Using Relevance to Train a Linear Mixture of Experts

Christopher C. Vogt, Garrison W. Cottrell,
Richard K. Belew, Brian T. Bartell
University of California, San Diego
Computer Science and Engineering 0114
La Jolla, CA 92093
{vogt,gary,rik}@cs.ucsd.edu
bbartell@verity.com

Abstract

A linear mixture of experts is used to combine three standard IR systems. The parameters for the mixture are determined automatically through training on document relevance assessments via optimization of a rank-order statistic which is empirically correlated with average precision. The mixture improves performance in some cases and degrades it in others, with the degradations possibly due to training techniques, model strength, and poor performance of the individual experts.

1 INTRODUCTION

The mixture of experts approach is one which is gaining in popularity in many areas of computer science and artificial intelligence (e.g., [Jordan and Jacobs, 1994]) and one which is especially applicable to information retrieval, since in practice the sets of relevant documents returned by different IR algorithms (or *experts*) often have little overlap. In fact, the pooling method used by past TREC's to determine which documents are relevant can be viewed as a sort of mixture model on the grandest of scales [Harman, 1995b]. Each organization represents an expert, and only the top-rated documents from each are passed along to the human judges. The most general mixture model is extremely flexible since it can incorporate any number of experts. Thus it can subsume any other approach by merely including it as another expert.

The two main difficulties with mixture models are first determining the class of models to use and then finding those model parameters which maximize the performance of the system. Bartell, Cottrell, and Belew have explored both of these issues in some depth - focusing primarily on both linear and nonlinear neural net models coupled with the use of optimization of rank-order statistics to determine model parameters [Bartell et al., 1994b, Bartell et al., 1994a]. Even with the simplest linear combination of experts, they achieved some impressive improvements - up to 47% higher average precision than the best individual expert. All of their experiments, however, are on relatively small collections. Others have successfully used mixture (a.k.a. fusion) approaches on larger collections, including TREC, but they hand pick the model parameters, clearly an undesirable approach (see [Kantor, 1995], [Knaus et al., 1995], [Shaw and Fox, 1995] in [Harman, 1995a]). Here we show how the mixture technique coupled with automatic parameter adjustment via rank-order statistic optimization scales up to the TREC collection. Our results indicate that we have yet to find the best class of model for this task.

2 METHOD

2.1 The TREC Tasks

The Text REtrieval Conference (TREC) has two main tracks: adhoc and routing. Furthermore, participants may choose to take part in one of three categories, category A (all of the data) category B (a subset of the data), or category C (for companies who wish only to submit results, and not a paper). We participated in both tracks, category B. In both tasks, participants are given a training set of documents and queries, along with relevance assessments. For the adhoc task, a new set of queries are then distributed and the task is to find those training documents which satisfy the new queries. The routing task addresses the converse problem: a subset of the training queries are selected and the goal is to find relevant documents from a new collection. For category B, the training set consisted of 74,520 Wall Street Journal articles (about 253Mb of data) with an average of about 300 terms per document, and 250 queries of varying lengths (from 8 to 180 terms, average of about 20). The adhoc task added 50 new queries with average length of 83 terms (16 for the short version). The average number of relevant documents per adhoc query was about 24, although some queries had no relevant documents. For the routing task, 61,578 Foreign Broadcast Information Service documents were used (about 225Mb). For the 50 training queries selected from the training set as routing queries, the average length was 83 terms, and the average number of relevant documents was 63.

2.2 The Experts

Our approach was to use three experts, two based on the standard Vector Space model [Salton and Buckley, 1988] and one based on Latent Semantic Indexing [Deerwester et al., 1990]. The two VS models were implemented via Cornell's SMART system [Salton, 1971]. Two very different weighting schemes were used:

one was binary and the other was logarithmically scaled tf-idf. The SMART codes for these two weighting schemes are "bnn" and "ltc" respectively, and we will refer to them using those codes throughout this paper. Since we participated as category B, only those 74,520 Wall Street Journal articles on the second TIPSTER disk were indexed for training and testing the adhoc task, and only the 61,578 FBIS articles were indexed for testing the routing task. Within these documents, only those sections delimited by <TEXT>, <LP>, <HL>, and <IN> were indexed, and were stopped and stemmed using the default SMART routines (with no special treatment for proper nouns). This produced document vectors with 104,113 (WSJ/adhoc) and 188,142 (FBIS/routing) components.

Queries were processed in a similar but slightly more complicated manner. Each topic was first lexically analyzed, and those phrases (as delimited by '.',';','unless', and 'except') which contained the string "not" followed by "relevant," "about," or "sufficient" were removed. All other non-SGML text was used, except those terms not found in any training document. The resulting test was indexed using SMART and both bnn and ltc weightings. Furthermore, the adhoc queries were indexed twice, once using the all of the parsed text of the topics, and once using only the <DESC> field (the so-called "short topics").

The LSI expert mimicked the approach used by Dumais in TREC-3 [Dumais, 1995]. Specifically, since doing a Singular Value Decomposition on the full $104,113 \times 74,520$ term-by-document matrix from the ltc expert would have taken much too long, it was subsampled. Only those terms occurring in 5 or more documents were used, and only a randomly selected subset of about 10% of the documents was used, leaving a $26,395 \times 7500$ matrix which was reduced down to 300×7500 via SVD using SVDPACKC [Berry, 1992]. This produced a 300 dimensional representation for all the WSJ documents, and the corresponding representations for the queries and FBIS documents were obtained by first removing any terms that were not in the reduced WSJ vocabulary, and then projecting the resulting 26,395-vector down to 300 dimensions as described in [Dumais, 1995].

Once document and query vectors were fixed, relevance scores were computed using the standard inner product rule.

2.3 Combining Scores

Bartell showed that one very effective measure of how well an IR system performs is one which compares the rank-ordering produced by the system to that specified by relevance feedback [Bartell, 1994]. This is in contrast to one which attempts to reproduce the user's relevance scores exactly. Specifically, Bartell defines a criterion (hereafter called J) based on Guttman's Point Alienation statistic as follows.

Defn: the ranking function implemented by an IR system is

$$\begin{gathered} R : \Theta \times D \times Q \to \Re \\ \text{where} \\ \Theta = \text{the set of system parameters} \\ D = \text{the set of document vectors} \\ Q = \text{the set of query vectors} \end{gathered}$$

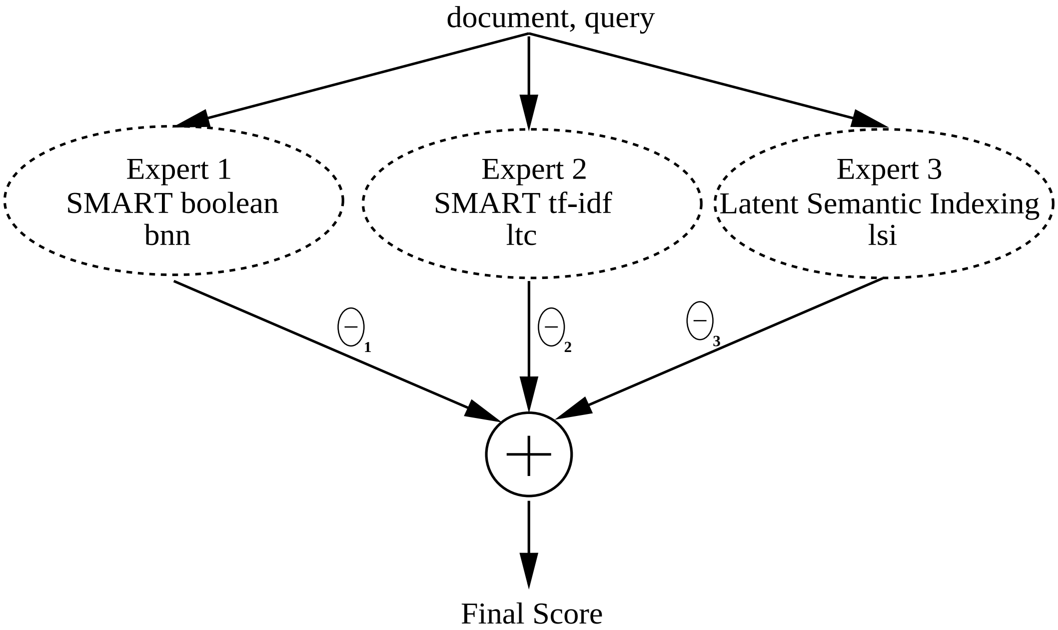

Figure 1: Linear Combination of Relevance Scores

Defn: Bartell's J criterion is:

$$J(R_\theta) = \frac{1}{|Q|} \sum_{q \in Q} \frac{\sum_{d \succ_q d'} (R(\theta, d, q) - R(\theta, d', q))}{\sum_{d \succ_q d'} |R(\theta, d, q) - R(\theta, d', q)|}$$

where $d \succ_q d'$ indicates document d is preferred to document d' on query q.

Note that J has a maximum value of 1 when the numerator and denominator are the same (i.e., the IR system ranks documents exactly as the user would), and a minimum value of -1 when the opposite is true.

Since Bartell rarely did better by using nonlinear models, and because such models tend to have a much larger set of parameters Θ, we chose to use a very simple linear mixture model, namely:

$$R_{mix} = \theta_{bnn} R_{bnn} + \theta_{ltc} R_{ltc} + \theta_{lsi} R_{lsi} \tag{1}$$

which has only three parameters. This is illustrated in Figure 1 Furthermore, other work by Bartell [Bartell, 1994] showed that if only the 15 top-ranked documents for each query (as determined by one of the experts) are used when optimizing J, *better* results are actually achieved. Thus, when we trained our model, we initially used only the top 15 documents (according to the ltc expert). The J criterion was maximized using a simple hill-climbing algorithm with multi-start. Starting from 5 different points in Θ space always yielded a $J \approx 0.59$, and more importantly, the Θ vectors at the maxima were all multiples of each other. Thus, with such a simple model, the J surface appears to have a set of global maxima, and the simple-minded hill-climbing approach to optimization is reasonable.

Expert	Mean (StdDev) for R_{expert}	θ_{expert}	Mean $R \times \theta$	% of total
bnn	10.6 (8.2)	1.0000	10.6	46
ltc	0.21 (0.076)	24.855	5.14	23
lsi	0.68 (0.095)	10.377	7.04	31

Table 1: Parameters and Relative Weightings for Training on the Top 15 Documents

Expert	Mean (StdDev) for R_{expert}	θ_{expert}	Mean $R \times \theta$	% of total
bnn	9.7 (7.4)	0.049	0.48	26
ltc	0.15 (0.062)	2.769	0.43	23
lsi	0.65 (0.086)	1.431	0.93	51

Table 2: Parameters and Relative Weightings for Training on the Top 100 Documents

Unfortunately, training on the top 15 documents for each query led to counter-intuitive results: the bnn expert was weighted more heavily than the other two. This can be seen in Table 1, where the bnn expert gets twice the weight of the ltc expert and 50% more than the lsi expert. This does not seem reasonable since the variance in the scores of the bnn expert is very high, suggesting it may be an unreliable judge of relevance.

Since Bartell's experiments with the top 15 documents were performed on collections an order of magnitude or smaller than the WSJ collection, we increased the number of documents used for training to the top 100 for each query. Once again, multi-start optimization always led to the same maximum of $J \approx 0.66$ and parameter vectors at the maxima were multiples of each other. These results, summarized in Table 2, show a more reasonable combination, with the lsi expert counting for half of the overall ranking score and the other two splitting the remainder nearly equally.

The three θ_{expert} values from Table 2 were used to combine the relevance scores of the three experts according to equation 1. The top 1000 ranked documents were submitted for three runs: one in which the routing queries were run against the FBIS documents, one where the full adhoc queries were run against the WSJ documents, and a third which ran the short-topic versions of the adhoc queries against WSJ.

3 RESULTS

Precision/Recall curves for each expert and the mixture determined by training on the top 100 documents are shown below for the three runs. Also included are the performance curves for the individual experts on those documents used for training (top 100) and those available for training, but which were not used to select Θ. The latter run is included to explore how well our technique of training on the top n

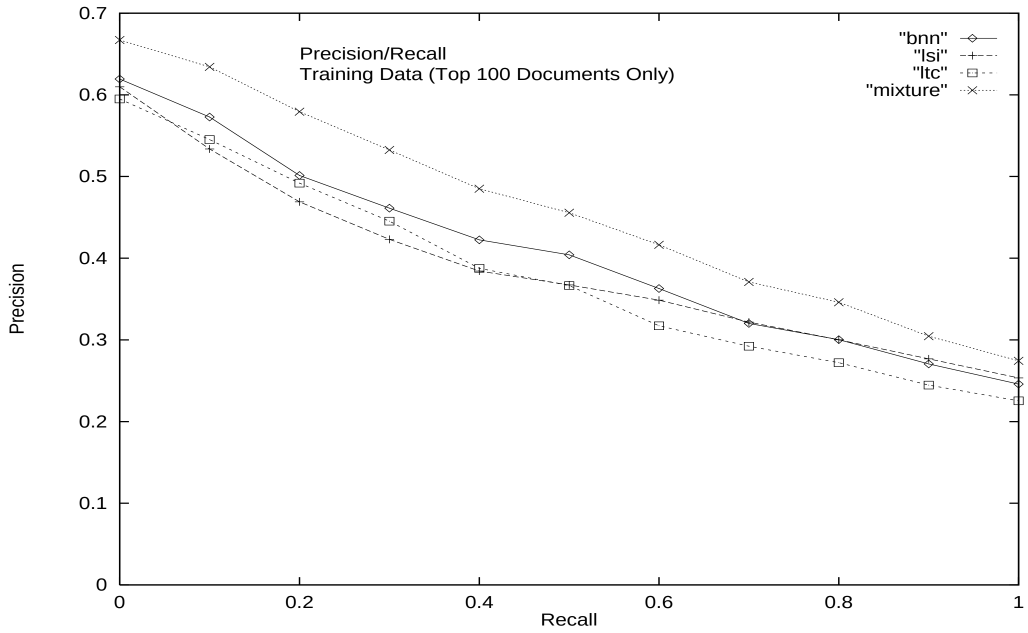

Figure 2: Performance on the Actual Training Data (Top 100 Documents)

documents generalizes to the entire available training set.

A couple of predictable trends are evident in the results. First, the short-topic version of the adhoc task performs worse than the version which uses all topic text. Second, as has been the case in past TREC's, the routing results are generally better than the adhoc ones.

Surprisingly, the lsi expert does not generally outperform the ltc expert on any of the runs. This is in contrast to what Dumais, et al. have found in past TREC's. However, this is not critical for the ideas we are exploring with this paper, since in theory any set of experts may be used regardless of their performance.

The interesting part of these results comes in comparing the mixture's performance with those of the individual experts. Ironically, this shows mixed results. As expected, the mixture performs significantly better than any individual expert on the actual training data (top 100 documents) (see Figure 2). This is expected because previous experiments have suggested that J is highly correlated with average precision, so by optimizing J, we are also optimizing precision. However, when the same mixture is applied to the entire available training set, it only does as well as the second-best expert (lsi) (Figure 3). Thus, generalization is poor to the entire training set, indicating that we may have overtrained, that our model is not powerful enough, or that the top 100 documents do not accurately represent the entire data set. The first explanation seems the least plausible, since there are only 3 parameters in our model. The second two are ideas we will have to explore in future TREC's. On a more positive note, for both versions of the adhoc task,

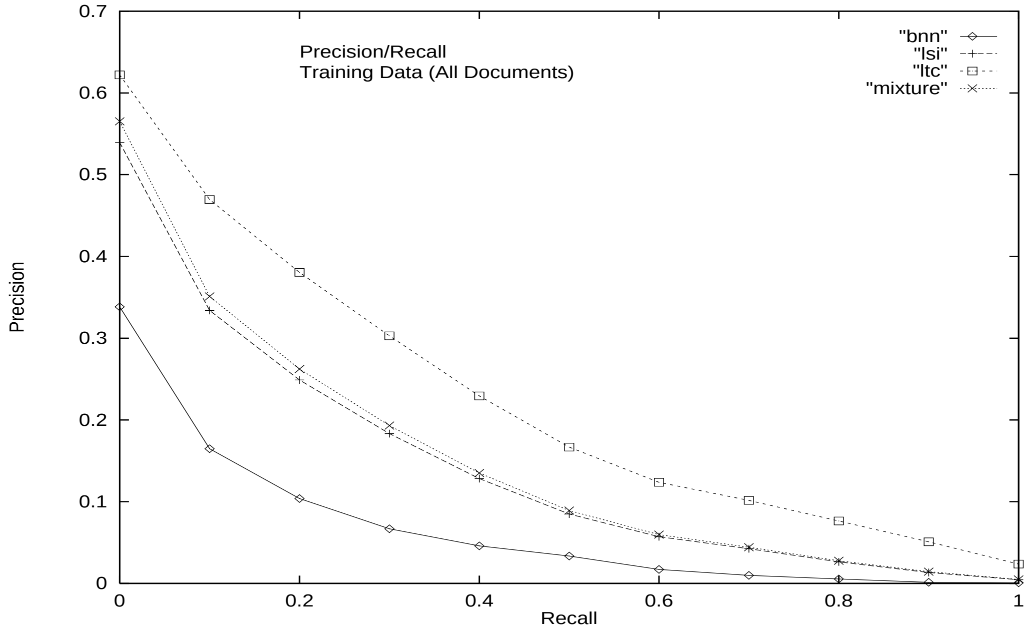

Figure 3: Performance on the Available Training Data (All Documents)

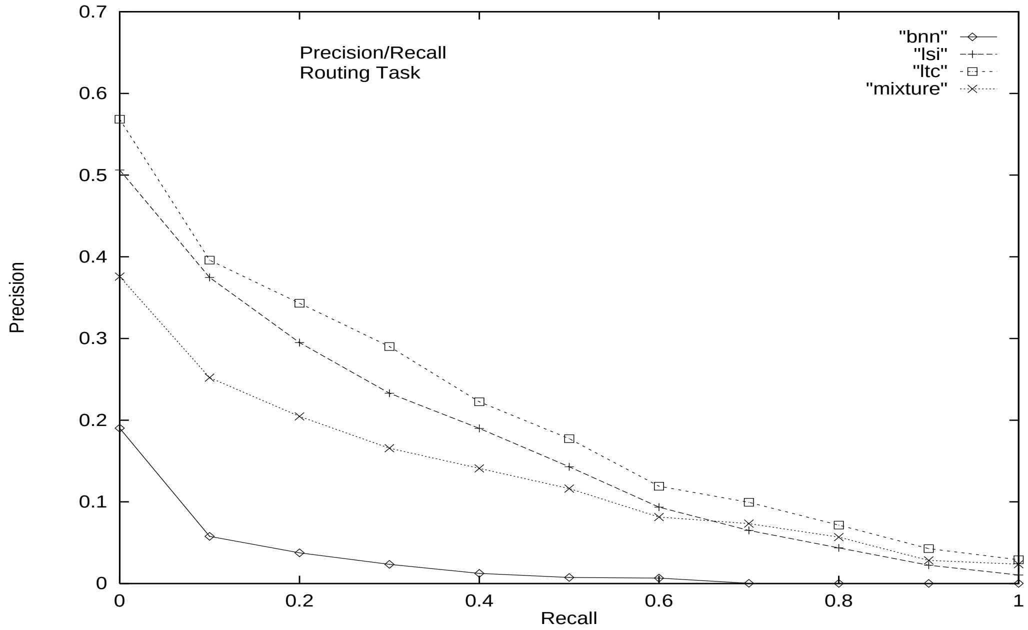

Figure 4: Performance on the Routing Task

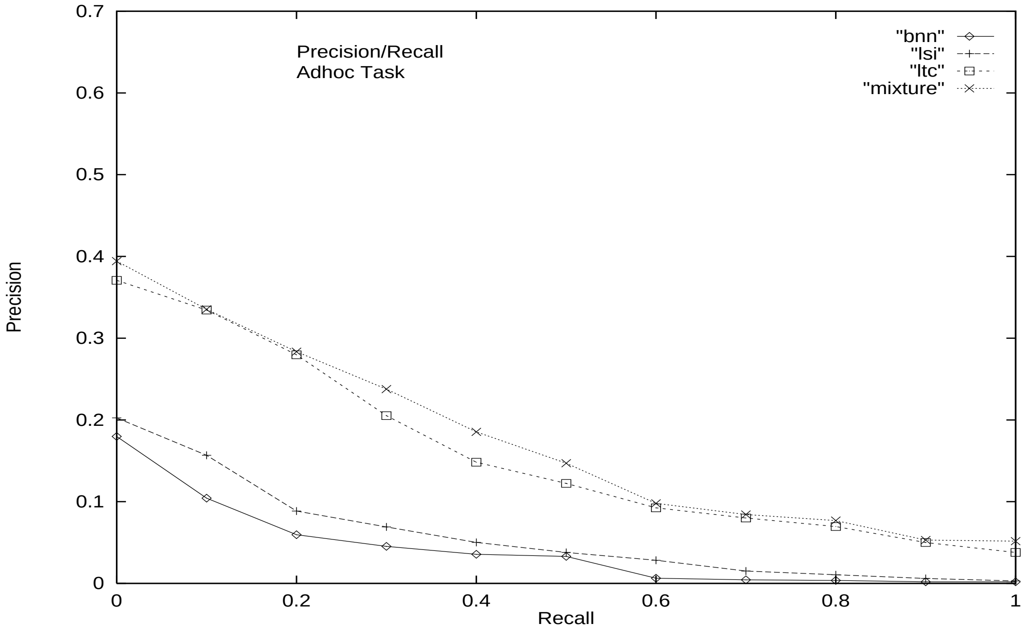

Figure 5: Performance on the Adhoc Task

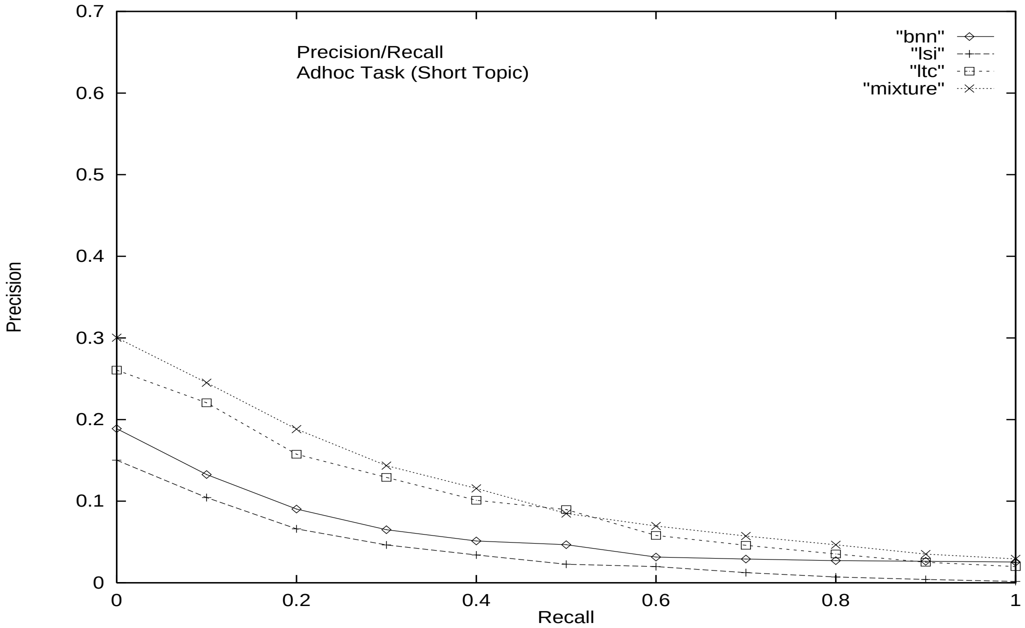

Figure 6: Performance on the Adhoc Task (short-topic)

the mixture outperforms *all* of the individual experts (Figures 5,6). However, for the routing task, the mixture model significantly underperforms two of the three experts (Figure 4).

4 SIMULATIONS

In order to better understand our model and when it can produce improved performance, we performed a series of simulations. First, 1000 pairs of "experts" were generated. An "expert" in this case is simply a list of 1000 documents for which scores have been randomly generated according to a normal distribution. For each pair of experts, we found an approximation to the "best" linear combination of the two by doing a raster scan of both weights in $(0, 1]$ in increments of one tenth, and taking the combination which had the highest average precision when 13 documents were randomly chosen as relevant.

Since our goal was to be able to predict *when* linearly combining the scores of IR systems can result in improvement, we collected some metrics about each pair of experts and used these as the independent variables in a linear regression with "percentage improvement of average precision over the better expert" as the dependent variable. The metrics we chose included: average precision for both experts (p_1, p_2), J for both experts (J_1, J_2), the Guttman's Point Alienation between the two experts (GPA) [1], and the GPA using only the relevant documents (GPA_r). The regression showed that by using only two of the variables we can account for over 80% of the variance in the improvement scores. Those two variables are J_2 (J for the worse of the two experts) and GPA_r (GPA calculated using only relevant documents), with J_2 being weighted slightly more. The importance of both metrics seems intuitively correct: since J_2 is one measure of how well the worse expert performs, it makes sense that an improvement in the better expert would be limited if J_2 was low. Likewise, since GPA_r is a measure of how similar the two experts rank the relevant documents, combining two experts with a very low GPA_r would result in a more random ranking.

5 DISCUSSION

In light of the above simulations, Table 3 shows GPA_r and J_2 for the TREC tasks for which we presented results above. Since the simulations were done on pairs of experts, each possible pairing is shown for each task. Recall that in the simulations, we used GPA_r and J_2 to predict the *best possible* improvement. In the table, we show what the linear regression predicts as the best possible improvement for the actual experts we used for the TREC tasks. Note that for almost every possible pairing in each task, a positive improvement is possible *except* in the routing task.

[1] The GPA can be calculated for any two lists of scores x and y as:

$$GPA = \frac{\sum_i \sum_j (x_i - x_j)(y_i - y_j)}{\sum_i \sum_j |x_i - x_j||y_i - y_j|}$$

GPA is a measure of how similar two rankings are to each other. Note that J is simply the GPA between an IR system and a user's relevance judgements, averaged over all queries.

Task	Pair of Experts	GPA_r	J_2	Best Possible Improvement (%)
Training (top 100)	lsi-bnn	0.20	0.35	8.8
Training (top 100)	ltc-bnn	0.11	0.32	7.5
Training (top 100)	ltc-lsi	0.54	0.32	11.1
Training (all)	lsi-bnn	0.22	0.11	3.4
Training (all)	ltc-bnn	0.21	0.11	4.1
Training (all)	ltc-lsi	0.57	0.29	10.6
Adhoc	lsi-bnn	-0.06	0.16	2.7
Adhoc	ltc-bnn	-0.04	0.16	2.8
Adhoc	ltc-lsi	0.50	0.20	8.2
Adhoc (short)	lsi-bnn	-0.03	-0.03	-0.8
Adhoc (short)	ltc-bnn	-0.10	0.18	2.8
Adhoc (short)	ltc-lsi	0.30	-0.03	2.0
Routing	lsi-bnn	-0.04	-0.37	-7.8
Routing	ltc-bnn	0.05	-0.37	-7.0
Routing	ltc-lsi	0.47	0.38	11.7

Table 3: GPA_r, J_2 and Predicted Improvement in Average Precision for Different TREC Tasks

This is precisely the task for which our actual mixture performed the worst. The predicted degradation on the routing task is due entirely to the largely negative value for J_2, which corresponds to the bnn expert. Thus, because of the poor performance of the bnn expert on the routing task, it seems that improved performance isn't even possible using the simple linear model. It is not clear why the bnn expert has such a low J, but we suspect it is because a large number of terms in the FBIS corpus were not in the training corpus and were ignored. The ltc and lsi experts could make up for the missing terms due to their more sophisticated weighting schemes, but since the bnn expert weights each term equally, each missing term would significantly affect performance.

Table 4 shows the actual improvements we achieved with our mixture, and points to the shortfalls of predicting the best possible performance. We see that whereas the prediction for the Training (all) task is positive, we actually see a significant degradation. Also, the maximum predicted improvement for the Adhoc (short) task is much less than what is actually achieved. Thus, whereas the simulations and the corresponding regression model are useful in explaining the routing task performance, they cannot answer all questions concerning this type of linear mixture model.

The poor performance on the entire available training data (Training (all)) points to the possibility that our technique of only training on the top 100 documents may not scale up from smaller databases to the TREC corpus. This could be due to a number of reasons: the number top-ranked documents that we used may not be representative of the whole training set, or we may have overtrained, or our model just may not be powerful enough to reliably ensure improvement.

Task	Actual Improvement (%)
Training (top 100)	13.1
Training (all)	-31.4
Adhoc	8.4
Adhoc (short)	15.1
Routing	-35.6

Table 4: Actual Improvement in Average Precision for Different TREC Tasks

Task	Expert	Prec	J
Training (top100)	bnn	0.41	0.42
Training (top100)	lsi	0.39	0.35
Training (top100)	ltc	0.38	0.32
Training (all)	bnn	0.07	0.11
Training (all)	lsi	0.15	0.29
Training (all)	ltc	0.23	0.45
Adhoc	bnn	0.04	0.16
Adhoc	lsi	0.06	0.20
Adhoc	ltc	0.16	0.53
Adhoc (short)	bnn	0.06	0.30
Adhoc (short)	lsi	0.04	-0.03
Adhoc (short)	ltc	0.10	0.18
Routing	bnn	0.03	-0.37
Routing	lsi	0.18	0.38
Routing	ltc	0.21	0.45

Table 5: Average Precision and J for Different Experts and TREC Tasks

Our previous work has shown that J and average precision are highly correlated, a finding which justifies optimizing J. Table 5 shows these two measures for the different experts on different tasks. A linear regression of the two measures shows an $r^2 = 0.35$. However, because the Training (top 100) precision was measured using a set of documents with a higher percentage relevant documents, it ss artificially inflated and should not be included in the regression. When it is left out, $r^2 = 0.62$, a more reasonable amount of correlation.

6 CONCLUSIONS and FUTURE WORK

We have shown that a simple linear combination of scores from different IR experts can possibly improve the performance of those systems on novel documents. However, such improvement is not guaranteed, and depends on the performance of the individual experts as well as how similarly each is to the other. The model we present is a simple one, yet generally applicable to any collection of IR systems since it does not require knowing about how the systems work, it just needs their scores.

We also present a way of choosing parameters for any IR system model which is independent of the details of the model itself: optimization of a criterion (J) which is correlated with average precision. Despite our mixed results, we still believe that optimizing J is a good way to adjust IR system parameters. We believe our mixed results are artifacts of the way we trained the system and strength of our model.

We intend to further analyze our results from this TREC while preparing for the next one. We will try using a better training regime, one which uses a better optimization technique along with cross-validation to avoid overtraining and to choose the right number of top-ranked documents to train on. We are also in the process of extending our simulation experiments, using more realistic "experts" with reasonable levels of precision and also allowing negative combination weights. For next year's conference, we hope to use a more sophisticated model – one which varies the weights on each expert according to the current document or query. We also hope to be able to use other participants' entries in order to show that any collection of experts can be successfully used.

References

[Bartell, 1994] Bartell, B. T. (1994). *Optimizing Ranking Functions: A Connectionist Approach to Adaptive Information Retrieval.* PhD thesis, Department of Computer Science & Engineering, The University of California, San Diego, CSE 0114.

[Bartell et al., 1994a] Bartell, B. T., Cottrell, G. W., and Belew, R. K. (1994a). Automatic combination of multiple ranked retrieval systems. In *Proceedings of the ACM SIGIR*, Dublin.

[Bartell et al., 1994b] Bartell, B. T., Cottrell, G. W., and Belew, R. K. (1994b). Optimizing parameters in a ranked retrieval system using multi-query relevance feedback. In *Proceedings of the Symposium on Document Analysis and Information Retrieval*, Las Vegas.

[Berry, 1992] Berry, M. W. (1992). Large scale singular value computations. *International Journal of Supercomputer Applications*, 6(1):13–49.

[Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

[Dumais, 1995] Dumais, S. T. (1995). Latent semantic indexing (LSI): TREC-3 report. In [Harman, 1995a].

[Harman, 1995a] Harman, D., editor (1995a). *The Third Text REtrieval Conference (TREC-3)*, Gaithersberg, MD. National Institute of Standards and Technology. NIST Special Publication.

[Harman, 1995b] Harman, D. K. (1995b). Overview of the Third Text REtrieval Conference (TREC-3). In [Harman, 1995a].

[Jordan and Jacobs, 1994] Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214.

[Kantor, 1995] Kantor, P. B. (1995). Decision level data fusion for routing of documents in the TREC3 context: A base case analysis of worst case results. In [Harman, 1995a].

[Knaus et al., 1995] Knaus, D., Mittendorf, E., and Schäuble, P. (1995). Improving a basic retrieval method by links and passage level evidence. In [Harman, 1995a].

[Salton, 1971] Salton, G., editor (1971). *The SMART Retrieval System – Experiments in Automatic Document Retrieval.* Prentice-Hall Inc., Englewood Cliffs, N.J.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–23.

[Shaw and Fox, 1995] Shaw, J. and Fox, E. (1995). Combination of multiple searches. In [Harman, 1995a].