The TREC–9 Filtering Track Final Report

Stephen Robertson
Microsoft Research
Cambridge, UK
ser@microsoft.com

David A. Hull
WhizzBang Labs
Pittsburgh PA, USA
dhull@whizbang.com

Abstract

The TREC–9 filtering track measures the ability of systems to build persistent user profiles which successfully separate relevant and non-relevant documents. It consists of three major subtasks: adaptive filtering, batch filtering, and routing. In adaptive filtering, the system begins with only a topic statement and a small number of positive examples, and must learn a better profile from on-line feedback. Batch filtering and routing are more traditional machine learning tasks where the system begins with a large sample of evaluated training documents. This report describes the track, presents some evaluation results, and provides a general commentary on lessons learned from this year's track.

1 Introduction

A text filtering system sifts through a stream of incoming information to find documents relevant to a set of user needs represented by profiles. Unlike the traditional search query, user profiles are persistent, and tend to reflect a long term information need. With user feedback, the system can learn a better profile, and improve its performance over time. The TREC filtering track tries to simulate on-line time-critical text filtering applications, where the value of a document decays rapidly with time. This means that potentially relevant documents must be presented immediately to the user. There is no time to accumulate and rank a set of documents. Evaluation is based only on the quality of the retrieved set.

Filtering differs from search in that documents arrive sequentially over time. The TREC filtering track consists of three subtasks: adaptive filtering, batch filtering, and routing. In adaptive filtering, the system starts with only a user profile and (in TREC–9) a very small number, two or four, of positive examples (relevant documents). It must begin filtering documents without any other prior information. Each retrieved document is immediately judged for relevance, and this information can be used by the system to adaptively update the filtering profile. In batch filtering and routing, the system starts with a large set of evaluated training documents which can be used to help construct the search profile. For batch filtering, the system must decide to accept or reject each document, while routing systems can return a ranked list of documents. The core tasks have remained the same in TREC–7 through TREC–9.

Traditional adhoc retrieval and routing simulate a non-interactive process where users look at documents once at the end of system processing. This allows for ranking or clustering of the retrieved set. The filtering model is based on the assumption that users examine documents periodically over time. The actual frequency of user interaction is unknown and task-dependent. Rather than create a complex simulation which includes partial batching and ranking of the document set, we make the simplifying assumption that users want to be notified about interesting documents as soon as they arrive. Therefore, a decision must be made about each document without reference to future documents, and the retrieved set is ordered by time, not estimated likelihood of relevance. The history and development of the TREC Filtering Track can be traced by reading the yearly final reports:

- TREC–8 http://trec.nist.gov/pubs/trec8/t8_proceedings.html (#3 - 2 files) [5]
- TREC–7 http://trec.nist.gov/pubs/trec7/t7_proceedings.html (#3 - 2 files) [4]
- TREC–6 http://trec.nist.gov/pubs/trec6/t6_proceedings.html (#4 and #5) [3]
- TREC–5 http://trec.nist.gov/pubs/trec5/t5_proceedings.html (#5) [7]
- TREC–4 http://trec.nist.gov/pubs/trec4/t4_proceedings.html (#11) [6]

Information on the participating groups and their filtering systems can be found in the individual site reports, also available from the TREC web site.

2 TREC–9 Task Description

The basic filtering tasks in TREC–9 have not changed from TREC–8, nor even from TREC–7, with two exceptions. These are: (a) the provision of a few positive training examples for the adaptive filtering task; and (b) the introduction of a new evaluation measure. The corpus and topics are also somewhat different from those used previously. In this section, we review the corpus, the three sub-tasks, the submission requirements, and the evaluation measures. For more background and motivation, please consult the TREC–7 track report [4].

2.1 Data

The TREC–9 filtering experiments went outside the usual TREC collections and used a slightly modified version of the OHSUMED test collection compiled by, and available from, Bill Hersh [1]. This consists of Medline documents from the years 1987–1991 and a set of requests (topics) and relevance judgements. The modified dataset used for filtering may be described as follows.

The entire collection contains about 350,000 documents. Actually these are bibliographic records containing the usual fields including abstract, although only about two thirds of the records contain abstracts. They also have a field containing MeSH headings, that is human-assigned index terms. These are assumed to arrive in identifier order, at a rate of approximately 6000 documents per month. The 1987 data (equivalent to about 9 months' worth) was extracted from the dataset to provide training material, as discussed below; the test set is therefore the 1988–91 data.

Sixty-three of the original OHSUMED topics were selected for filtering (they were selected to have a minimum of 2 definitely relevant documents in the training set)[1]. These 63 topics form the OHSU set. In addition, the MeSH headings were treated as if they were topics: the text of the topic was taken from the scope notes available for MeSH headings, and assignments of headings to documents were regarded as relevance judgements. Again they were selected, to have a minimum of 4 relevant documents in the training set and to have at least one in the final year; also very

[1]Relevance judgements for OHSUMED topics were made on a 3-point scale, not relevant, possibly relevant and definitely relevant. The training documents for adaptive filtering were definitely relevant. Systems were free to make use of the graded relevance judgements in any way they saw fit, but the final evaluation was based on treating both possibly relevant and definitely relevant as relevant.

rare and very frequent headings were excluded.[2] The remaining 4903 MeSH headings formed the MSH topic set. Finally, because of the size of this topic set which made it difficult to process in its entirety, a random sample of 500 of these was made, to form the MSH-SMP set.

2.2 Tasks

The adaptive filtering task is designed to model the text filtering process from the moment of profile construction. In TREC–9, in contrast to previous TRECs, we model the situation where the user arrives with a small number of known positive examples (relevant documents). Subsequently, once a document is retrieved, the relevance assessment (when one exists) is immediately made available to the system. Unfortunately, it is not feasible in practice to have interactive human assessment by NIST. Instead, assessment is simulated by releasing the pre-existing relevance judgement for that document. Judgements for unretrieved documents are never revealed to the system. Once the system makes a decision about whether or not to retrieve a document, that decision is final. No back-tracking or temporary caching of documents is allowed. While not always realistic, this condition reduces the complexity of the task and makes it easier to compare performance between different systems.

Systems are allowed to use the whole of the training set of 1987 documents to generate collection frequency statistics (such as IDF) or auxiliary data structures (such as automatically-generated thesauri). Resources outside the OHSUMED collection could also be used, as could the OHSUMED topics and MeSH headings excluded from the filtering task, with all relevance judgements on the 1987 documents, for system training. As documents were processed, the text could be used to update term frequency statistics and auxiliary document structures even if the document was not matched to any profile. Groups had the option to treat unevaluated documents as not relevant.

In batch filtering, all 1987 documents and all relevance judgements on that set of documents are available in advance as a training set. The 1988-91 documents form the test set. As in adaptive filtering, systems may use the relevance judgement from any retrieved document to update the filtering profile (if these are used it becomes batch-adaptive filtering). For routing, the training data is the same as for batch filtering; systems return a ranked list of the top 1000 retrieved documents from the 1988-91 set. Batch filtering and routing are included to open participation to as many different groups as possible.

2.3 Evaluation and optimisation

For the TREC experiments, filtering systems are expected to make a binary decision to accept or reject a document for each profile. Therefore, the retrieved set consists of an unranked list of documents. This fact has implications for evaluation, in that it demands a measure of effectiveness which can be applied to such an unranked set. Many of the standard measures used in the evaluation of ranked retrieval (such as average precision) are not applicable. Furthermore, the choice of primary measure of performance will impact the systems in a way that does not happen in ranked retrieval. While good ranking algorithms seem to be relatively independent of the evaluation measure used, good classification algorithms need to relate very strongly to the measure it is desired to optimise.

Two measures were used in TREC–9 for this purpose (as alternative sub-tasks). One was essentially the linear utility measure used in previous TRECs, and described below. The other was new for TREC–9, and is described as a precision-oriented measure.

[2]The reason for excluding those MeSH headings not represented in the final year was to avoid headings which had been dropped out of MeSH (which undergoes continual modification) during the period.

Precision-oriented measure

The idea of this measure is to set a target number of documents to be retrieved over the period of the simulation; the target was set at 50 documents for each topic (the same for all topics). This situation might be said to correspond roughly with cases where the user indicates what sort of volume of material they expect / are prepared for / are able to deal with / would like to see. Clearly a fixed target is a simplification of such cases (each of which is a little different from the others), but may be seen as an acceptable simplification for experimental purposes.

The measure is essentially precision, but with a penalty for not reaching the target:

```
            Number of relevant retrieved documents
      T9P = ------------------------------------------
            Max(Target, Number of retrieved documents)

      Target = 50 documents
```

This may be regarded as something akin to a "precision at [target] documents" measure. The relationship is discussed further below.

Linear utility

The idea of a linear utility measure has been described in previous TREC reports (e.g. [5]). The particular parameters being used are a credit of 2 for a relevant document retrieved and a debit of 1 for a non-relevant document retrieved:

```
      Utility = 2*R+ - N+          --> retrieve if P(rel) > .33
```

Filtering according to a utility function is equivalent to filtering by estimated probability of relevance; the corresponding probability threshold is shown.

When evaluation is based on utility, it is difficult to compare performance across topics. Simple averaging of the utility measure gives each retrieved document equal weight, which means that the average scores will be dominated by the topics with large retrieved sets (as in micro-averaging). Furthermore, the utility scale is effectively unbounded below but bounded above; a single very poor query might completely swamp any number of good queries. In TREC–8 we experimented with a range of scaled utility functions for averaging purposes. This year we have taken a simpler approach, which deals crudely with the unboundedness but not with the micro-averaging: a minimum (maximum negative) score is applied. Thus:

```
      T9U = Max(2*R+ - N+, MinU)

      MinU = -100 for OHSU topics, -400 for MeSH topics
```

Other measures

In the results presented below, and in the official results tables, a number of measures are included as well as the measure for which any particular run was specifically optimised. The range is as follows:

For adaptive and batch filtering:

MnT9P The mean value of the T9P measure over topics

MacP Mean set precision over topics (macro average = average of ratios)

MacR Mean set recall (similar)

MnT9U The mean value of the T9U measure (unnormalised) over topics

MnSU Mean normalised T9U. The T9U value for each topic was divided by the maximum possible for that topic, i.e. 2∗(total relevant) before taking the mean

Zeros The number of topics for which no documents were retrieved over the period

For routing:

AveP Mean average uninterpolated precision

P@50 Precision at 50 documents retrieved

In the official results tables, in addition to the above, total retrieved and relevant retrieved over all topics are given. For the routing runs, the full output of trec_eval is given.

2.4 Submission Requirements

Each participating group could submit a limited number of runs, according to the following table:

	OHSU	MESH	MSH-SMP
(A) Adaptive filtering runs	4	1	2
(B) Batch filtering runs	2	1	1
(C) Routing runs	2	1	1

Any of the filtering runs could be optimised for either T9P or T9U. There were no required runs, but participants were asked if possible to provide an adaptive filtering run on the OHSU topics with T9P optimisation.

Another variable allowed in the guidelines was between automatic, manual and manual with feedback for the initial query formulation. However, all groups opted to do automatic runs.

Groups were also asked to indicate whether they used other parts of the TREC collection, or other external sources, to build term collection statistics or other resources. It was also possible to use the MeSH field of the original records when running OHSU searches (not, for obvious reasons, when running MSH searches) – again, groups were asked to declare this.

3 TREC–9 results

Fourteen groups participated in the TREC–9 filtering track (the same number as in TREC–8) and submitted a total of 75 runs (substantially more than last time). These break down as follows:

Topics:	OHSU:	53 runs
	MESH:	10 runs
	MSH-SMP:	12 runs
Tasks:	Adaptive filtering:	42 runs
	Batch:	19 runs (of which 8 adaptive)
	Routing:	14 runs

```
Measures:    T9P:                 32 runs
             T9U:                 27 runs
             (undeclared):         2 runs
```

The total of 61 under 'Measures' represents the 61 Adaptive filtering or Batch runs. The two whose optimisation measure was undeclared were evaluated for both measures.

Here is a list of the participating groups, including [abbreviations] and (run identifiers). Participants will generally be referred to by their abbreviations in this paper. The run identifiers can be used to recognize which runs belong to which groups in the plotted results.

- Carnegie-Mellon University, Ault & Yang [CMU-Y] (CMUCAT)
- Carnegie-Mellon University, Zhang & Callan [CMU-C] (CMUDIR)
- Queens College CUNY [CUNY] (pirc)
- Fudan University [Fudan] (FDU)
- Informatique-CDC - Groupe Caisse des Dépôts / ESPCI [ICDC] (S2RN)
- University of Iowa [Iowa] (IOWAF)
- IRIT / University of Toulouse [IRIT] (Mer9)
- Korea Advanced Institute of Science and Technology [KAIST] (KAIST)
- KDD R&D Laboratories [KDD] (kdd)
- Microsoft Research - Cambridge [Microsoft] (ok9)
- University of Montreal [Montreal] (reliefs)
- University of Nijmegen [Nijmegen] (KUN)
- Rutgers University [Rutgers] (ant)
- Seoul [Seoul] (scai)

3.1 Summary of approaches

These very brief summaries are intended only to point readers towards other work. A few of the groups do not have papers in this volume.

Carnegie-Mellon (CMU-Y) have adapted a k-nearest-neighbour text categorization algorithm to the filtering task. In their paper in this volume, they also present arguments against the T9P and T9U measures.

The other Carnegie-Mellon group (CMU-C) used an incremental Rocchio algorithm for query adaptation, and introduced some modifications into this algorithm and their idf term weighting.

CUNY ran last year's adaptive filtering system without modification.

Fudan used a Rocchio-like algorithm, with feature selection using mutual information.

ICDC (routing task) used a neural network without hidden neurons, but with strong feature selection (very few features per topic), with local context.

Iowa used an approach based on dynamic, two-level clustering: each topic has a single first-level cluster, within which further clusters develop.

IRIT's method involved a profile of the non-relevant documents for each profile.

KAIST combined query zoning, a support vector machine and Rocchio's algorithm.

KDD also used a non-relevant document profile, and pseudo-relevance feedback.

Microsoft used limited term selection, and a complex threshold adaptation regime.

Montreal combined the 'document implies query' relevance probability with the reverse implication, and used word conjunctions.

Nijmegen used a method of threshold adaptation based on score distributions, with a Rocchio algorithm. Relevant documents are treated differently according to their date.

Rutgers used Boolean expressions based on the Logical Analysis of Data.

Seoul used a boosted naive bayes method.

3.2 Evaluation results

Some results for the various participating groups are presented in the following tables. Tables 1–4 show the adaptive filtering results on OHSU and MSH-SMP topics, for the two optimisation measures. Various measures are shown in the tables, in addition to the optimisation measure used.

Figures 1–4 show the same result sets broken down by year. In the case of the T9P optimised results, we cannot calculate T9P itself for each year, because the target number of documents applies only to the whole period. Instead, precision is shown. Similarly, it is not appropriate to apply the minimum utility value used in T9U to each individual year – in this case, unadjusted utility is used.

The year graphs for precision might be a little misleading. It would clearly be possible for a system to set its threshold too high at the beginning, so that it obtained good precision but not enough documents; it would then have to relax its threshold in order to retrieve enough documents, but probably get lower precision.

Tables 5–8 show the batch filtering and routing results.

Table 1: Adaptive filtering – OHSU – best T9P results

	MacR	MacP	**MnT9P**	MnT9U	Zeros
Microsoft	38.8	29.4	**29.4**	-6.3	0
CMU-C	41.4	27.9	**27.9**	-5.3	0
Fudan	30.1	27.3	**26.5**	-7.0	0
Nijmegen	29.3	25.8	**25.8**	-7.3	0
CMU-Y	38.0	27.8	**22.4**	-22.1	0
Montreal	17.7	28.0	**16.8**	-1.5	0
Iowa	16.3	19.5	**13.8**	-11.4	5
Rutgers	33.2	15.3	**10.2**	-43.6	7
OHSU topics Runs optimised for T9P Best runs from each group					

3.3 Some comparisons

The new T9P measure provides an interesting opportunity to compare the results of filtering runs with traditional ranked-retrieval runs. The evaluation program trec_eval for ranked retrieval calculates P@n values – precision at n documents retrieved – for various values of n. T9P is a sort of

Table 2: Adaptive filtering – OHSU – best T9U results

	MnT9U	MnSU	MacR	MacP	Zeros
Nijmegen	**17.3**	0.06	23.1	36.7	0
Microsoft	**10.7**	0.01	21.2	33.0	0
CMU-C	**10.1**	0.04	19.0	42.6	0
Fudan	**9.6**	0.00	18.1	31.9	0
Montreal	**1.1**	-0.08	12.0	32.1	1
Iowa	**-5.9**	-0.16	12.9	19.8	6
Rutgers	**-32.3**	-1.75	27.0	14.3	12
KDD	**-35.3**	-0.90	8.3	10.2	0
CUNY	**-55.7**	-11.20	22.2	10.8	0
OHSU topics					
Runs optimised for T9U					
Best runs from each group					

Table 3: Adaptive filtering – MSH – best T9P results

	MacR	MacP	**MnT9P**	MnT9U	Zeros
MSH topics					
Microsoft	18.5	41.9	**41.9**	14.8	0
CMU-C	18.5	35.9	**35.9**	19.2	0
Fudan	13.9	35.8	**35.1**	4.5	0
CMU-Y	21.0	35.9	**30.3**	-40.8	105
MSH-SMP topics					
Microsoft	18.9	43.0	**43.0**	16.5	0
CMU-C	18.9	36.3	**36.3**	17.5	0
Fudan	14.2	36.3	**35.6**	5.6	0
CMU-Y	21.4	36.4	**30.4**	-37.4	10
Runs optimised for T9P					
Best runs from each group					

Table 4: Adaptive filtering – MSH – best T9U results

	MnT9U	MnSU	MacR	MacP	Zeros
MSH topics					
CMU-C	**20.3**	0.07	9.8	47.9	0
MSH-SMP topics					
Microsoft	**46.5**	0.10	18.2	43.6	0
Fudan	**29.3**	0.04	15.4	34.6	0
CMU-C	**26.7**	0.08	13.7	47.1	0
Iowa	**12.9**	0.0	11.5	52.5	52
Runs optimised for T9U					
Best runs from each group					

Table 5: Batch filtering – OHSU – best T9P results

	MacR	MacP	**MnT9P**	MnT9U	Zeros
Batch-adaptive					
Fudan	37.9	32.2	**31.7**	-1.1	0
Microsoft	38.8	30.5	**30.5**	-5.3	0
Non-adaptive					
CMU-Y	57.4	28.7	**26.1**	-26.9	1
KAIST	22.7	42.1	**20.0**	12.2	0
OHSU topics					
Runs optimised for T9P					
Best runs from each group					

Table 6: Batch filtering – OHSU – best T9U results

	MnT9U	MnSU	MacR	MacP	Zeros
Batch-adaptive					
Nijmegen	**19.4**	0.03	41.0	37.6	0
Fudan	**13.6**	0.05	24.5	39.0	0
Non-adaptive					
IRIT	**7.5**	-0.22	21.6	46.5	5
Nijmegen	**5.0**	-0.15	22.6	40.0	3
Seoul	**2.8**	0.02	3.7	80.8	33
OHSU topics					
Runs optimised for T9U					
Best runs from each group					

Table 7: Batch filtering – MSH – best T9P & T9U results

	MacR	MacP	**MnT9P**	MnT9U	Zeros
MSH topics – adaptive – T9P					
Fudan	16.1	42.2	**41.8**	14.6	0
MSH topics – non-adaptive – T9P					
CMU-Y	26.0	53.1	**43.6**	4.9	97
MSH-SMP topics – adaptive – T9P					
Fudan	17.6	45.0	**44.0**	19.4	0
Microsoft	18.7	43.3	**43.3**	16.9	0
MSH-SMP topics – non-adaptive – T9P					
CMU-Y	25.7	54.6	**44.3**	11.2	9
KAIST	24.5	54.3	**41.9**	86.4	0
MSH-SMP topics – non-adaptive – T9U					
	MnT9U	MnSU	MacR	MacP	Zeros
Seoul	**20.0**	0.01	5.4	58.8	127
Best runs from each group					

Table 8: Best Routing results

	AveP	P@50	RPrec
OHSU topics			
ICDC	**0.385**	37.0	39.5
Microsoft	**0.326**	33.6	35.1
Nijmegen	**0.237**	28.2	28.6
IRIT	**0.235**	27.9	27.8
Rutgers	**0.182**	21.5	22.9
MSH-SMP topics			
ICDC	**0.335**	53.7	41.0
Microsoft	**0.253**	45.5	32.6
Rutgers	**0.158**	27.0	23.0
Best runs from each group			

P@50[3].

Actually, the comparison between P@50 and T9P is slightly more complex. There are two reasons why we might expect T9P figures to be somewhat lower than P@50:

1. In order to reach a target of 50, we have in fact to aim higher, as discussed above. We would be substantially penalised for not reaching 50. However, going higher is also likely to penalize us somewhat on precision.

2. Even supposing we were to retrieve exactly 50 over the period, they would not necessarily be the best 50 – if we had to adjust the threshold at any stage to achieve the target number at the end, for part of the period we would have retrieved documents in a range of scores which we would have rejected in another part.

As against these arguments, of course, there is the possibility for improving the results through adaptation. In table 9 we show P@50 values for some of the routing runs and T9P for some batch and adaptive filtering runs. The following observations qualify this table: The ICDC and CMU-Y runs made use of the MeSH field of the records. The CMU-Y run was batch non-adaptive; the other two were adaptive. As indicated in the text, the routing and batch filtering runs could make use of all of the relevant documents in the training set. Given that some topics have less than 50 relevant documents, the maximum possible P@50 is 68%.

Although it is clearly possible, for the reasons suggested above, to do better at P@50 than at T9P, it seems that a good adaptive filtering system can overcome much of this handicap.

4 General Commentary

In this section last year, we made the following observation:

> Following the progression of system performance from TREC–7 to TREC–8 (or lack thereof!), it is becoming increasingly clear that the adaptive filtering task is too hard.

We believe, in contrast, that the TREC–9 adaptive filtering task was not too hard, and provided a solid experimental test from which a number of systems emerged with good performances. We may make the following observations in support of this claim:

[3]trec_eval does not by default include $n = 50$; however, a simple modification of a header file allows it.

Table 9: Comparing T9P and P@50

Routing: P@50		Batch filtering: T9P		Adaptive filtering: T9P	
ICDC	37.0	Fudan	31.7	Microsoft	29.4
Microsoft	33.6	Microsoft	30.5	CMU-LTI	27.9
Nijmegen	28.2	CMU-Y	26.1	Fudan	26.5
OHSU topics Filtering runs optimised for T9P Best runs from best 3 group					

- In the linear utility version of the task, several systems with non-conservative strategies achieved good positive average utilities.
- With the new precision-oriented measure, several systems achieved effectiveness levels close to those reached in the less-demanding P@50 task in ranked-output retrieval.

It is not immediately clear which particular differences between the TREC–8 and TREC–9 tasks brought about this change. Probably all of the following had some effect, but it would be necessary to do some more diagnostic work to discover their relative importance:

- The use of a small number of positive examples for training;
- The greater number of relevant documents in the test collection;
- Improvements in the systems.

Acknowledgements We give our thanks to all the people who have contributed to the development of the TREC filtering track over the years, in particular David Lewis, Karen Sparck Jones, Chris Buckley, Paul Kantor, Ellen Voorhees, the TREC program committee, and the team at NIST.

References

[1] Hersh, W.R., Buckley, C., Leone, T.J. and Hickam, D.H. OHSUMED: an interactive retrieval evaluation and new large test collection for research. In *SIGIR '94: Proceedings of the 17th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, Springer-Verlag, pages 192–201, 1994.

[2] I.J. Good. The Decision Theory Approach to the Analysis of Information Retrieval Systems. *Information Storage and Retrieval Systems*, 3:31–34, 1967.

[3] Hull, David A. The TREC-6 Filtering Track: Description and Analysis. In *The 6th Text Retrieval Conference (TREC-6), NIST SP 500-240*, pages 45–68, 1998.

[4] Hull, David A. The TREC-7 Filtering Track: Description and Analysis. In *The 7th Text Retrieval Conference (TREC-7), NIST SP 500-242*, pages 33–56, 1999.

[5] Hull, David A. and Robertson, Stephen The TREC-8 Filtering Track final report. In *The 8th Text Retrieval Conference (TREC-8), NIST SP 500-246*, pages 35–56, 2000.

[6] Lewis, David. The TREC-4 Filtering Track. In *The 4th Text Retrieval Conference (TREC-4), NIST SP 500-236*, pages 165–180, 1996.

[7] David Lewis. The TREC-5 Filtering Track. In *The 5th Text Retrieval Conference (TREC-5), NIST SP 500-238*, pages 75–96, 1997.