software in header information. The DMC spam filter was run on the corpus several times and disagreements between the filter and the recorded judgement were adjudicated.

5.2 Public Chinese Corpus - trec06c

The Public Chinese corpus used data provided by the CERNET Computer Emergency Response Team (CCERT) at Tsinghuaua University, Beijing. The ham messages consisted of those send by to a mailing list; the spam messages were those sent to a spam trap in the same internet domain. Headers and bodies of spam messages were modified to make them appear to have been delivered to the same servers as the ham messages, in the same time interval. Both the ham and spam messages were modified to as to remove structural elements not in common with those of the other class, such as embedded signature files, certain kinds of HTML markup, and the like.

Pilot filtering using DMC revealed that the Chinese corpus was quite easy to classify; it was felt nevertheless that the corpus would reveal any western bias in filtering strategies.

5.3 Private Corpus – MrX2

The MrX2 corpus was derived from the same source as the MrX corpus used for TREC 2005. For comparability with MrX, a random subset of X's email from October 2005 through April 2006 was selected so as to yield the same corpus size and ham/spam ratio as for MrX. This selection involved primarily the elimination of spam messages, whose volume had increased about 50% since the 2003-2004 interval in which the original MrX corpus was collection. Ham volume was insubstantially different.

5.4 Private Corpus – SB2

The SB2 corpus was collected from the same source as last year's SB corpus. Spam volume tripled since last year; all delivered messages were used in the corpus.

5.5 Aggregate Pseudo-Corpus

The subject filters were run separately on the various corpora. That is, each filter was subject to eight test runs – four with immediate feedback and four with delayed feedback. For each filter and each type of feedback, an *aggregate run* was created combining its results on the four corpora as if they were one. The evaluation component of the toolkit was run on the aggregate results, consisting of 163,641 messages

for each type of feedback – 52,989 spam and 110,652 ham. The summary results on the aggregate runs provide a composite view of the performance on all corpora, but are not the results of running the filter on an aggregate corpus; hence we dub the aggregate a pseudo-corpus.

6 Spam Track Participation

Group	Filter Prefix
Beijing University of Posts and Telecommunications	bpt
Harbin Institute of Technology	hit
Humboldt University Berlin & Strato AG	hub
Tufts University	tuf
Dalhousie University	dal
Jozef Stefan Institute	ijs
Tony Meyer	tam
Mitsubishi Electric Research Labs (CRM114)	CRM
Fidelis Assis	ofl

Table 2: Participant filters

Corpus / Task	Filter Suffix
trec06p / immediate feedback	pei
trec06p / delayed feedback	ped
trec06c / immediate feedback	pci
trec06c / delayed feedback	pcd
MrX2 / immediate feedback	x2
MrX2 /delayed feedback	x2d
SB2 / immediate feedback	b2
SB2 / delayed feedback	b2d

Table 3: Run-id suffixes

Nine groups participated in the TREC 2006 filtering tasks; five of them also participated in the active learning task. For each task, 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 follwoing 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 .

7 Results

Figures 2 through 10 show the results for the filtering runs – immediate and delayed feedback – on the four corpora and on the aggregate pseudo-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 participant is shown in the figures; table 13 shows 1-ROCA% for all filter runs on all corpora. Full details for all runs are given in the notebook appendix.

Figures 11 through 14 show the performance of the active learning filters as a function of n – the number of training messages. Only the best run from each participant is shown. Full details are given in the notebook appendix.

8 Conclusions

Although the Chinese corpus was much easier than the others, and SB2 was harder, results are generally consistent. With a few exceptions, performance on the delayed feedback task was inferior to that of the

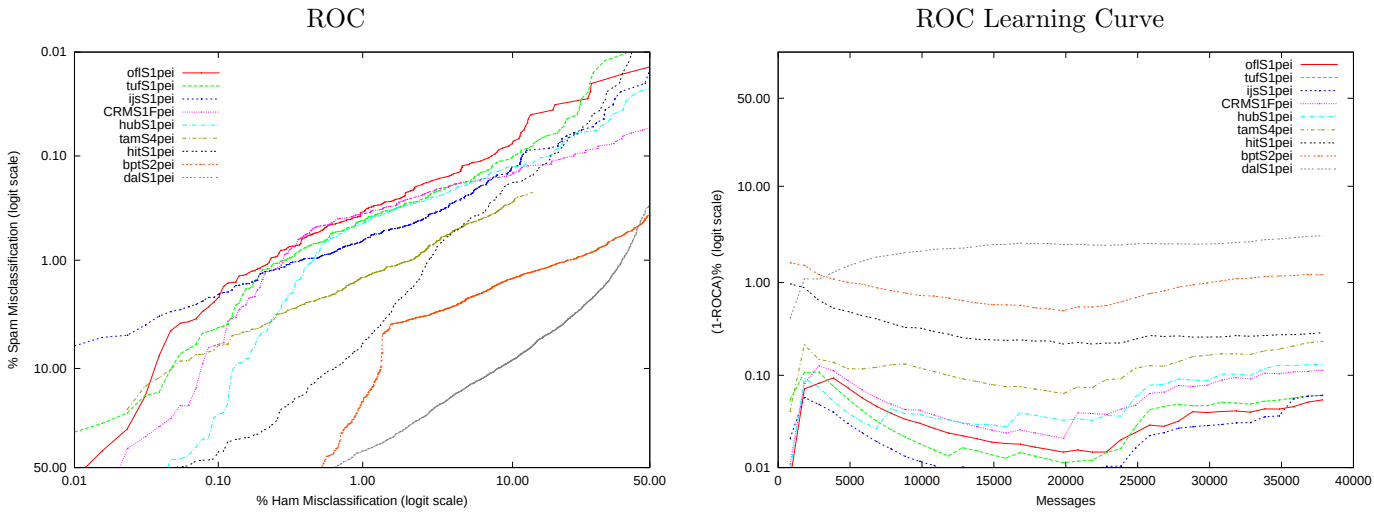

Figure 1: trec06p Public Corpus – Immediate Feedback

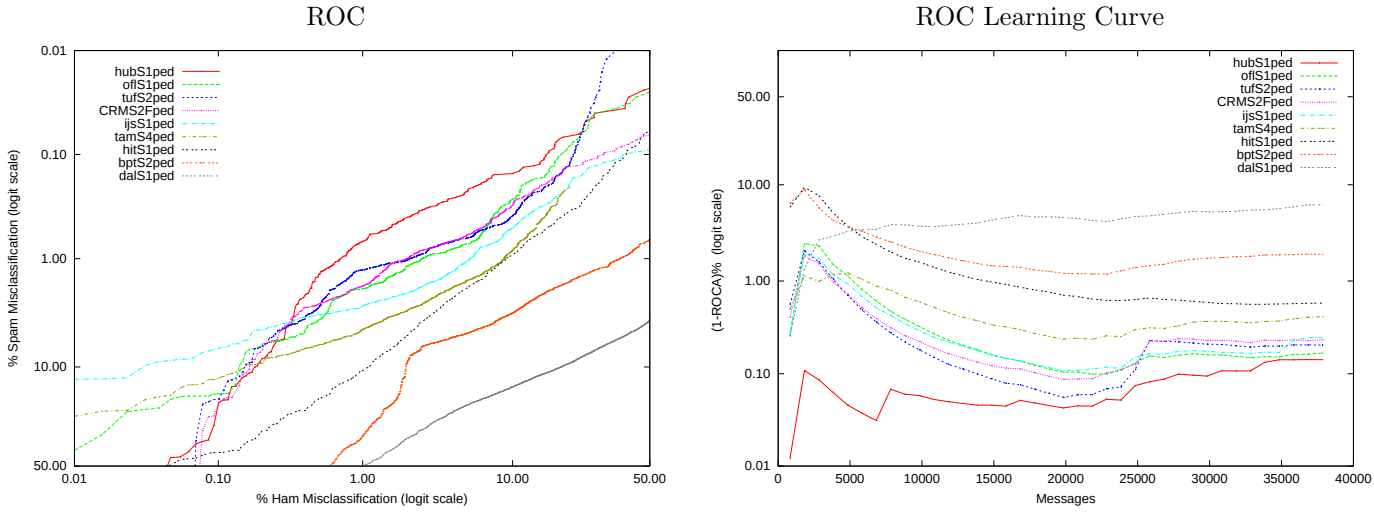

Figure 2: trec06p Public Corpus – Delayed Feedback

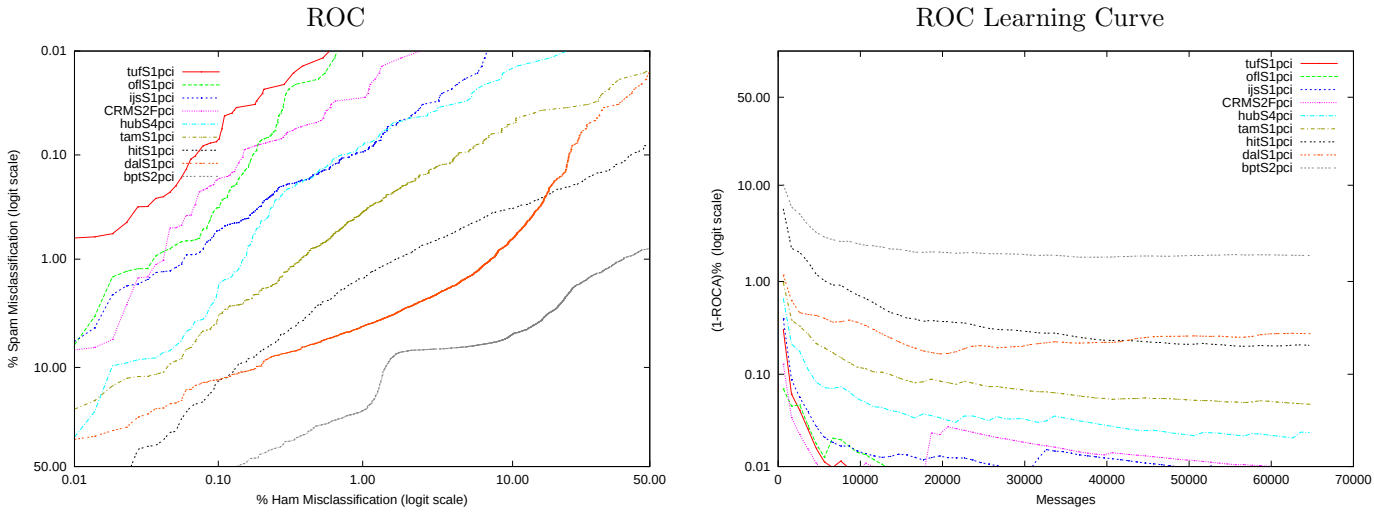

Figure 3: trec06c Chinese Corpus – Immediate Feedback

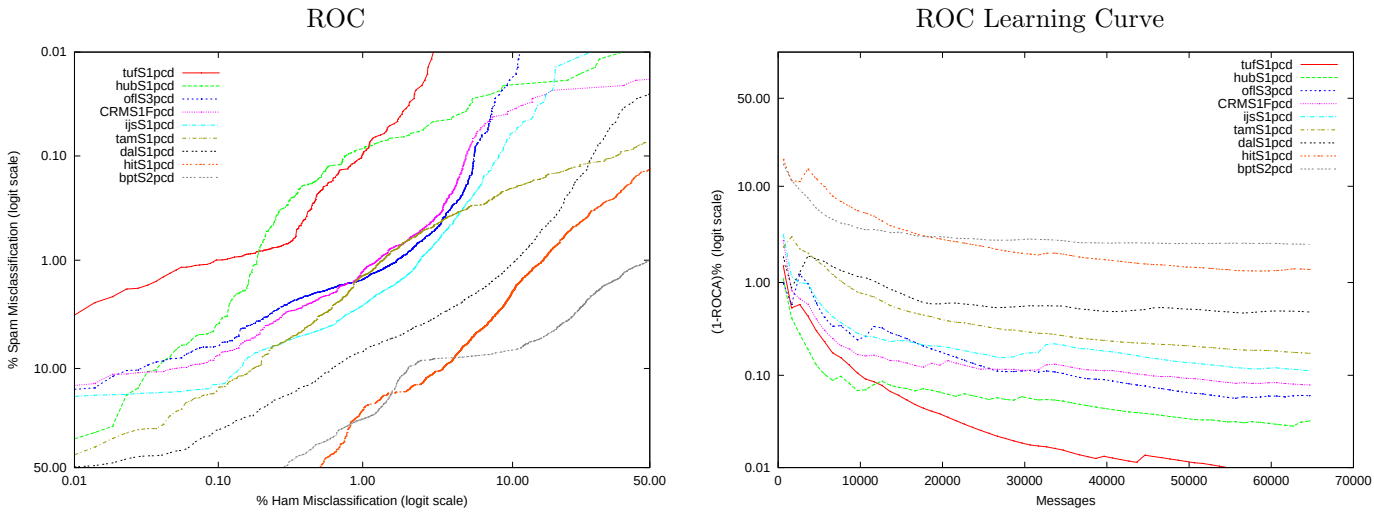

Figure 4: trec06c Chinese Corpus – Delayed Feedback

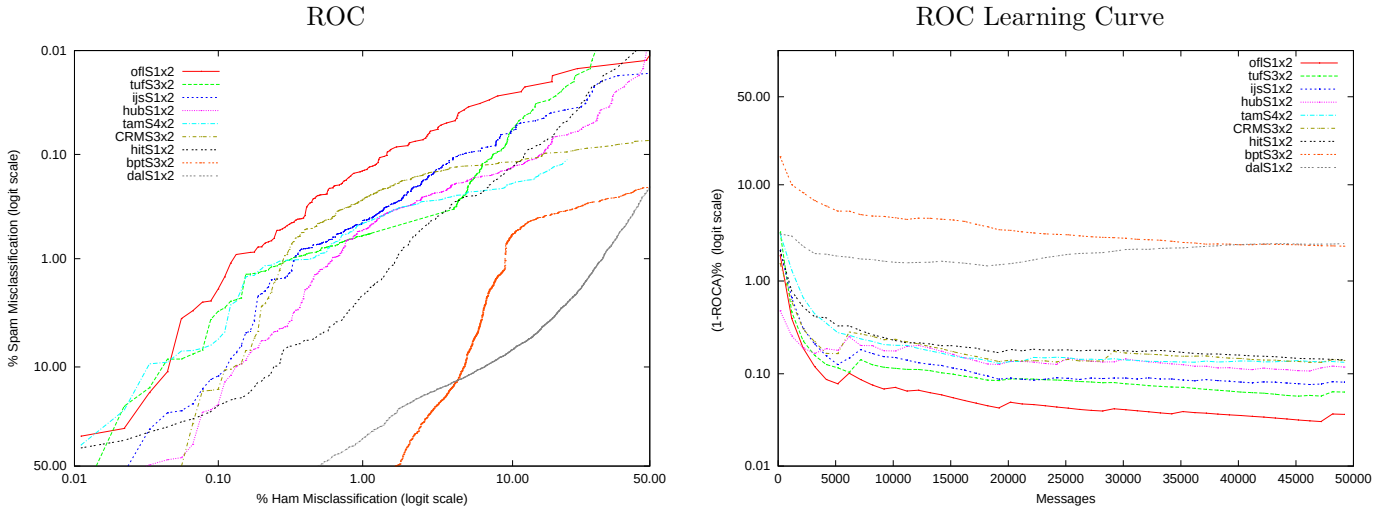

Figure 5: MrX2 Corpus – Immediate Feedback

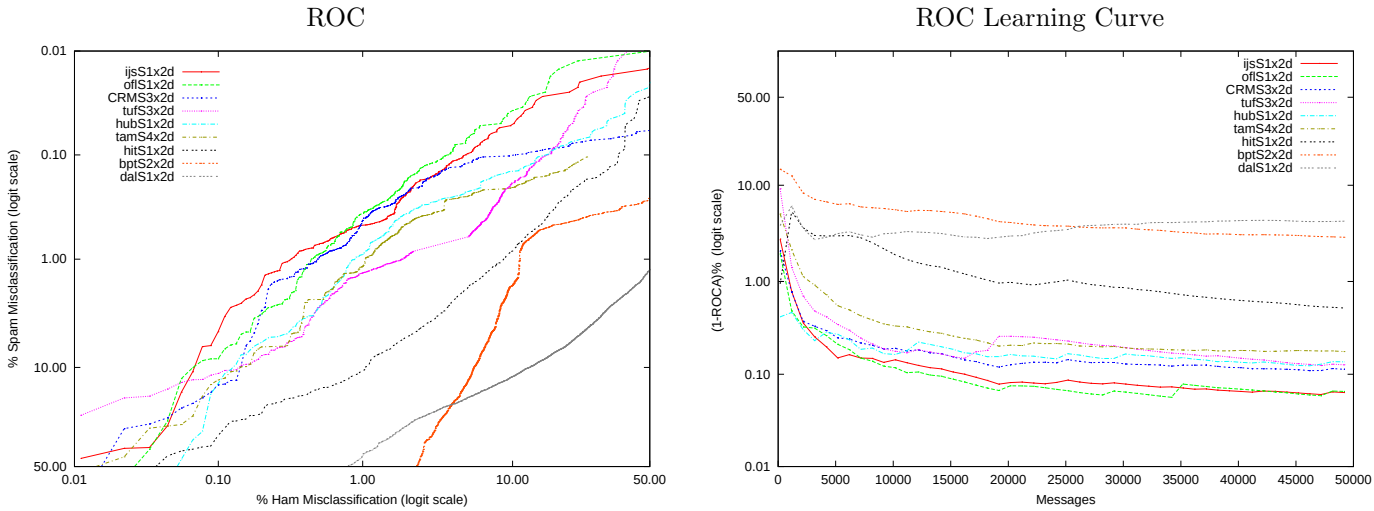

Figure 6: MrX2 Corpus – Delayed Feedback

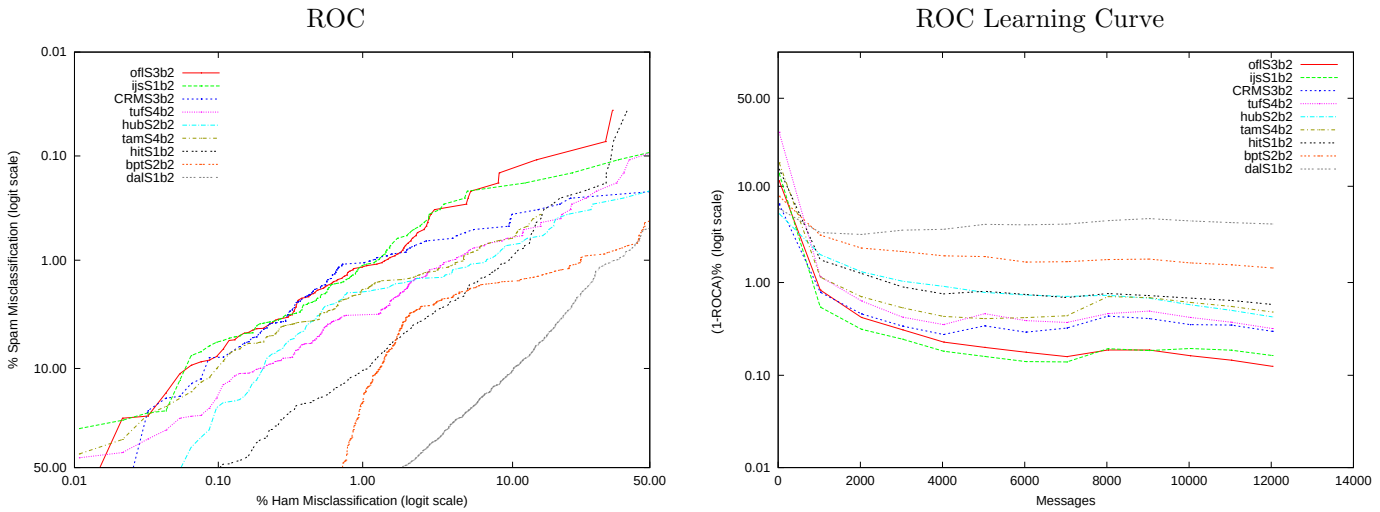

Figure 7: SB2 Corpus – Immediate Feedback

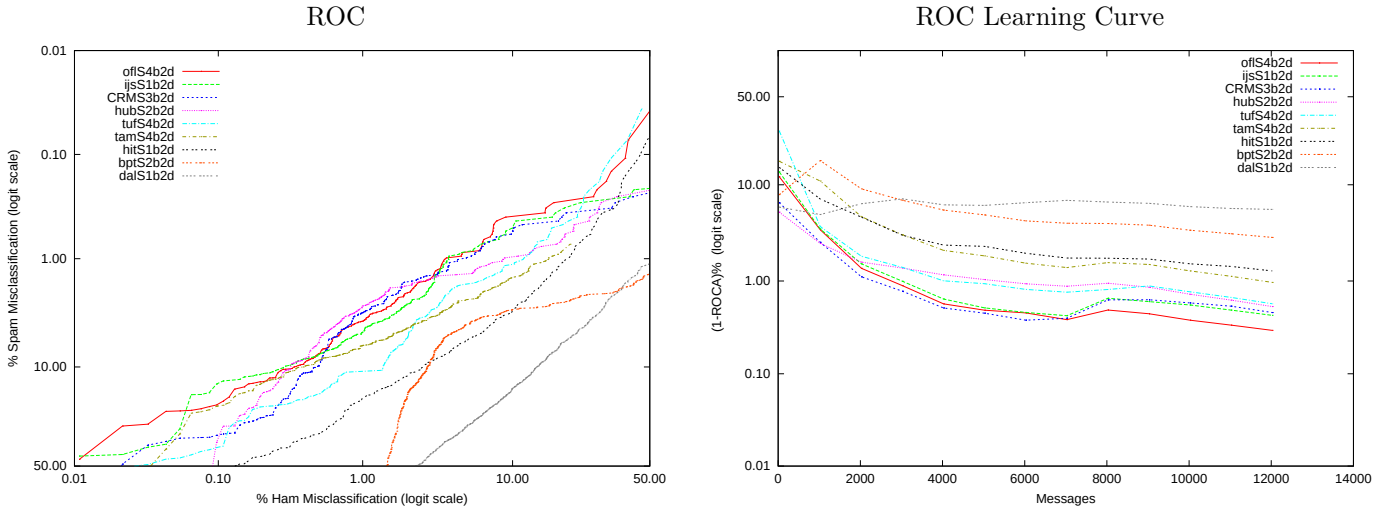

Figure 8: SB2 Corpus – Delayed Feedback

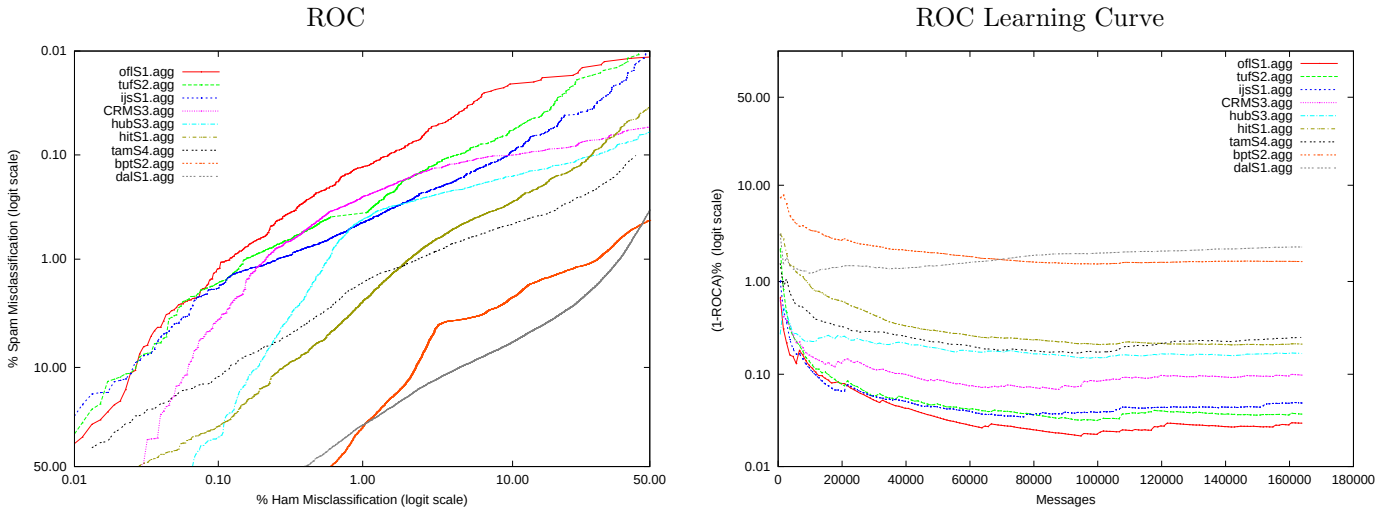

Figure 9: Aggregate Pseudo-Corpus – Immediate Feedback

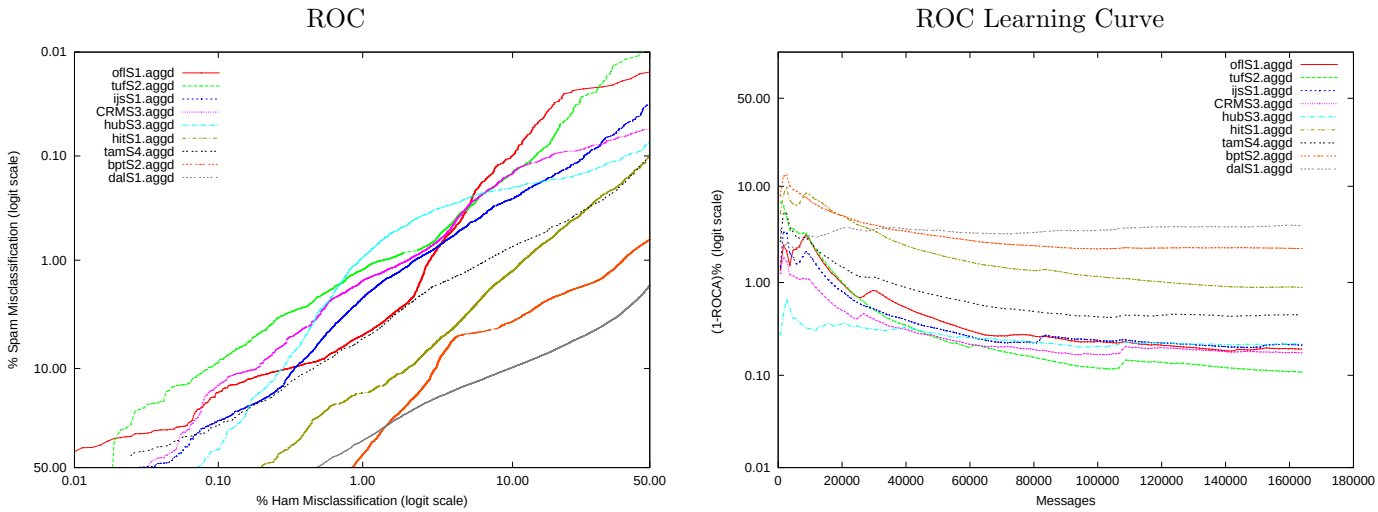

Figure 10: Aggregate Pseudo-corpus – Delayed Feedback

	Aggregate		trec06p		trec06c		MrX2		SB2	
Filter\Feedback	immediate	delay	immediate	delay	immediate	delay	immediate	delay	immediate	delay
oflS1	0.0295	0.1914	0.0540	0.1668	0.0035	0.0666	0.0363	0.0651	0.1300	0.3692
oflS3	0.0327	0.1908	0.0562	0.1702	0.0035	0.0601	0.0523	0.0824	0.1249	0.3174
oflS2	0.0365	0.2018	0.0597	0.2045	0.0104	0.1297	0.0525	0.0931	0.1479	0.3659
tufS2	0.0370	0.1079	0.0602	0.2038	0.0031	0.0104	0.0691	0.1449	0.3379	0.6923
oflS4	0.0381	0.1828	0.0583	0.1965	0.0077	0.0855	0.0718	0.1155	0.1407	0.2941
tufS1	0.0445	0.1262	0.0602	0.2110	0.0023	0.0081	0.0953	0.1991	0.3899	0.8361
ijsS1	0.0488	0.2119	0.0605	0.2457	0.0083	0.1117	0.0809	0.0633	0.1633	0.4276
tufS3	0.0705	0.1497	-	-	-	-	0.0633	0.1263	0.3350	0.6137
tufS4	0.0749	0.1452	-	-	-	-	0.0750	0.1314	0.3199	0.5696
CRMS3	0.0978	0.1743	0.1136	0.2762	0.0105	0.0888	0.1393	0.1129	0.2983	0.4584
CRMS2	0.1011	0.1667	0.1153	0.2325	0.0094	0.0975	0.1592	0.1143	0.4196	0.6006
CRMS1	0.1081	0.2165	0.1135	0.2447	0.0218	0.0784	0.1498	0.1341	0.3852	0.6346
hubS3	0.1674	0.2170	0.1564	0.1958	0.0353	0.0495	0.2102	0.2294	0.6225	0.8104
hubS4	0.1717	0.2400	0.1329	0.2006	0.0233	0.0330	0.1385	0.1763	0.5777	0.6784
hubS1	0.1731	0.2013	0.1310	0.1418	0.0238	0.0319	0.1180	0.1359	0.5295	0.5779
hubS2	0.1945	0.2716	0.1694	0.2952	0.0273	0.0369	0.1450	0.1827	0.4276	0.5306
hitS1	0.2112	0.8846	0.2884	0.5783	0.2054	1.3803	0.1412	0.5184	0.5806	1.2829
CRMS4	0.2375	1.5324	0.4675	2.1950	0.0579	1.7675	0.3056	0.4898	0.9653	2.0009
tamS4	0.2493	0.4480	0.2326	0.4129	0.1173	0.2705	0.1328	0.1755	0.4813	0.9653
tamS1	0.3008	1.0910	0.4103	0.8367	0.0473	0.1726	0.4011	0.6714	0.5912	4.5170
tamS2	0.9374	3.2366	1.2414	3.9352	0.4464	1.5370	-	-	6.5258	23.8125
tamS3	1.5309	2.2236	1.0602	1.8279	0.2899	1.0860	0.9514	1.5965	1.8462	6.0056
bptS2	1.6313	2.2999	1.2109	1.9264	1.8912	2.5444	2.5486	2.9571	1.4311	2.9050
bptS1	1.7867	2.6169	1.3690	2.0924	2.2829	3.0341	2.5926	3.6977	1.5545	2.9271
bptS3	1.9401	2.5669	1.3813	1.9520	2.9886	3.5715	2.3501	3.0866	1.6350	3.0487
bptS4	1.9818	2.6557	1.3215	1.9539	2.8267	3.3317	2.5100	3.4217	1.4970	3.0337
hitS2	2.1643	6.6776	0.8807	2.1074	3.2501	10.4413	1.2270	5.7253	1.9922	5.5975
dalS1	2.3278	4.0038	3.1383	6.3238	0.2739	0.4817	2.5035	4.3461	4.1620	5.6777
dalS2	3.2034	5.2315	4.7879	7.8412	0.4715	0.7934	5.8405	9.7809	6.9847	9.6615
hitS3	3.8063	7.8970	3.4365	7.9408	4.9442	9.6859	1.7927	5.5140	5.0801	7.7840
dalS3	6.2410	8.9847	4.0860	7.2674	0.7827	1.3635	22.9897	34.5147	4.6356	7.8237
dalS4	11.9983	19.3618	15.2705	28.3349	5.2031	9.4180	18.2197	25.6102	17.4061	24.1379

Table 4: Summary 1-ROCA (%)

baseline, as expected. It is not apparent that filters made much use of the unlabeled data in the delayed

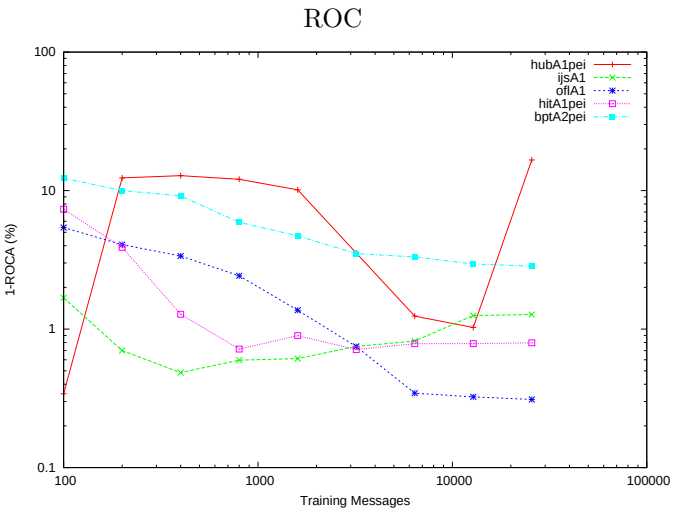

Figure 11: Active Learning – trec06p Public Corpus

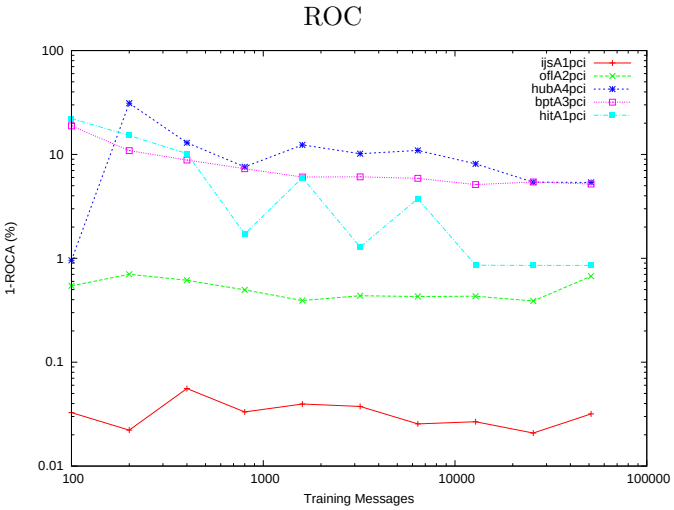

Figure 12: Active Learning – trec06c Chinese Corpus

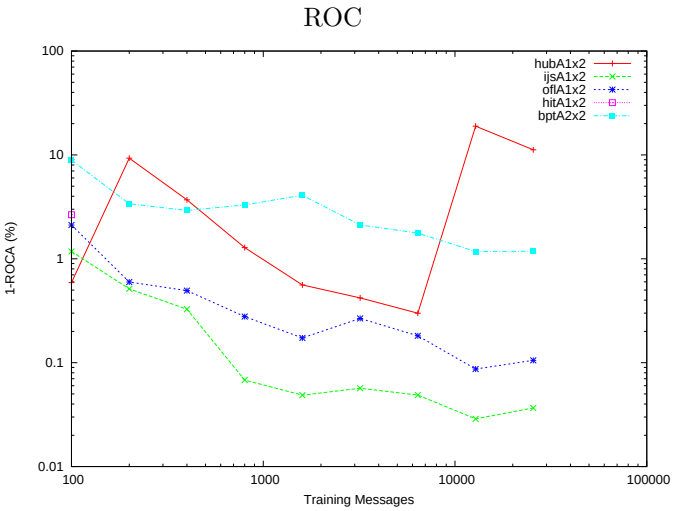

Figure 13: Active Learning – MrX2 Corpus

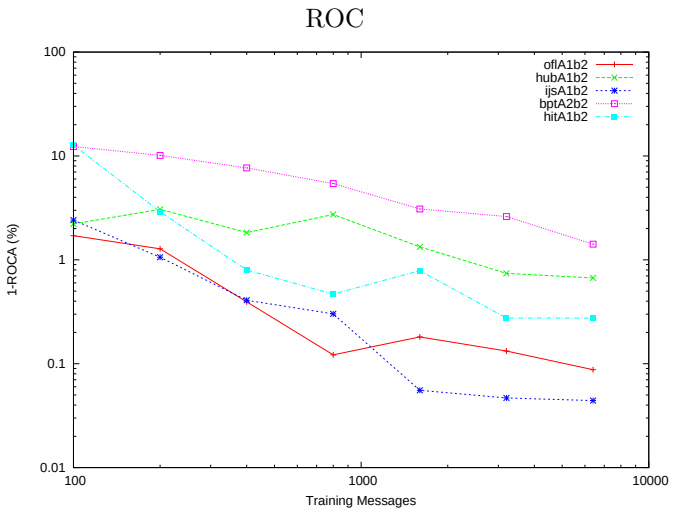

Figure 14: Active Learning – SB2 Corpus

feedback task; individual participant reports will reveal this. The active learning task presents a significant challenge.

Detailed comparison between TREC 2005 and TREC 2006 results have yet to be made, but it appears that

1. The best (and median) filter performance has improved over last year
2. The new corpora are no "harder" than the old ones: spammers have not defeated filters
3. Challenges remain in exploiting unlabeled data for spam classification, within the framework of the delayed filtering and active learning tasks.

9 Acknowledgements

The authors thank Stefan Buettcher and Quang-Anh Tran for their invaluable contributions to this effort.

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., and Bratko, A. Batch and on-line spam filter evaluation. In *Proceedings of CEAS 2006* (Mountain View, CA, 2006).