Minimal Test Collections for Relevance Feedback*

Ben Carterette, Praveen Chandar, Aparna Kailasam, Divya Muppaneni, Lekha Thota
Department of Computer & Information Sciences
University of Delaware, Newark, DE, USA

The Information Retrieval Lab at the University of Delaware participated in the Relevance Feedback track at TREC 2009. We used only the Category B subset of the ClueWeb collection; our preprocessing and indexing steps are described in our paper on *ad hoc* and diversity runs [10].

The second year of the Relevance Feedback track focused on selection of documents for feedback. Our hypothesis is that documents that are good at distinguishing systems in terms of their effectiveness by mean average precision or some other evaluation measure will also be good documents for relevance feedback. Thus we have applied the document selection algorithm MTC (Minimal Test Collections) developed by Carterette et al. [6, 4, 9, 5] that is used in the Million Query Track [2, 1, 8] for selecting documents to be judged to find the right ranking of systems. Our approach can therefore be described as "MTC for Relevance Feedback".

1 MTC Overview

MTC is a greedy algorithm for selecting documents to be judged. It takes as input a set of relevance judgments (possibly empty) and a set of ranked lists of documents for a query or set of queries; as output it produces a set of "importance weights" for each unique document ranked for each query. The weights reflect the utility of the document for ranking the input lists by their mean average precision. After a document or set of documents has been judged, the algorithm can be run again with those judgments and the same input runs; the weights in computes will then be based on any existing judgments as well as the runs. Note that the weights do not necessarily have anything to do with relevance; any correlation between the weights and document relevance is unintended. In fact, since judgments of nonrelevance often say more about the difference in MAP between two systems than judgments of relevance, it is more likely that the weights negatively correlate to relevance.

MTC judgments can be used with `trec_eval`, making the assumption that unjudged documents are not relevant. This is not optimal, however; instead, the MTC method uses the judgments to fit a classifier, which it then uses to predict the relevance of unjudged documents. These predicted relevance judgments are then used to calculate expected values of MAP; the maximum-likelihood ranking of systems is the one by expected MAP.

2 MTC for Relevance Feedback

2.1 Phase 1: Selecting Documents for Feedback

The traditional approach to relevance feedback is that a user provides feedback on some of the top documents retrieved by some system; that system then uses that feedback to rerank documents (often by expanding the original query). The key difference for the MTC approach is that there are a *set* of systems from which a few documents are selected to ask for feedback. After receiving feedback, the documents will be reranked.

Therefore the first step in applying MTC is to generate several different ranked lists for each query. Using the Indri retrieval system, we applied the following methods, each generating a different ranked list:

*Notebook version.

1. basic query-likelihood language modeling with Dirichlet smoothing;
2. the dependence model of Metzler & Croft [13];
3. pseudo-relevance feedback with external query expansion (top 10 documents retrieved by Google for the same query) [11];
4. maximum marginal relevance ranking [3];
5. greedy similarity-based pruning, as described by Carterette & Chandar [7] (also used in the diversity task for the Web track [10]);
6. various automatically-generated queries using Indri operators like `#uwN, #odN, #weight`;
7. weighting the appearance of query terms in `title` and `heading` fields higher than the `body` field.

In all, 11 ranked lists for each query were used as input to MTC. Note that the pseudo-feedback and other re-ranking approaches were used to generate some of these; our RF track experiments are completely separate from such fully-automatic approaches.

MTC produced a weight for each document retrieved by all 11 runs. The top 5 highest-weighted documents were selected to be judged in Phase 1 of the track (the `udel.1` Phase 1 submission; our `udel.2` submission did not use MTC and is described in our paper on `ad hoc` and diversity [10]).

2.2 Phase 2: Query Expansion and Re-ranking

Within the MTC framework, there are several possible ways to use the Phase 1 judgments for relevance feedback:

1. use them to evaluate the input lists, then choose the top-performing list as the final ranking;
2. use them to train a relevance classifier, then use the predictions of relevance from that classifier to rank documents in decreasing order of relevance, with this ranking being the final ranking shown to the user;
3. use them to train a relevance classifier, then use the predictions of relevance from that classifier to expand the original query; re-rank documents using the expanded query.

The structure of the RF track required that we choose one of these. Based on some limited experiments with previous years' TREC data, we chose the third approach for our official submissions. In Section 3 we will evaluate all three, so we will describe each in more detail here.

Each of these approaches can be applied to any set of feedback judgments. We did each approach for each set of Phase 1 judgments we received.

2.2.1 Selecting the Top-Performing Ranking

This is a very straightforward application of MTC: use the judgments and the MTC evaluation to rank the input systems, then choose the best one as the final ranking. In some sense this is using relevance feedback to select among possible models/ranking algorithms rather than perform any query expansion or reranking.

2.2.2 Ranking Documents by Probability of Relevance

As described above, MTC requires probabilities of relevance to evaluate systems. We could use these probabilities to directly rank documents. If our classifier is any good, the new ranking should perform well compared to any input ranking. In practice, MTC does not actually require good estimates of relevance in order to produce good rankings of systems (they only need be "good enough", and MTC will tolerate a lot of error in these estimates), and therefore we cannot necessarily expect the classifier to be very good.

Our classifier is a logistic linear model. We use features extracted from the input runs reflecting how well they were able to rank the judged documents; the procedure is described in detail by Carterette [4].

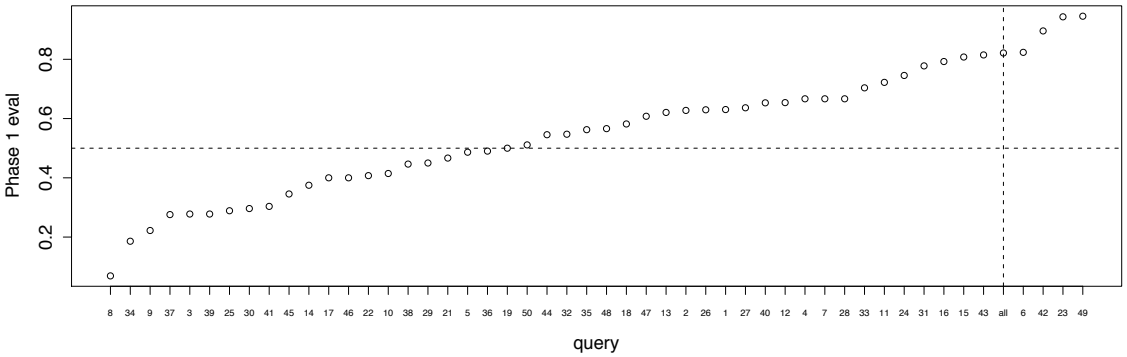

Figure 1: `udel.1` performance on each of the first 50 queries and over all queries. The vertical dashed line shows performance over all queries. The horizontal line is at 50% performance, where `udel.1` is the median Phase 1 set. We were better than the median in 58% of queries.

2.2.3 Using Probabilities of Relevance for Query Expansion

Instead of ranking documents directly by probabilities of relevance, knowing that they are quite errorful, we can instead use them to do some query expansion. In some sense this results in averaging over a large number of documents and thus if our classifier is "good enough" it may provide a decent expanded query.

We used the same classifiers and features as in the previous section to get probabilities of relevance for unjudged documents. Documents were ranked by these probabilities (filling in judgments from Phase 1 if known); this ranking was then used to estimate a relevance model as described by Lavrenko & Croft [12]. A relevance model gives a probability to each term in the top ranked documents; these probabilities are based on the term frequencies, collection frequencies, and probabilities of relevance of those documents. This model is then used to re-rank the collection to produce a final ranking.

3 Results

3.1 Phase 1: Document Selection

The official Phase 1 evaluation is based on comparing a set of Phase 1 judgments to other Phase 1 judgments used as input to the same Phase 2 approach. For example, in Phase 2 we used Phase 1 judgments `SIEL.1`, `Sab.1`, `UCSC.1`, `WatS.1`, `fub.1`, `twen.1`, `udel.1`, and `udel.2`. The effectiveness of `udel.1` is the ratio of the number of these Phase 1 sets that resulted in worse Phase 2 results than `udel.1` to the total number of Phase 2 results (removing ties).

Our Phase 1 submission `udel.1` based on MTC selection from 11 (automatic) input runs outperformed 82.14% of the other Phase 1 submissions used for the same Phase 2 approach (aggregating over different Phase 2 approaches and averaging over queries). Performance on individual queries is shown in Figure 1.

It may be worth looking at the proportion of Phase 1 documents found to be relevant. In our case, 39.6% of the documents selected by MTC were judged relevant. This was by no means the greatest of any Phase 1 submission; we ranked 9th among all Phase 1 submission by this measure. Of course, the goal of MTC is not to select *relevant* documents, but to select documents that are good at distinguishing between systems.

3.2 Phase 2: Relevance Feedback

We were assigned eight Phase 1 sets to apply our relevance feedback approaches to: `SIEL.1`, `Sab.1`, `UCSC.1`, `WatS.1`, `fub.1`, `twen.1`, `udel.1`, and `udel.2`. For each of these sets, our official Phase 2 submission was

Phase 1 set	statMAP	eMAP	eMAP (unofficial)
Sab.1	0.1092	0.0328	0.0593
SIEL.1	0.1311	0.0355	0.0686
WatS.1	0.1387	0.0367	0.0748
twen.1	0.1443	0.0383	0.0744
udel.2	0.1480	0.0350	0.0729
base	0.1689	**0.0421**	0.0638
fub.1	0.1702	0.0377	0.0828
UCSC.1	0.1720	0.0382	0.0816
udel.1	**0.1762**	0.0393	**0.0838**

Table 1: Official Phase 2 evaluation results for each Phase 1 input (sorted by statMAP). Bold text indicates the best result in the column. The last column shows unofficial results using the Million Query erels.

based on the third approach described in Section 2.2: train a relevance classifier using the judgments in the set, use that classifier to predict the relevance of the unjudged documents, and use those predictions (along with any judgments in the set) to estimate a relevance model (a weighted expanded query). The final ranking is then based on ranking the collection to the relevance model query. As a baseline, we used straight pseudo-feedback relevance modeling; no judgments were used.

Since we used the category B subset, the official evaluation measures are statMAP (a low-bias estimate of MAP based on a sample of documents) and expected MAP (described above). Table 1 shows results for our official submissions, comparing our feedback approach with different Phase 1 inputs. Though the two measures disagree on the ranking, they agree that our `udel.1` submission provided better Phase 2 results than any other set, suggesting that MTC selection is the best way to select documents if an MTC-like approach is to be used in reranking (though it is unlikely these differences are significant). Note, however, that by the official eMAP scores, none of the Phase 1 sets outperformed blind feedback. We included a second set of eMAP estimates that show `udel.1` again outperforming the rest; with no way to know which is most correct, it seems safe to conclude that the `udel.1` judgments are better than the rest.

4 Additional Results

Here we present some additional analysis and results outside of the official track results.

4.1 Confidence in Pairwise Differences

The eMAP evaluation can be used to estimate the degree of confidence in pairwise differences between systems. These confidence scores can give us some idea of how "definite" the ranking is: low confidences (near 0.5) indicate that more relevance judgments could possible cause the two systems to swap; high confidences (near 1.0) indicate that the systems are unlikely to swap even with more judgments. Table 2 shows the confidences in the difference in eMAP for all pairs of runs.

Because the runs have been residualized (Phase 1 judgments removed), these confidence scores should be taken with a grain of salt. Nevertheless, they provide a rough guide to interpreting the eMAP scores.

4.2 Alternative Approaches to Feedback

In section 2.2 we described three possible approaches to relevance feedback that fit within the general MTC framework. Table 3 shows a comparison of these different approaches with the different Phase 1 sets used as input. Note that the best input was the external expansion run in all but two cases; the only reason the evaluation results are different is slight differences in which Phase 1 documents were removed before evaluation. Though the ranking of Phase 1 sets by statMAP and eMAP is roughly the same in both the probability ranking and RM methods, the RM method achieved much better results. We conclude from

run	udel.2	SIEL.1	WatS.1	fub.1	UCSC.1	twen.1	udel.1	base
Sab.1	0.8078	0.9216	0.9349	0.9737	0.9789	0.9836	0.9939	0.9998
udel.2		0.5638	0.7813	0.8376	0.8783	0.9594	0.9685	0.9930
SIEL.1			0.6766	0.8128	0.8425	0.8601	0.9257	0.9905
WatS.1				0.6458	0.6975	0.7895	0.8505	0.9696
fub.1					0.5892	0.5800	0.7563	0.9177
UCSC.1						0.5239	0.7082	0.8960
twen.1							0.6539	0.9112
udel.1								0.8278

Table 2: Confidences between Phase 2 runs with different Phase 1 input sets. Each value is the probability that the corresponding two runs would swap in the ranking if more relevance judgments were available.

this that RM does have the ability to "improve" the probability ranking, partly by taking into account the original query, and partly by averaging over all documents.

5 Conclusions

Our conclusion actually relates more to the MTC approach to selecting documents to judge than to relevance feedback: our results show that a small number of MTC selections are better for training a relevance model than any other selection process we compared to. This may have to do with the active-learning "feel" of MTC. Of course, in this case we would have been better off simply using MTC to select the best input run, but because (a) that run is a web expansion run and is thus capitalizing on years of research by Google without providing any insight into retrieval, and (b) that would not have been interesting from a research perspective, we did not do that.

References

[1] James Allan, Javed A. Aslam, Ben Carterette, Virgil Pavlu, and Evangelos Kanoulas. Million Query Track 2008 overview. In *Proceedings of TREC*, 2008.

[2] James Allan, Ben Carterette, Javed A. Aslam, Vrigil Pavlu, Blagovest Dachev, and Evangelos Kanoulas. Million Query Track 2007 overview. In *Proceedings of TREC*, 2007.

[3] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, pages 335–336, 1998.

[4] Ben Carterette. Robust test collections for retrieval evaluation. In *Proceedings of SIGIR*, 2007.

[5] Ben Carterette. *Low-Cost and Robust Evaluation of Information Retrieval Systems*. PhD thesis, University of Massachusetts Amherst, 2008.

[6] Ben Carterette, James Allan, and Ramesh K. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of SIGIR*, pages 268–275, 2006.

[7] Ben Carterette and Praveen Chandar. Probabilistic models of novel document ranking for faceted topic retrieval. In *Proceedings of CIKM*, 2009.

[8] Ben Carterette, Virgil Pavlu, Hui Fang, and Evangelos Kanoulas. Million Query Track 2009 overview. In *Notebook Proceedings of TREC*, 2009.

[9] Ben Carterette and Mark D. Smucker. Hypothesis testing with incomplete relevance judgments. In *Proceedings of CIKM*, pages 643–652, 2007.

Phase 1 set	best input	prob ranking	prob + RM
Sab.1	0.2285	0.0666	0.1092
	0.0473	0.0118	0.0328
SIEL.1	**0.2300**	0.0737	0.1311
	0.0476	0.0126	0.0355
WatS.1	0.2251	0.1180	0.1387
	0.0475	0.0137	0.0367
twen.1	0.1044	0.1129	0.1443
	0.0135	0.0136	0.0383
udel.2	0.0983	0.1193	0.1480
	0.0131	0.0141	0.0350
base	0.1689	**0.1689**	0.1689
	0.0421	**0.0421**	**0.0421**
fub.1	0.2260	0.1201	0.1702
	0.0473	0.0146	0.0377
UCSC.1	0.2228	0.1407	0.1720
	0.0467	0.0147	0.0382
udel.1	0.2218	0.1441	**0.1762**
	0.0470	0.0148	0.0393

Table 3: Different feedback approaches using different Phase 1 inputs. Each cell contains statMAP and eMAP for the pair. The last column contains our official submission results. Bold text indicates the best statMAP and eMAP result in each column.

[10] Praveen Chandar, Aparna Kailasam, Divya Muppaneni, Lekha Thota, and Ben Carterette. Ad hoc and diversity retrieval at the university of delaware. In *Notebook Proceedings of TREC*, 2009.

[11] Fernando Diaz and Donald Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, 2006.

[12] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, 2001.

[13] Donald Metzler and W. Bruce Croft. A markov random field for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR)*, pages 472–479, 2005.