The TREC 2002 Filtering Track Report

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

Ian Soboroff
NIST
Gaithersburg, MD, USA
ian.soboroff@nist.gov

Abstract

The TREC–11 filtering track measures the ability of systems to build persistent user profiles which successfully separate relevant and non-relevant documents in an incoming stream. 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 a very small number 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 larger 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 for TREC–11 are very similar to those investigated in TREC–7 through TREC–10.

Traditional ad hoc 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–10 http://trec.nist.gov/pubs/trec10/t10_proceedings.html (#3) [9]
- TREC–9 http://trec.nist.gov/pubs/trec9/t9_proceedings.html (#3) [8]
- TREC–8 http://trec.nist.gov/pubs/trec8/t8_proceedings.html (#3 - 2 files) [4]
- TREC–7 http://trec.nist.gov/pubs/trec7/t7_proceedings.html (#3 - 2 files) [3]
- TREC–6 http://trec.nist.gov/pubs/trec6/t6_proceedings.html (#4 and #5) [2]
- TREC–5 http://trec.nist.gov/pubs/trec5/t5_proceedings.html (#5) [6]
- TREC–4 http://trec.nist.gov/pubs/trec4/t4_proceedings.html (#11) [5]

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–11 Task Description

For those familiar with previous TRECs, the basic filtering tasks in TREC–11 are similar to those investigated in TREC–7 through TREC–10. The corpus is the same as for TREC–10, but a new set of topics has been prepared. 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 [3].

2.1 Data

This year, the track has again used the RCV1 corpus provided by Reuters for research purposes [7]. This is a collection of about 800,000 news stories, covering a time period of a year in 1996-7. Items in the collection have unique identifiers and are dated but not timed. For the purpose of the experiment, it is assumed that the time-order of items within one day is the same as identifier order. (Item id on its own is insufficient for ordering, as there is some conflict across days). The first 6 weeks' items, 20 August through 30 September 1996, were taken as the training set (which could be used in ways specified below). The remainder of the collection formed the test set.

A new set of 100 topics was prepared for this year. Fifty of these were constructed in the traditional TREC fashion, by the assessors at NIST. In order to provide the necessary relevance judgements for training (including adaptive filtering), extensive searches using multiple retrieval and classification systems were conducted at NIST after initial definition of the topics, and the assessors made relevance judgements on the fused output. This process included several feedback stages, so that after one round of such assessment, relevance information was used to improve the queries and another round of assessments was made. Feedback continued until no more relevant documents were found in a given round, or until five rounds had passed. Each topic received between two and seven rounds of judging (some topics had more than five rounds due to glitches in the feedback system).

Additional relevance judgements were made for these assessor topics after submission of results by the participants, on documents taken from the pooled submissions for each topic. These resulted in the identification of additional relevant documents, which were not available to the adaptive systems, but which were included for the purpose of evaluating all systems. All results below are based on the full set of relevance judgements. Further details and analysis on this post-submission phase of judgements is given below (section 4.1). Additional discussion of both pre- and post-submission judgements, and the whole process of constructing the new topic sets, is given in [10].

The remaining fifty topics were constructed as intersections of pairs of Reuters categories. Pairs of categories were chosen to be apparently meaningful as search topics, to have a minimum of three relevant documents in the training set, and to have an overall number of relevant documents in the range of the assessor-built topics. (Relevant documents are here defined as documents assigned both category labels in the Reuters collection.) For the purposes of training for batch filtering and routing, and in order to make this set of topics similar to the previous set of 50, a selection of non-relevant documents was included in the set of relevance judgements provided for each topic. These non-relevant documents were chosen randomly from those assigned either of the category labels, but not both. This places the non-relevant documents in the "neighbourhood" of the intersection, hopefully similar to highly-ranked documents in a pool which are judged irrelevant by an assessor.

This second set of topics represents a trial of a relatively cheap way of constructing topics for retrieval experiments, given a collection with category labels assigned. It is regarded as an experiment to assess whether such a methodology is likely to be useful for future experiments.

2.2 Tasks

The adaptive filtering task is designed to model the text filtering process from the moment of profile construction. In TREC–11, following the idea first used in TREC–9, we model the situation where the user arrives with a small number of known positive examples (relevant documents). For each topic, the last three relevant documents in the training set were made available to the participants for this purpose; no other relevance judgements from the training set could be used. 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 documents (but no other relevance judgements than the three provided for each topic) to generate collection frequency statistics (such as inverse document frequencies) or auxiliary data structures (such as automatically-generated thesauri). Resources outside the Reuters collection could also be used. 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 the training set documents and all relevance judgements on that set are available in advance. Once the system is trained, the test set is processed in its entirety. For each topic, the system returns a single retrieved set. For routing, the training data is the same as for batch filtering, but in this case systems return a ranked list of the top 1000 retrieved documents from the test set. Batch filtering and routing are included in order to encourage 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–11 for this purpose (as alternative sub-tasks). One was essentially the linear utility measure used in previous TRECs, and described below. The other was a version of the van Rijsbergen measure of retrieval performance, first used in TREC–10.

F-beta

This measure, based on one defined by van Rijsbergen, is a function of recall and precision, together with a free parameter beta which determines the relative weighting of recall and precision. For any beta, the measure lies in the range zero (bad) to 1 (good). For TREC–11 (as for TREC–10), a value of beta=0.5 has been chosen, corresponding to an emphasis on precision (beta=1 is neutral). The measure (with this choice of beta) may be expressed as follows:

$$\mathrm{T11F} = \frac{1.25 \times \text{No. of relevant docs retrieved}}{\text{No. of retrieved docs} + 0.25 \times \text{No. of relevant docs}}$$

(T11F is defined as zero if the number of retrieved documents is zero.)

Linear utility

The idea of a linear utility measure has been described in previous TREC reports (e.g. [4]). 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:

$$\mathrm{T11U} = 2 \times \text{No. of relevant docs retrieved} - \text{No. of non} - \text{relevant docs retrieved}$$

which corresponds to the retrieval rule:

$$\text{retrieve if } \; P(\text{rel}) > .33$$

Filtering according to a linear 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.

For the purpose of averaging across topics, the method used for TREC–11 is a slightly modified version of one used in TREC–9 (modification proposed by Ault). First, utilities are normalised by the maximum possible utility for the topic, namely

$$\mathrm{MaxU} = 2 \times (\text{No. of relevant docs})$$

I.e.

$$\mathrm{T11NU} = \frac{\mathrm{T11U}}{\mathrm{MaxU}}$$

The lower limit is some negative normalised utility, MinNU, which may be thought of as the minimum (maximum negative) utility that a user would tolerate, over the lifetime of the profile. If the T11NU value falls below this minimum, it will be assumed that the user stops looking at documents, and therefore the minimum is used. For each topic,

$$\mathrm{T11SU} = \frac{\max(\mathrm{T11NU}, \mathrm{MinNU}) - \mathrm{MinNU}}{1 - \mathrm{MinNU}}$$

and MeanT11SU is the mean of T11SU over topics.

Different values of MinNU may be chosen. The primary evaluation measure has

$$\mathrm{MinNU} = -0.5$$

Other measures

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:

- Mean T11SU (scaled utility) over topics, over the whole period and broken down by time period for adaptive filtering. Note that this is referred to in the tables as T11U, but is in fact T11SU.
- Mean T11F (F-beta, with beta = 0.5) over topics.
- Mean set recall
- Mean set precision
- Zeros (number of topics for which no documents were retrieved over the period)

All means are macro-averages, that is, averaged across topics. For routing, the usual range of ranked-output performance measures computed by `trec_eval` are given.

2.4 Submission Requirements

Each participating group could submit a limited number of runs, in each category: Adaptive filtering 4; Batch filtering 2; Routing 2.

Any of the filtering runs could be optimised for either T11F or T11SU – a declaration was required of the measure for which each run was optimised. There were no required runs, but participants were encouraged to provide an adaptive filtering run with T11SU optimisation.

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.

3 TREC–11 results

Twenty one groups participated in the TREC–11 filtering track (two more than in TREC–10) and submitted a total of 73 runs (seven more than in TREC–10)). These break down as follows: 14 groups submitted adaptive filtering runs, 10 submitted to batch filtering, and 10 to routing.

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 recognise which runs belong to which groups in the plotted results.

	Abbreviation	Run identifier
University of North Texas	north_texas	UNTextCat
KerMIT Consortium	kerMIT	KerMIT
Carnegie Mellon University	cmu_lti	CMUDIR
University of Hertfordshire	hertfordshire	UHcl
Microsoft Research Cambridge	microsoft_cambridge	ok11, msPUM
Moscow Medical Academy	moscow_med	mma2002
Rutgers University	rutgers-kantor	dimacs11
David D. Lewis, Independent Consultant	Lewis	dimacsdd
SUNY Buffalo	buffalo_cedar	cedar02
CLIPS Laoratory, IMAG	clips-imag	relief
National Institute of Informatics	nii	kNII11
Clairvoyance Corporation	clairvoyance	CCT11
Institut de Recherche en Informatique de Toulouse	irit	iritsig
Tampere University of Technology	tampere	Visa
Fudan University	Fudan	FDUT11
Queens College, City University of New York	cuny	pirc2
Chinese Academy of Sciences	chinese_academy	ICT
Queensland University of Technology	queensland	QUT
Johns Hopkins University Applied Physics Lab	jhu_apl	apl11
Tsinghua University	tsinghua	thuT11
University of Iowa	uiowa	UIowa02

3.1 Summary of approaches

These brief summaries are intended only to point readers toward other work. Not all groups have a paper in the proceedings.

University of North Texas participated in the batch filtering and routing tasks. Their TextCat system employs multiple simple text classifiers (an n-gram based one and Ripper) which may be combined by stacking them in series or using a voting scheme.

KerMIT Consortium participated in all three tasks. Their focus is on support vector machine (SVM) kernel methods. For routing, they used a linear SVM, and for batch filtering used the same SVM with a threshold selection mechanism. For adaptive filtering, they used second-order perceptrons and combined SVMs and perceptrons with uneven margins.

CMU participated in the adaptive filtering task. Their system was the same as used in TREC 9 and 10 and uses Rocchio's algorithm for profile updating. Their thresholding and term selection processes were chosen and tuned using past TREC filtering data.

University of Hertfordshire participated in the routing task. They manually selected sets of keywords using the topic descriptions and the adaptive training examples.

Microsoft Research Cambridge participated in the adaptive filtering and routing tasks. Their probabilistic Okapi/Keenbow system is very similar to that used in previous years, but the adaptive filtering component was rewritten for this year. The new filtering component allows updating of profiles and thresholds at each document retrieved. For routing, a new system using perceptrons with uneven margins was used.

Rutgers University participated in the adaptive and batch filtering tasks.[1] Their adaptive system is based on a Rocchio classifier and pseudo-relevance feedback. For batch filtering, they used rank-based feature

[1] David Lewis, part of the Rutgers group, submitted runs two adaptive filtering runs as a separate group. His results are presented in the Rutgers proceedings paper.

selection to identify a very small set of features to represent the collection and trained a simple classifier using these features.

State University of New York at Buffalo participated in the adaptive and batch filtering tasks. They used two main approaches, SVMs with weighted margins and language modeling.

CLIPS participated in the adaptive filtering task. Their RELIEFS system, introduced in TREC 9, is based on a probabilistic model of terms and relevance. This year, they focused on threshold adaptation and estimating relevance.

National Institute of Informatics participated in the batch filtering task. Their approach involved reweighting terms co-occurring in relevant training documents, and modeling these term sets as "virtual" relevant documents. They then used SVMs to learn a decision boundary based on the enlarged training set.

Clairvoyance Corporation participated in the batch filtering task. Their experiments focused on the performance of the monolithic filters which in their CLARIT system can be arranged to create ensemble filters. Their paper describes post-TREC experiments comparing their IR-based approaches to SVMs.

IRIT participated in all three tasks. Their Mercure system is based on a connectionist model. This year their experiments focused on threshold calibration.

Tampere University of Technology participated in the routing task. Their approach is based on word coding and characterizing the histograms of encoded documents.

Fudan University participated in the adaptive filtering task. They used the topic and training samples to create an initial Winnow classifier, and with that gathered a larger set of pseudo-relevant documents to further train the classifier.

Queens College, CUNY participated in the adaptive filtering task. They used a two-stage approach; initially, a simple profile reweighting and threshold adjustment scheme is used; but as more relevance information is available, the profile is expanded.

Chinese Academy of Sciences participated in all three tasks. Their experiments in adaptive filtering focused on making use of retrieved documents whose relevance is unknown in profile adaptation.

JHU/APL participated in all three tasks. For filtering, they used linear SVMs, with system parameters tuned using the TREC-8 filtering data. For routing, one run used SVMs and the other run merged the SVM run with an unsubmitted language modeling-based run.

Tsinghua University participated in the adaptive filtering task. Their incremental learning approach uses pseudo-relevance feedback to form the initial profile and threshold. They also experimented with a language modeling run using the Lemur toolkit.

University of Iowa participated in the adaptive filtering task. Their system uses two-level dynamic clustering. Documents placed into a topics first-level cluster are further divided into secondary clusters which are responsible for determining whether a document will be retrieved.

3.2 Evaluation Results

Some results are presented in the following graphs. Figures 1 and 2 show the adaptive filtering results for the utility and F measures. In each graph, the horizontal line inside a run's box is the median topic score, the box shows interquartile distance, the whiskers extend to the furthest topic within 1.5 times the interquartile distance, and the circles are outliers. In all graphs of T11SU scores, the horizontal line through the graph shows the baseline utility which can be achieved by retrieving no documents.

Figures 3 and 4 show the utility and F-beta results for batch filtering. Figure 5 shows mean uninterpolated average precision for routing.

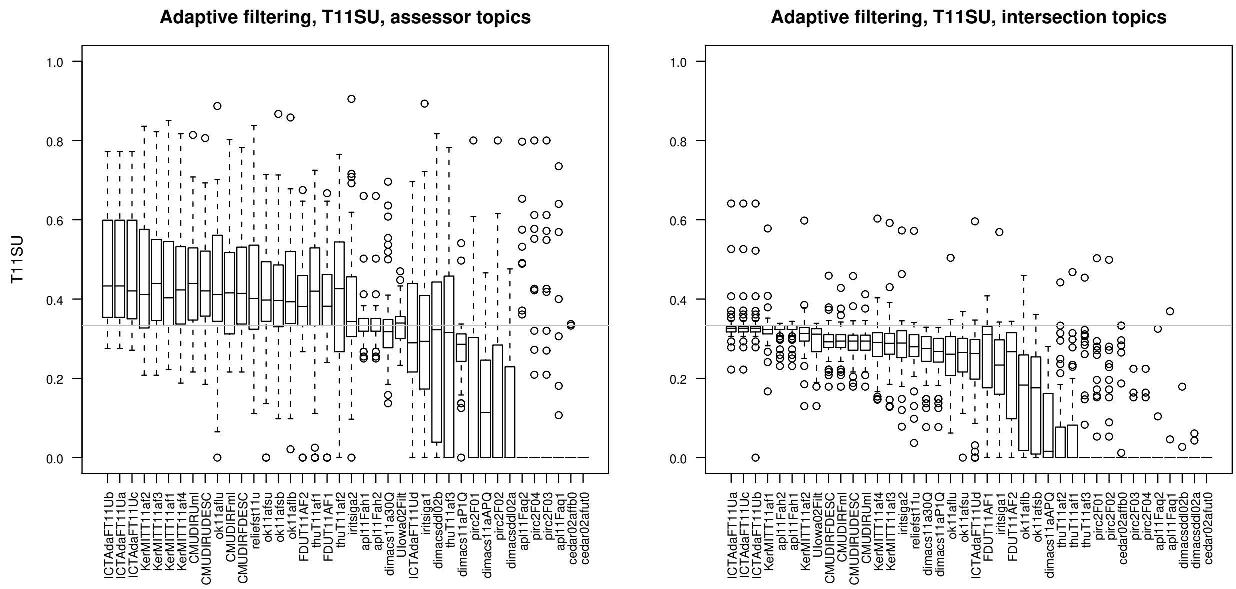

Figure 1: Adaptive filtering – T11SU

4 General Commentary

4.1 Post-submission judgements

Although more than 21,000 relevance judgements were made during topic creation and released with the topics, we were concerned that participants would still find more relevant documents. In order to make sure systems were measured fairly, NIST pooled participants' runs and judged any previously unjudged documents in the pool. Pooling was done as follows. Each participating group was allotted a fixed budget of documents to be pooled from their runs. If the group had any routing runs, we added unjudged documents from the top 100 ranks to the pool. If the group also had filtering runs, at most half the budget was expended on routing documents. We then merged all batch and adaptive filtering runs from that group and took a random sample of documents from the combined runs to fill out the pool budget. In all, another 42,000 documents were judged during this second round of assessment.

Figure 6 shows the numbers of relevant documents found for each topic in the first and second rounds of judging. Note that overall the topics have between 9 and 599 relevant documents apiece, much fewer than the TREC 2001 categories and closer to TREC ad hoc scale. For most topics only a few new relevant documents were found in the second round (median = 8.5), but seven topics had more than fifty new. Four of these topics had more than twenty new relevant documents found in their last round of feedback during the creation phase. Although our pooling process is radically different, these findings agree with Harman's analysis of the TREC-3 relevance judgements [1], as well as those of Zobel [11] that the "largest" topics (those with the most relevant documents) tend to yield even more relevant documents upon further searching. We have seen that such topics tend to have a greater number of relevant documents found in the last round of judging. In retrospect it probably would have been a good idea to discard these topics.

Another important factor is that five topics were judged by a different assessor in the second round than the one who had created it. Although as a general rule assessors always judged their own topics, due to time constraints we were forced to move these topics to different assessors. In these cases, the assessor was shown all of the relevant documents found in the first round as orientation to the topic. Four of these

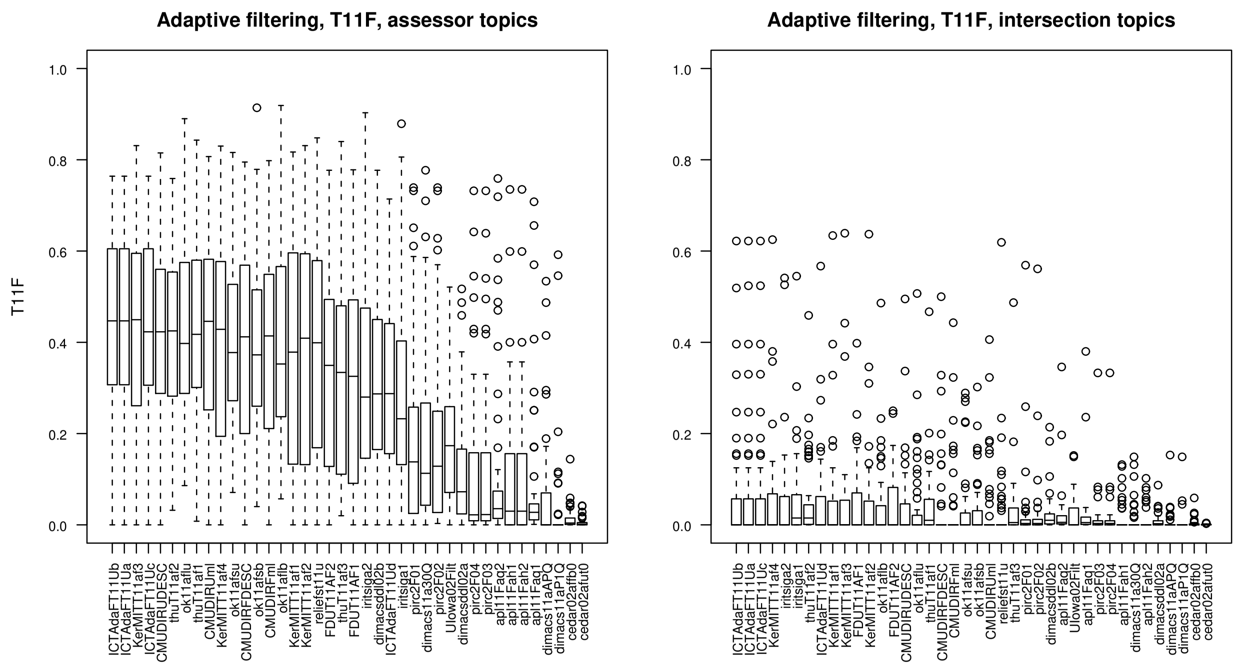

Figure 2: Adaptive filtering – T11F

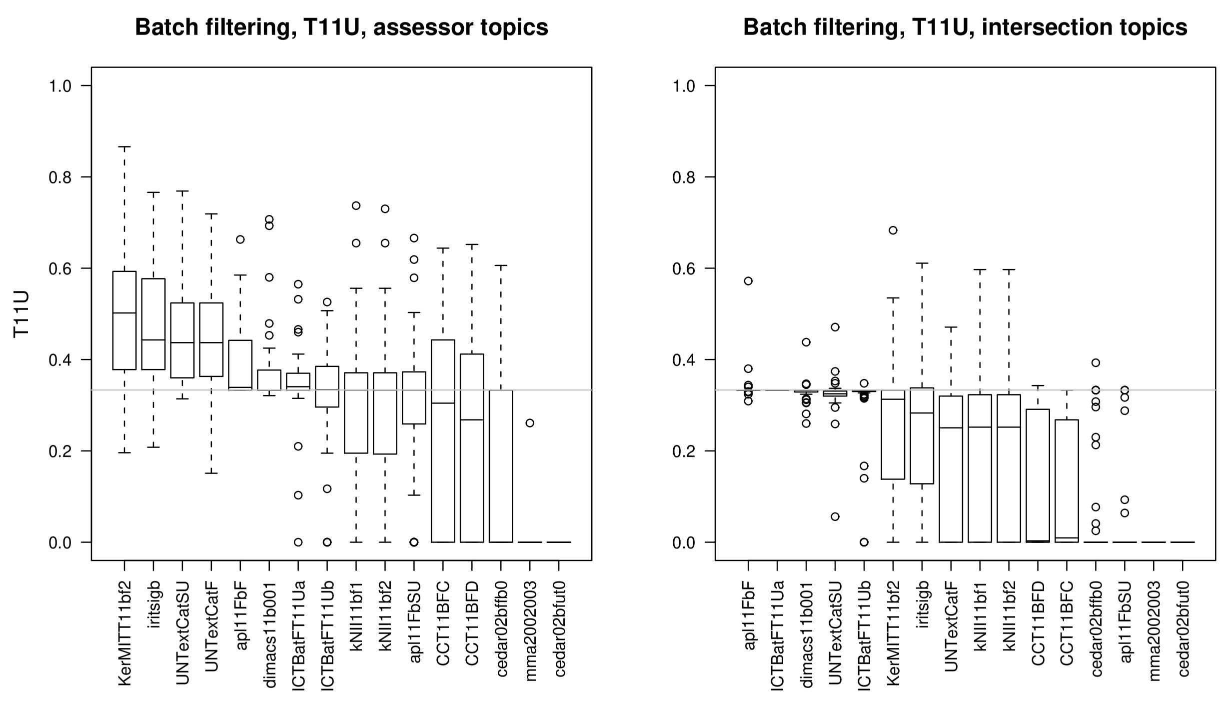

Figure 3: Batch filtering – T11SU

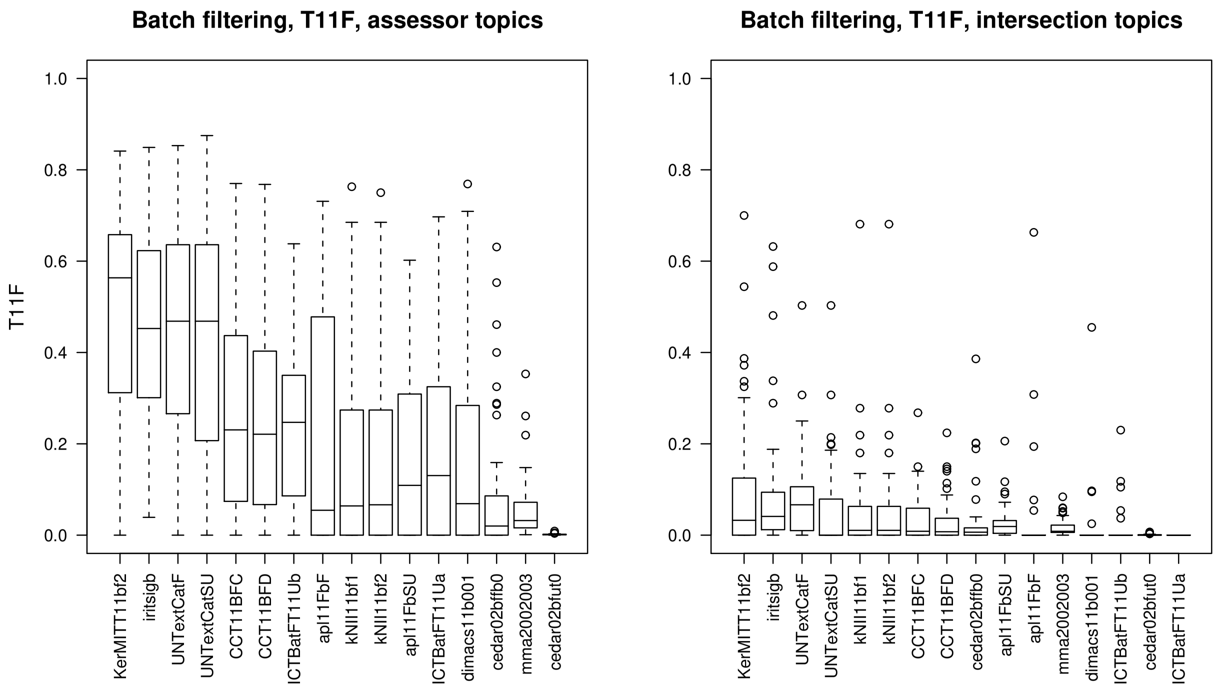

Figure 4: Batch filtering – T11F

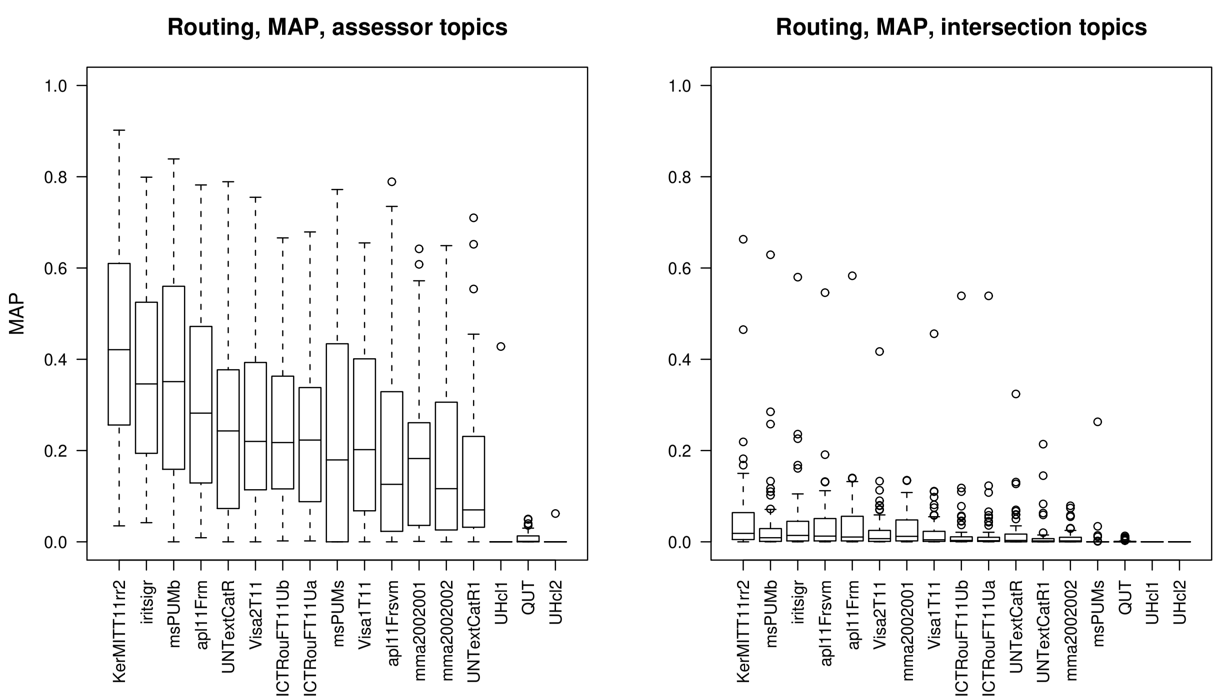

Figure 5: Routing – Mean Average Precision

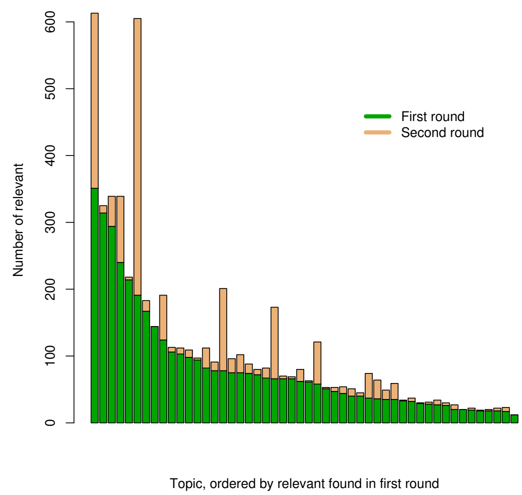

Figure 6: Relevant documents found in the first and second rounds of judging.

	T11U	T11F
Adaptive	0.969	0.936
Batch	0.996	0.983
Routing	0.912 (MAP)	

Table 1: Correlation of the official TREC results to a system ranking measured using the first-round relevance judgements only.

"moved" topics were also topics with more than fifty new relevant found, suggesting that these topics were not judged as well as the others.

4.2 Intersection topics

One issue this year concerned the experiment on the intersection method of building topics and making relevance judgements. This method would be considerably cheaper than the usual method involving assessors for both tasks: the process of making relevance judgements is a substantial effort. Our hope was that it would prove to be a viable alternative, providing a way of constructing test collections with much larger numbers of topics than we have at present, even if the quality is not quite as good.

In the event, the immediate impression of the intersection topics must be that they are not useful. The discrepancy in performance between assessor and intersection topics is huge. We might be tempted to hypothesise that the intersection topics are simply much harder than the assessor topics, but nevertheless represent a realistic task. However, it is hard to maintain that view in the light of the size of the discrepancy.

Possible hypotheses

Some hypotheses have been suggested (by participants and in subsequent discussion) for why the performance on the intersection topics was so poor. Roughly speaking, we may divide them into two classes: those which focus on the individual classes and those which focus on the intersection operation. In some cases at least, the hypothesis suggests experiments that may help to elucidate the problem; failure analysis on the official or other runs may also be informative. By and large these experiments and analyses have not yet been performed, although a few participants have done some failure analysis – they require some thought and effort, and being directed at a methodological question, they are not about specific systems, models or approaches, and therefore maybe of less immediate interest to participants. Nevertheless, the methodological question is of interest, and deserves investigation.

One hypothesis is that Reuters' assignment of category labels is simply too inconsistent, compared to assessor relevance judgements, to allow a system to learn adequately how to predict it. This hypothesis would suggest that the same problem would apply to topics defined as individual classes as to topics defined as intersections of pairs. However, the TREC 2001 experiment used individual classes and although many systems had significant difficulties, several systems performed adequately well on these topics. This would tend to suggest that the hypothesis as it stands is not a sufficient explanation.

A second is that Reuters' rules for category assignment specify that at least one category must be assigned to each document. Editors are happy if they can assign one category; extra ones are only assigned if (a) they immediately stand out as necessary, or (b) there is significant doubt about which is the correct one.[2] Either way, there is likely to be significantly more noise in second category assignment than in first category assignment, which will adversely affect the intersection topics. Experiments could be designed to substantiate this hypothesis.

A third is that categories may be of very different sizes; an intersection of a large category with a small one may be difficult to learn. A variant on this is qualitative rather than quantitative: some categories may be much harder to learn than others, and an intersection may be as hard as the harder of the two categories.

4.3 Overall performance

On the utility measure, most of the adaptive systems now outperform the baseline system which retrieves no documents ever. This is a welcome result. Furthermore, on the whole the adaptive systems are performing similarly to the batch filtering systems. In other words, despite starting from considerably less information they can through adaptation pull themselves up to a similar level overall. This suggests that at the end of the time period, they are likely to perform better than the batch 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, David Hull, Karen Sparck Jones, Chris Buckley, Paul Kantor, Ellen Voorhees, the TREC program committee, and the team at NIST.

References

[1] D K Harman. Overview of the Third Text REtrieval Conference (TREC–3). In D K Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC–3)*, pages 1–20. Gaithersburg, MD: NIST, 1994. NIST Special Publication 500-225.

[2] D A Hull. The TREC-6 filtering track: Description and analysis. In E M Voorhees and D K Harman, editors, *The Sixth Text REtrieval Conference (TREC–6)*, pages 45–68. Gaithersburg, MD: NIST, 1998. NIST Special Publication 500-240.

[3] D A Hull. The TREC-7 filtering track: Description and analysis. In E M Voorhees and D K Harman, editors, *The Seventh Text REtrieval Conference (TREC–7)*, pages 33–56. Gaithersburg, MD: NIST, 1999. NIST Special Publication 500-242.

[2] Suggested by David Lewis (private communication)

[4] D A Hull and S Robertson. The TREC-8 filtering track final report. In E M Voorhees and D K Harman, editors, *The Eighth Text REtrieval Conference (TREC–8)*, pages 35–56. Gaithersburg, MD: NIST, 2000. NIST Special Publication 500-246.

[5] D Lewis. The TREC–4 filtering track. In D K Harman, editor, *The Fourth Text REtrieval Conference (TREC–4)*, pages 165–180. Gaithersburg, MD: NIST, 1996. NIST Special Publication 500-236.

[6] D Lewis. The TREC-5 filtering track. In E M Voorhees and D K Harman, editors, *The Fifth Text REtrieval Conference (TREC–5)*, pages 75–96. Gaithersburg, MD: NIST, 1997. NIST Special Publication 500-238.

[7] Reuters corpus volume 1. http://about.reuters.com/researchandstandards/corpus/. Visited 26 September 2002.

[8] S Robertson and D A Hull. The TREC-9 filtering track final report. In E M Voorhees and D K Harman, editors, *The Ninth Text REtrieval Conference (TREC–9)*, pages 25–40. Gaithersburg, MD: NIST, 2001. NIST Special Publication 500-249.

[9] S Robertson and I Soboroff. The TREC 2001 filtering track report. In E M Voorhees and D K Harman, editors, *The Tenth Text REtrieval Conference, TREC 2001*, pages 26–37. Gaithersburg, MD: NIST, 2002. NIST Special Publication 500-250.

[10] I Soboroff and S Robertson. Building a Filtering Test Collection for TREC 2002. To appear in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '03).*

[11] J Zobel. How Reliable are the Results of Large-Scale Retrieval Experiments? In W B Croft, A Moffat, C J van Rijsbergen, R Wilkinson, and J Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, pages 307–314. ACM Press: Melbourne, Australia, August 1998.