Information Retrieval, 1, 151–173 (1999)

Fusion Via a Linear Combination of Scores[1]

CHRISTOPHER C. VOGT vogt@cs.ucsd.edu
GARRISON W. COTTRELL gary@cs.ucsd.edu
Computer Science and Engineering, University of California San Diego, La Jolla CA 92093-0114, USA

Editors: Norbert Fuhr and Kui Lam Kwok

Received June 30, 1998; Revised February 4, 1999; Accepted March 29, 1999

Abstract. We present a thorough analysis of the capabilities of the linear combination (LC) model for fusion of information retrieval systems. The LC model combines the results lists of multiple IR systems by scoring each document using a weighted sum of the scores from each of the component systems. We first present both empirical and analytical justification for the hypotheses that such a model should only be used when the systems involved have high performance, a large overlap of relevant documents, and a small overlap of nonrelevant documents. The empirical approach allows us to very accurately predict the performance of a combined system. We also derive a formula for a theoretically optimal weighting scheme for combining 2 systems. We introduce d—the difference between the average score on relevant documents and the average score on nonrelevant documents—as a performance measure which not only allows mathematical reasoning about system performance, but also allows the selection of weights which generalize well to new documents. We describe a number of experiments involving large numbers of different IR systems which support these findings.

Keywords: linear combination, fusion, neural networks, routing, performance evaluation

1. Introduction

In the past, neural network models which have been applied to the Information Retrieval problem have typically used very large feature vectors (document and query vectors) which are traditionally used in IR systems (see for example, Crestani 1994, Wong et al. 1993, Boughanem et al. 1993). Unfortunately, the resulting large networks generally require large numbers of training examples, a rare commodity in the IR setting. Although work using Latent Semantic Indexing (Deerwester et al. 1990) to reduce the number of features has met with some success (Vogt et al. 1999), LSI is itself computationally expensive.

Perhaps a wiser approach can be found in fusion, where the results from multiple IR systems are combined to generate a single (hopefully better) list of potentially relevant documents in response to presentation of a single query to a number of component systems. Fusion allows a significant reduction in the number of features, often to just one feature per system—typically the system's estimate of the document's probability of relevance. As such, a fusion neural network model can be much smaller than a traditional one based on document vectors. Fusion also allows leveraging of the component systems in several ways by exploiting a number of effects (from Diamond 1998):

- **The Skimming Effect** happens when retrieval approaches represent documents differently and thus retrieve different relevant items. A combination model that takes the

top-ranked items from each of the retrieval approaches (i.e., "skims" the results lists) could then not only increase recall (due to the different relevant documents) but also precision (assuming the systems have a higher density of relevant documents near the beginning of their results lists).

- **The Chorus Effect** occurs when several retrieval approaches suggest that an item is relevant to a query; this tends to be stronger evidence for relevance than that of a single approach. A combination model which ranks documents in the intersection of the retrieved lists higher would be able to exploit this effect.
- **The Dark Horse Effect** occurs because a retrieval approach may produce unusually accurate (or inaccurate) estimates of relevance for some documents, relative to the other retrieval approaches. In order to exploit this effect, a combination model would have to condition its combination technique on the document being scored, "listening" to one system more than another based on some features of the document.

It should be noted that when choosing how to combine the results from different IR systems, the Dark Horse Effect is at odds with the Chorus Effect—one argues for listening to just one system, while the other advocates listening to them all. Likewise, a large Chorus Effect cuts into the possible gain from the Skimming Effect—if the combination model only upranks documents in the intersection of results lists, then the gains in recall from skimming become nearly impossible. These phenomena argue for a sophisticated fusion model which is able to predict when these effects will occur and take advantage of them. Such models would almost certainly need to make use of training data in the form of user feedback in order to fine tune their performance.

Small neural network fusion models may allow easier training, but their simplicity inflicts a penalty—they lack expressive power because they have so few parameters (see Hertz et al. 1991). Thus, it seems likely that the possible improvement in performance from these models would be limited. In this article, we examine in detail one such fusion model: the linear combination of scores (LC). The LC model has been used by many IR researchers with varying degrees of success: (Bartell et al. 1994, Kantor 1995, Knaus et al. 1995, Selberg and Etzioni 1996, Shaw and Fox 1995), and (Vogt et al. 1997). Our analysis of the model reveals what types of systems the model works best with, and explores techniques for training the model.

This article has the following format. First we describe the model and the specific problems we are examining, along with the data we use and assumptions. Next, we both empirically and mathematically derive explanations for when it makes sense to use the LC model. This also leads to an expression for the performance and the optimal weighting of an LC fusion system. Finally, we describe experiments in training the LC model, and a training technique which generalizes to large numbers of systems as well as more complicated models. Through these analyses, we gain insight into why and when the LC model works well.

2. Background

2.1. The LC Model

The linear combination model calculates the real-valued relevance ρ of a document x to a query q based on the weights $\mathbf{w} = (w_1, w_2, \ldots, w_s)$ given to each of the s individual IR

systems, and their estimates of relevance ρ_i:

$$\rho(\mathbf{w}, x, q) = \sum_{i=1}^{s} w_i \rho_i(x, q)$$

This value is then used to rank the documents. For only two IR systems, this simplifies to:

$$\rho(w_1, w_2, x, q) = w_1\rho_1(x, q) + w_2\rho_2(x, q) \tag{1}$$

However, all that really matters is the ranking given by the combined system. Thus, for the case of two systems, only the ratio of the two weights and the relationship of the signs on the weights are important, and equation (1) can be replaced by one using a single weight:

$$\rho(w, x, q) = \sin(w)\rho_1(x, q) + \cos(w)\rho_2(x, q) \tag{2}$$

where $\sin(w) = w_1$ and $\cos(w) = w_2$. This formulation allows for all possible relationships of the signs of the two weights (+/+,+/-,-/+,-/-) and all possible ratios w_1/w_2 (assuming adequate machine precision).

The LC model is more flexible than others which have been tried in the past. Most of them consist of a single choice of parameters, such as the sum of scores or the maximum score, or a fixed weighting based on individual system performance (see Belkin et al. 1995, Shaw and Fox 1995, Kantor 1995). The difference with our approach is that we will use a search procedure to optimize the weights based on metrics of system performance. The LC model is also equivalent to the simplest kind of neural network, a single layer net (see Hertz et al. 1991). The addition of a squashing function on the output unit does not add any power, as it would not change the ranking induced by the combined system.

2.2. The Data

Our study of fusion necessitates the availability of a large number of IR systems. We have chosen to use the entries in the TREC5 adhoc track (Harman 1997). TREC (Text REtrieval Conference) is an annual conference sponsored by the National Institute of Standards and Technology, in which participants are given very large text corpora and submit the results of their retrieval techniques in a sort of contest. Specifically, in the adhoc track, each participant submits the top 1000 documents returned by their system in response to 50 queries supplied by NIST, and each participant can submit up to 4 runs. In TREC5, 61 runs were submitted, for a total of 3050 lists of 1000 documents, each of which we treat as a separate IR "system".

One issue arises when combining such lists of top-ranked documents—what score should be given to documents returned by one system but not the other? We have assumed that for such documents, the system which did not return them gave them a score of zero. In so doing, we also had to eliminate any negative scores for documents, because otherwise the unreturned documents would get ranked above those with negative scores. We did this by adding the absolute value of the lowest score to all documents for any system which

had negative scores. We believe that zero scores for unseen documents is a reasonable choice—the vast majority of documents are not relevant, and most systems give a zero score to nonrelevant documents.

2.3. Problems Addressed

A common dichotomy used by fusion researchers is the difference between data fusion and collection fusion. The former takes place in a setting when all of the IR systems involved have access to the same text collection. The latter is used when the collections searched by all IR systems are disjoint. Because of our use of TREC entries, which consist of lists of 1000 or fewer documents, our work falls under the label of data fusion, but it is not data fusion in the purest sense. This is because it is possible for a document to be in the list returned by one system and not in the list of the other. Thus, our work falls somewhere between data and collection fusion.

Another distinction is in the types of IR problems. TREC distinguishes two main problems: adhoc and routing. Adhoc retrieval occurs when the text collection is relatively static, and new queries are constantly being submitted by users. In contrast, routing has a standing query, and new documents are arriving which need to be filtered according to that query.

The difference between these two tasks is important in the context of the LC model. For the routing task, a new set of weights can be trained for each standing query. However, for the adhoc task, one set of weights must be applied for all queries. Clearly, the routing task should be easier to solve using this model, since the weights can be tuned on a per-query basis.

3. Limitations of the LC Model

Using the LC model does not always result in an improvement in performance. Previous work using the TREC data has shown that even on the training set, significant improvement is achievable for less than half of the possible combinations (Vogt and Cottrell 1998a). Why is it that it works sometimes and not others? When does the LC model work?

In this section we use two techniques, one empirical and one analytical, to determine the sources of the LC model's power. In the process, we gain a deeper understanding of why it works, its limitations, and which systems can be successfully combined using the LC model.

3.1. Empirical Analysis[2]

We introduce here a technique for analyzing the behavior of fusion models and apply it to the LC model. The technique involves measuring various properties of the component IR systems, and using them in a linear regression to predict the performance of the combination. By examining how the measures are weighted by the regression, we gain an intuitive feel for when using the model pays off. Our technique works surprisingly well—the resulting regression can predict the performance of unseen combinations very accurately (on the test set, $r^2 = 0.95$).[3]

3.1.1. Method Our data set is all 61 TREC5 entries on the first 20 queries—1220 lists of up to 1000 documents. For each query, there are 1830 possible pairs of systems. We begin by making a number of measures of all $36{,}600 = 1830 \times 20$ triples of system pairs/queries. These measures are meant to indicate either how well each system performs or how similar the two systems are to each other. The performance measures include average precision (indicated by **P**) and a statistical measure of rank correlation between the system and the relevance judgments (**J**) (Bartell et al. 1994). Average precision was used because it incorporates both precision and recall into one measure. **J** is defined as:

$$\mathbf{J} = \frac{\sum_{x,x':x\succ_q x'} \rho(\mathbf{w}, x, q) - \rho(\mathbf{w}, x', q)}{\sum_{x,x':x\succ_q x'} |\rho(\mathbf{w}, x, q) - \rho(\mathbf{w}, x', q)|}$$

where $x \succ_q x'$ indicates the user prefers document x to document x' 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. These two measures are subscripted a for the better of the two systems and b for the worse (as measured by **P**). The pairwise similarity measures include:

- Guttman's Point Alienation (GPA) (Guttman 1978)—a measure of how similar two rankings are to each other, which is calculated as:

$$GPA = \frac{\sum_{x,x'}(\rho_1(x, q) - \rho_1(x', q))(\rho_2(x, q) - \rho_2(x', q))}{\sum_{x,x'} |\rho_1(x, q) - \rho_1(x', q)||\rho_2(x, q) - \rho_2(x', q)|}$$

- the number of documents in the intersection of the two lists of returned documents (I),
- the correlation coefficient from a linear regression of the scores of documents in the intersection of the two systems (C), which is actually the r^2 value of a regression which uses one system's scores to predict the other's,
- the number of relevant documents returned by one system but not the other, divided by the total number of relevant documents returned by that system (U for uniqueness),
- Lee's (Lee 1997) overlap measures, O_{rel} and O_{nonrel}, which measure the proportion of relevant and nonrelevant documents in the intersection of the two lists. These two measures are calculated as:

$$O_{\text{rel}} = \frac{2 \times I_{\text{rel}}}{R_1 + R_2}$$

$$O_{\text{nonrel}} = \frac{2 \times I_{\text{nonrel}}}{N_1 + N_2}$$

  where R_i is the number of relevant documents and N_i is the number of nonrelevant documents returned by system i, respectively, and
- the ratio of the performance of the two systems: $\mathbf{P_b}/\mathbf{P_a}$, since Ng (1998) found this to be an important predictive factor of the improvement of the combination.

Also, because it seemed likely that measuring the similarity of the two systems on relevant documents is more important than on nonrelevant ones, the first three measures were also calculated using only relevant documents, and are denoted: GPA_{rel}, I_{rel}, and C_{rel}. One last measure, GPA_{ni} (for "not irrelevant") is the GPA using pairs of documents where at least one document is relevant.

After normalizing the scores for each system on each query by dividing by their respective means we found the optimal combination for each possible pair. For each query and each pair of systems, the single weight w was chosen by optimizing average precision using golden section search (Press et al. 1995), and the best w was used to generate a combined system (of 1000 documents) according to Eq. (2). Golden section search is a simple bracketing optimization technique for finding local maxima of single variable functions which does not require gradient information. Hence, it is useful in this case, because $\mathbf{P}$ cannot be differentiated with respect to the weight w. Because systems were combined on a per-query basis, this experimental setup most accurately simulated the routing task.

We then performed a multiple linear regression using the aforementioned measures as predictor variables and the average precision of the optimal combination as the target. 80% of the pairs (29,280 total—the "training set") were used in the regression.

3.1.2. Results Table 1 presents the results of the multiple linear regression. Measures are sorted by decreasing F value, indicating roughly how important each measure is in predicting the average precision of the optimally combined system. All measures above the horizontal line in the table contribute to some degree (as indicated by F values much larger than 1), and have significance of $p < 0.005$. The $r^2 = 0.94$ value indicates that the fit of the model is very accurate. Furthermore, the model generalizes extremely well to new data—when the remaining 20% of the pairs (the "test set") were plugged into the model, $r^2 = 0.95$.

Positive regression coefficients in Table 1 can be interpreted as meaning that the corresponding measures should be maximized in order to maximize the performance of the combined system. Likewise, measures with negative coefficients should be minimized. This leads to the following conclusions: the better system should have high performance ($\mathbf{P_a}$ and $\mathbf{J_a}$ have positive coefficients), whereas the performance of the worse system may or may not be good ($\mathbf{P_b}$ and $\mathbf{J_b}$ have opposite signs). The positive coefficients on GPA and C_{rel} indicate that the two systems should generally rank documents in their intersection similarly and the distribution of scores by both systems should be similar to each other. On the other hand, the negative coefficients on GPA_{rel} and GPA_{ni} indicate that each system should rank *relevant* documents differently than the other system. Finally, the negative coefficient on O_{nonrel} means that the two systems should retrieve different sets of nonrelevant documents.

The table leads to conflicting conclusions about the overlap of relevant documents. The negative coefficients for U_a and U_b indicate that both systems should not return unique relevant documents, whereas the negative coefficient on O_{rel} indicates they should. As it turns out, the negative coefficient on O_{rel} is inaccurate because O_{rel} is directly related to U_a and U_b by $\frac{1}{2O_{\text{rel}}} = \frac{1}{1-U_a} + \frac{1}{1-U_b}$. The regression simply accounts for the effect of uniqueness using U_a and U_b alone.

Table 1. Results of linear regression.

Measure	Normalized regression coefficient	F
$\mathbf{P_a}$	0.8990	105226.5632
U_a	−0.1267	433.7695
U_b	−0.0386	357.3758
$\mathbf{J_b}$	0.0428	338.4343
GPA_{ni}	−0.0363	229.5316
$\mathbf{J_a}$	0.0276	193.4874
O_{rel}	−0.0546	60.1901
$\mathbf{P_b}$	−0.0188	37.1321
C_{rel}	0.0120	33.8159
O_{nonrel}	−0.0444	23.0822
GPA	0.0131	20.9807
I_{rel}	0.0080	14.707
GPA_{rel}	−0.0082	6.2473
$\mathbf{P_b}/\mathbf{P_a}$	−0.0054	4.5710
I	−0.0129	1.7961
C	0.0022	1.1175

Results of a linear regression for predicting the combination's average precision ($r^2 = 0.94$). Positive coefficients indicate the measure should be maximized, and negative coefficients indicate it should be minimized.

In fact, by repeating the regression using only $\mathbf{P_a}$, $\mathbf{P_b}$, O_{rel} and O_{nonrel}, we can predict the combined system's precision with nearly the same accuracy as the original regression ($r^2 = 0.94$, see Table 2). This table shows that the following three conclusions about when it makes sense to use the LC model are strongest, namely when:

- at least one exhibits good performance,
- both return similar sets of relevant documents, and
- both return dissimilar sets of nonrelevant documents.

The discussion of U_a, U_b vs. O_{rel} above, and the inclusion of a second regression (in Table 2), point to a subtle difficulty in our use of regression—the problems of correlated predictor variables and variable selection. The typical technique for dealing with large numbers of predictor variables is to select a subset of relevant variables via stepwise regression or some similar approach. Unfortunately, these approaches do not fare well when the predictor variables are well correlated, as is the case for the variables used in the above regressions (every measure is correlated with at least one other measure with $r^2 > 0.2$). Thus, it was necessary for us to spot variable correlations manually. It was also necessary for us to examine various different subsets of the predictor variables, based on the correlations and our own hypotheses of which would prove most informative.

Table 2. Results of linear regression on a subset of predictors.

Measure	Normalized regression coefficient	F
$\mathbf{P_a}$	0.9366	191543.1029
O_{rel}	0.1021	2249.4031
O_{nonrel}	−0.0581	975.4101
$\mathbf{P_b}$	−0.0228	119.1705

3.1.3. Predicting Improvement In the above analysis, we try to predict the performance of the combined system. Ng (1998) argued that a more important measure of the effectiveness of fusion is the amount of improvement gained by using the fusion over just using one of the two systems. Thus, in addition to the above analysis, we also used the measures to predict two metrics of improvement: $\mathbf{P} - \mathbf{P_a}$ and $(\mathbf{P} - \mathbf{P_a})/\mathbf{P_a}$.

The regression using $\mathbf{P} - \mathbf{P_a}$ as the target generated coefficients of the same sign, similar magnitude, and same order of importance as those reported in Table 1, *except* for the coefficient on $\mathbf{P_a}$, which was still the most significant measure, but now had a large negative coefficient. This is to be expected, since it would probably be difficult to improve systems which are already very good. The r^2 for this regression was 0.12 (and only 0.04 on the test data), so it seems that predicting the raw improvement using a linear regression is more difficult than predicting the actual performance. This makes sense since the performances of the individual systems are included as a predictor variables, and it seems likely that these would weigh heavily in the performance of the combination (as Table 1 confirms), but there are no corresponding predictor variables for the improvement. Despite this fact, the conclusions drawn from this regression would be the same as those drawn from the original regression on performance.

Table 3 reports the results of the regression to predict the second measure of improvement: $(\mathbf{P} - \mathbf{P_a})/\mathbf{P_a}$. Because of the low r^2 (0.06), conclusions drawn from this analysis alone are dubious. However, since all of the significant coefficients are of the same sign and relative magnitude as those for the other definition of improvement, we once again support the conclusions from the original regression.

3.2. Mathematical Analysis

The conclusions in the previous section were arrived at empirically, and give little insight into why they may be true. We now provide a mathematical justification for these hypotheses. Furthermore, we mathematically derive an expression for the optimal weighting and from this derivation arrive at an expression for the performance of an LC model. As in the empirical case, we concentrate on the routing problem only.

3.2.1. d as a Performance Measure Our analysis hinges on the use of d as a performance measure. d is equal to the difference between the system's mean score on all positive examples (relevant documents) $\bar{p}$ and the mean score on all negative examples (nonrelevant

Table 3. Results of linear regression.

Measure	Normalized regression coefficient	F
$\mathbf{P_a}$	−0.2322	410.9723
U_a	−0.4326	295.9000
U_b	−0.1251	219.4178
O_{rel}	−0.2402	68.1932
GPA_{ni}	−0.0684	47.7767
C	0.0449	27.0335
O_{nonrel}	−0.1907	24.9586
$\mathbf{P_b}/\mathbf{P_a}$	0.0356	11.7822
I_{rel}	−0.0234	7.2895
$\mathbf{P_b}$	−0.0273	4.5785
GPA_{rel}	−0.0270	3.9932
$\mathbf{J_b}$	0.0171	3.1843
C_{rel}	0.0093	1.1863
GPA	0.0110	0.8724
I	0.0230	0.3327
$\mathbf{J_a}$	0.000	0.0000

Results of a linear regression for predicting the combination's improvement $(\mathbf{P} - \mathbf{P_a})/\mathbf{P_a}$ ($r^2 = 0.06$).

documents) $\bar{n}$. As such, d is only applicable in the situation where documents need to be placed in one of two categories (relevant and nonrelevant). Without loss of generality, we assume the scores have been normalized to the interval $[0, 1]$, to make values of d comparable across systems. d is equal to the numerator of D_c from signal detection theory (Egan 1975). D_c normalizes this difference by the standard deviations of the positive and negative example score distributions:

$$D_c = \frac{\bar{p} - \bar{n}}{\sqrt{\sigma_P \sigma_N}}$$

A variation of D_c called the Swets measure has been examined before in the information retrieval setting (van Rijsbergen 1979, p. 157):

$$S = \frac{\bar{p} - \bar{n}}{\frac{1}{2}(\sigma_P + \sigma_N)}$$

However, despite their statistically based theoretical attractiveness and the excellent argument put forth by Swets, neither D_c nor S have ever caught on as a basis for performance evaluation in IR. Before proceeding with our analysis based on d, we first justify its use as a performance measure. Our first step is to argue for D_c as a performance metric.

First, we argue that IR systems typically distribute scores according to an exponential distribution. Figure 1 shows the empirical distribution of two typical IR systems from TREC5 after scores have been normalized to $[0, 1]$. These are typical in the sense that they are close to the average distribution over all 61 TREC5 entries. Note that these distributions are summed over all 50 queries for each system. Figure 2 shows two atypical distributions. The curves in these graphs do not show the typical consistent dropoff of number of documents with higher scores. These atypical curves must be due the particular scoring function used by each system, and may also be an artifact of the summing over all 50 queries. However, the important thing to note is that the *typical* distribution not only has the appearance of a negative exponential, but as figure 3 shows, it also has the property that its mean is approximately equal to its standard deviation—a property of all negative exponential distributions. Note that this property is stronger for the negative score distributions (the negative score histogram has lower variance), indicating that these scores are more reliably distributed in this manner.

According to signal detection theory (Egan 1975 (p. 136)), the proper coefficient of discrimination for a negative exponential distribution is D_c. This coefficient measures the efficacy of the system which is attempting to separate the positive examples from the negative. Intuitively, D_c makes sense since it is maximized by increasing the scores (and thus decreasing the ranks) on relevant documents, and minimizing nonrelevant scores.

Thus, D_c is both a theoretically proper and an intuitive measure of IR system performance. We now argue for the use of d, the numerator of D_c. As previously noted, d is simply the mean score of relevant documents minus the mean score of nonrelevant. Since we assume scores which are normalized to $[0, 1]$, this means that a perfect IR system would have $d = 1$, since all documents would have a score of 1 and all nonrelevent documents a score of 0. Likewise, the worst possible IR system would have $d = -1$. Real IR systems typically have a positive d (the average across all of the TREC5 entries is $d = 0.175$, with minimum of -0.657 and maximum of 0.999).

First, we note that d is very highly correlated with D_c ($r^2 = 0.93$), so that optimizing one would most likely optimize the other. This makes sense, since D_c is a normalized version of d. d is also very well correlated with traditional IR measures like average precision (for the LC model on TREC5 data, $r^2 = 0.78$). Furthermore, d can be directly interpreted in terms of the probability ranking principle. If an IR system ranks documents according to their true probability of relevance, then all of the relevant documents would have high scores (and nonrelevant would have low scores), leading to a large d. Thus, d maintains the intuitiveness of D_c. However, d has a simpler algebraic form—a property which we exploit in our analysis below. Finally, as we will show later, optimizing d has effects similar to optimizing average precision directly (as would be expected due to their high correlation). For all of these reasons, we claim that d is a reasonable measure of IR system performance.

3.2.2. Mathematical Support for the 3 Hypotheses The Appendix presents details of a mathematical derivation for the performance of an optimally combined system. The two major conclusions from the derivation are that:

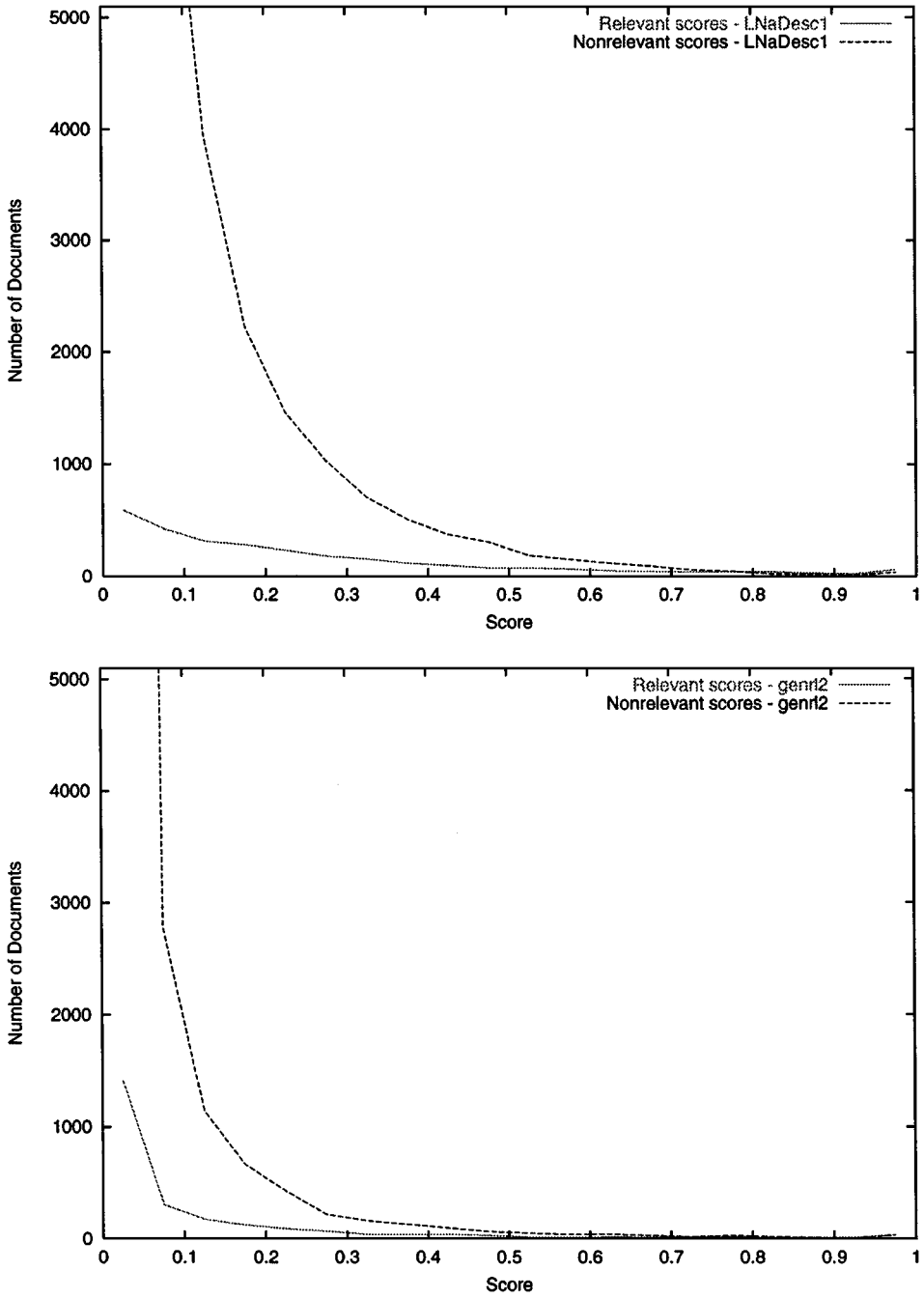

Figure 1. Smoothed histogram of *typical* relevant and nonrelevant score distributions (from entries LNaDesc1 and genrl2).

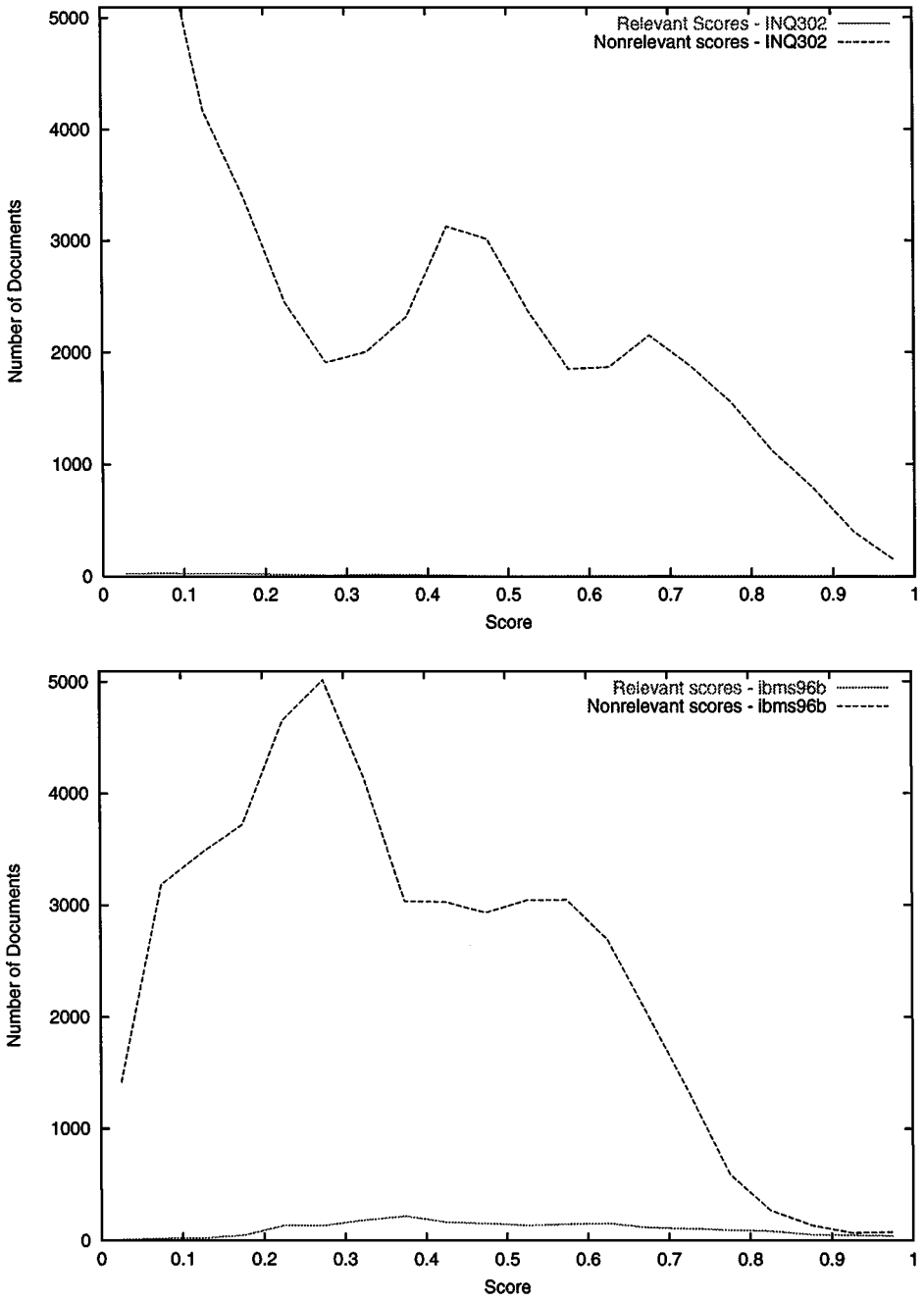

Figure 2. Smoothed histogram of *atypical* relevant and nonrelevant score distributions (from entries INQ302 and ibms96b).

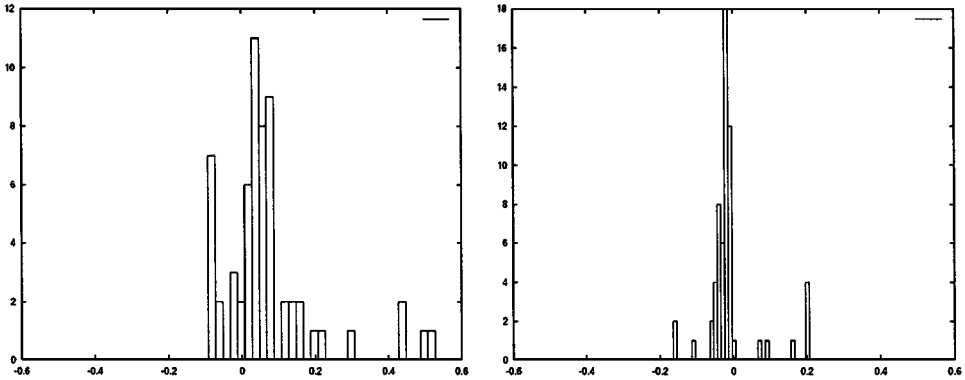

Figure 3. Histograms of the difference between the mean and the standard deviation for positive example scores ($\bar{p} - \sigma_P$, left) and negative example scores ($\bar{n} - \sigma_N$, right) on the 61 TREC entries. Note that both are centered near 0 and the negative score histogram has smaller variance.

1. the optimal weight w_{opt} to use when combining two systems (a and b) satisfies the equation:

$$\tan(w_{\text{opt}}) = \frac{\delta_a}{\delta_b}$$

2. the combined system's performance when using this weight is:

$$d_{\text{opt}} \propto \delta_a^2 + \delta_b^2$$

where $\delta_a = \alpha_p \bar{p}_a - \alpha_n \bar{n}_a$, and $\delta_b = \beta_p \bar{p}_b - \beta_n \bar{n}_b$, and the coefficients on the score averages are defined by:

$$\alpha_p = \frac{P_a + P_{ab}}{P_{ab} + P_a + P_b}$$

$$\beta_p = \frac{P_b + P_{ab}}{P_{ab} + P_a + P_b}$$

$$\alpha_n = \frac{N_a + N_{ab}}{N_{ab} + N_a + N_b}$$

$$\beta_n = \frac{N_b + N_{ab}}{N_{ab} + N_a + N_b}$$

where P_a indicates the number of relevant documents returned only by system a, P_b is the same for system b, P_{ab} is the number of relevant documents in the intersection, and the N's refer to number of nonrelevant documents, with subscripts taking the same meanings as for relevant.

In other words, the weight is a function of the ratio of the component systems' performances, as mediated by ratios of the number of documents in the overlaps. The performance of the combination is a direct function of these mediated performance values.

In order to maximize d_{opt}, it would be best to maximize both δ_a and δ_b. It is easy to show that since $0 \leq \alpha_p, \alpha_n, \bar{p}_a, \bar{n}_a \leq 1$, δ_a is optimal when $\alpha_p = 1$ and $\alpha_n = 0$. Similarly, when $\bar{p}_a = 1$ and $\bar{n}_a = 0$ (i.e., when d_a is maximized), δ_a is maximal. Similar arguments hold for δ_b, β_p, β_n, $\bar{p}_b$ and $\bar{n}_b$.

We are now in a position to address the three hypotheses: Linear combination is warranted when,

1. at least one system exhibits good performance,
2. both systems return similar sets of relevant documents,
3. both systems return dissimilar sets of nonrelevant documents.

The first point follows directly from the fact that δ_a and δ_b are maximal when $d_a = \bar{p}_a - \bar{n}_a$ and $d_b = \bar{p}_b - \bar{n}_b$ are maximal. The second two points can be concretely supported as follows. Recall that δ_a is optimal when $\alpha_p = 1$ and $\alpha_n = 0$. Since $\alpha_p = \frac{P_a + P_{ab}}{P_a + P_b + P_{ab}}$, it is equal to 1 when $P_b = 0$. Likewise, $\beta_p = 1$ when $P_a = 0$. Thus, δ_a and δ_b are optimal when both systems return no *unique* relevant documents (i.e., they maximize the overlap of relevant documents). Conversely, since $\alpha_n = \frac{N_a + N_{ab}}{N_a + N_b + N_{ab}}$, it goes to 0 as $N_b \to \infty$. The same argument holds for β_n. So, δ_a and δ_b are optimal when both systems are retrieving different sets of nonrelevant documents.

3.2.3. Summary We have provided mathematical support for our empirically derived conclusions about when to use the linear model. Namely, when at least one system exhibits good performance, when both systems return similar sets of relevant documents, and when both systems return dissimilar sets of nonrelevant documents. These conclusions basically state that the LC model primarily exploits only the Chorus Effect. Note that this conclusion applies specifically to the LC model—other fusion models would presumably exploit other effects. We have also derived a formula for the optimal weighting, as well as an expression for the performance of the combined system assuming the optimal weighting.

4. Training the LC Model

We now turn to the practical side of things. In the previous section we derived an optimal weighting for two systems according to d. However, using this weighting is not a practical technique for several reasons. First, it is not general, in that it does not cover the case of more than two systems. Second, it would be more satisfying if we could choose a weight which optimizes a more traditional IR performance measure, such as average precision or exact precision. Finally, the optimal weights used in the previous section were not tested for generalization to new data. In this section, we describe a series of experiments in which we compare two different techniques for choosing the weights of an LC model.

4.1. Advantages of Using d as an Optimization Criterion

Recent work (Vogt and Cottrell 1998b) has argued in favor of using $d' = \frac{\bar{p}-\bar{n}}{\sigma_N}$ as an optimization measure for ranking problems such as those seen in IR. The list of reasons to prefer d' also apply to d. Namely,

- it is differentiable with respect to model parameters (i.e., weights), thus it can be applied to more complex models (via gradient based techniques) than optimizing precision directly, which has no gradient,
- it may be suitable for online learning,
- it is roughly correlated with other well-known measures of performance like average precision,
- it is generally cheaper to calculate than rank-order statistics, such as **J** (Bartell et al. 1994),
- it has an intuitive meaning: as a measure of how well the scores on positive examples are separated from those on negative examples.

Below we show that optimizing d to select a weighting scheme for combining two IR systems works nearly as well (as measured by average precision) as optimizing average precision itself. It is important that we distinguish between an optimization criterion and a performance measure. Often, the same function plays both roles. However, in the following experiments we may choose the weights by optimizing either d or average precision (**P**), but we will always evaluate the resulting combined system using **P**.

4.2. Training a Model for the Routing Problem

As previously noted, the routing problem should be easier for the LC model (or any parameterized model) because a separate model can be trained for each query, assuming enough training data. For this reason, we begin by examining the routing problem.

4.2.1. Method We examined all 61 submissions to the TREC5 adhoc track. For each of the 1830 possible pairs, on each of the 50 TREC5 adhoc queries, and 70% of all possible documents, we chose w in one of two ways:

- using a golden section search (Press et al. 1995) to optimize **P** directly, or
- using a golden section search to optimize d directly.

An examination of the surfaces of **P** and d has shown that they usually have only a single maximum, which this technique should always find. Although we are able to calculate the optimal weight for d using equation (6), we use the golden section technique because in the future we would like to be able to apply the technique of optimizing d to models with more than one parameter, for which an equation may not be derivable.

We did not use a hold out set to stop training since the LC model has only one parameter and thus is unlikely to overfit the training data. We tested each of the trained combinations on the remaining 30% of the documents, evaluating each combination using **P**.

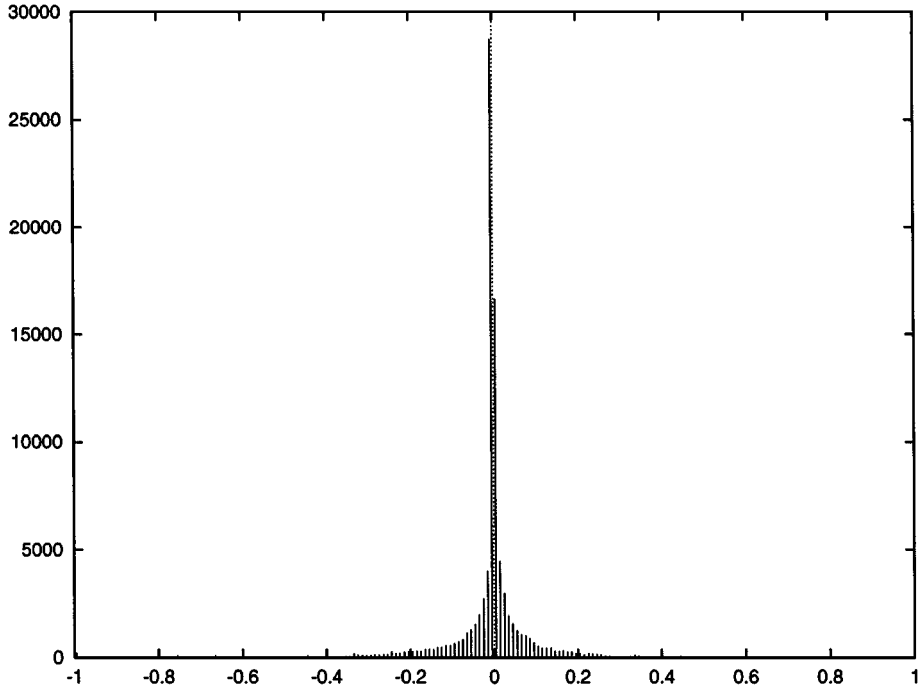

Figure 4. Histogram of $\mathbf{P_P} - \mathbf{P}_d$ on the Routing test set for 91500 TREC5 combinations.

4.2.2. Results and Discussion For each of the $91500 = 1830 \times 50$ pair/query triples, we calculated the difference between the value of average precision on the test set as determined by the weight being trained on d and the weight being trained on **P**. That is, we are looking at the quantity $\mathbf{P_P} - \mathbf{P}_d$ on the test set. This measures how much better training on **P** is than training on d. Figure 4 displays the histogram of this difference. The average difference is -0.009. This means that on average, training using d gives better generalization when **P** is the measure of performance. Although an ANOVA shows that the $\mathbf{P_P}$ and $\mathbf{P}_d$ distributions are different ($p < 0.0001$), to interpret whether or not this is really a meaningful difference, consider a query that has 75 relevant documents (the median for a TREC5 query). Then a difference of 0.009 in **P** means that only one or fewer more relevant documents are returned. It is also interesting to note that there are a significant number of combinations where training on d is better than on **P** (differences less than zero).

Table 4 gives some idea as to the amount of improvement gained by using the LC model. The motivation behind the data presented in this table is that a reasonable heuristic to deciding whether a particular pair of systems should be combined is to examine performance on the training set, and combine *only* if there is improvement. As noted previously, the LC model is not always capable of improving over the two component systems. Of the 91500 pair/query triples, 80324 (88%) saw improvement on the *training* set over both of the component systems when trained using **P**. Of those 80324, 32493 (40%) also saw improvement on the *test* set. The table also shows that over the 80324 combinations

Table 4. Training results for routing.

Training method	# Combos showing some improvement on the training set	# Combos showing improvement on both training and test sets	% of improved training combos which also improve on test set	Average degradation or improvement on test set $((\mathbf{P}/\mathbf{P_a}) - 1)$
P	80324	32493	40%	−14%
d	37661	21028	56%	+15%

Number of pair/query triples (out of 91500) which achieved better performance (as measured by **P**) than the better component system ($\mathbf{P_a}$) on the routing problem. Average improvement/degradation on the test set is over all combinations counted in the second column.

which saw improvement on the training set, the average degradation (as measured by the percentage of change in average precision of the combination as compared to the better of the two systems) on the test set over the better of the two component systems was −14%, a significant drop. The fact that the LC model tends to degrade performance for most pairs of systems should not be surprising—recall the results of the previous section, which indicate that the LC model should only be used in certain situations. Apparently, training using *d* allows better discrimination of when these situations arise. While training on *d* results in fewer systems which see improvement on the training set (37661), it allows for much better generalization when those weights are used on the test data (21028, or 56%, see improvement). Furthermore, of those 37661 triples which see an improvement on the training set, they also give rise to an average improvement on the test set, as opposed to the degradation incurred by training on **P**. The improvement is on the order of 15% higher average precision for the combination than the better of the two systems.

4.3. Training a Model for the Adhoc Problem

All of our analysis thus far has focused on the routing problem. One can certainly argue that the adhoc problem, where new queries are constantly being submitted to the IR system, is also an important problem, especially for the World Wide Web. The following experiments explore this aspect of IR.

4.3.1. Method For the routing case we were able to examine the full set of all the TREC5 queries. However, for the adhoc problem, we reduce the space of combinations by only examining three subsets of the total 61 systems. The first subset, labeled "chorus", consists of 10 systems (see Table 5). These systems were chosen because many of the 45 possible pairings have a high overlap of relevant documents and low overlap of nonrelevant documents, and thus are theoretically able to exploit the linear combination model. The second subset, labeled "diverse," was chosen to maximize the differences between the systems in the subset. These differences were simply the average of all the pairwise measures used in our empirical analysis above after they had all been normalized to *z*-scores. Recall that these pairwise measures included the correlation of scores, similarity of rank order as measured by Guttman's Point Alienation statistic, and the size of the intersection of the documents

Table 5. Subsets used in adhoc experiments.

Subset	Systems
Chorus	KUSG2 KUSG3 anu5man4 anu5man6 gmu96ma1 gmu96ma2 mds002 mds003 uwgcx0 uwgcx1
Diverse	CLTHES DCU961 anu5aut1 anu5man6 brkly17 colm1 fsclt4 gmu96ma1 mds002 uwgcx0
Random	CLCLUS Cor5M2rf DCU961 ETHas1 ETHme1 LNaDesc2 erliA1 gmu96ma2 ibmgd1 ibmge2

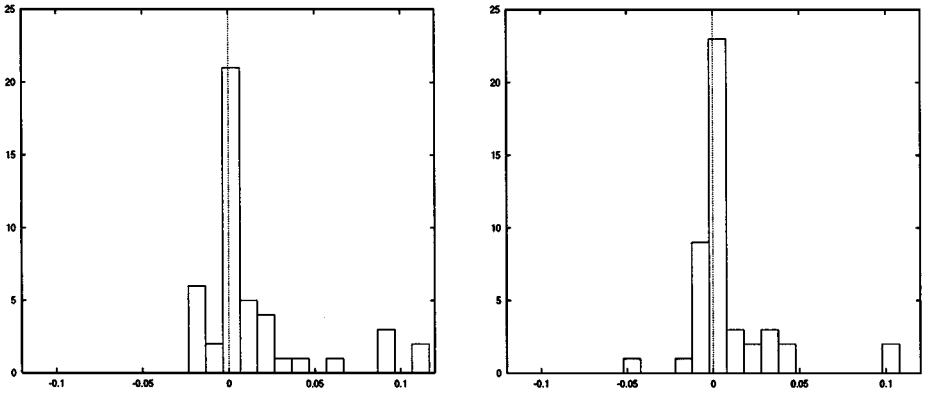

Figure 5. Histogram of $\bar{\mathbf{P}}_{\bar{\mathbf{P}}} - \bar{\mathbf{P}}_{\bar{d}}$ on the Adhoc test set for the chorus (left) and random (right) subsets.

returned by both systems. The third subset of ten systems was chosen randomly from the 61 entries.

For each subset and each of the possible 45 pairs of systems from that subset, w was chosen using golden section search to maximize either $\bar{\mathbf{P}}$ or $\bar{d}$ ($\mathbf{P}$ or d when averaged over 35 randomly chosen queries). We then tested each of the trained combinations on 13 of the remaining queries, evaluating each combination using $\bar{\mathbf{P}}$.

4.3.2. Results and Discussion Once again, we use $\bar{\mathbf{P}}_{\bar{\mathbf{P}}} - \bar{\mathbf{P}}_{\bar{d}}$ on the test set as our comparison metric. Figure 5 displays histograms of this difference for the chorus and random subsets (the diverse histogram is similar to the one for chorus). The average difference for the chorus subset is 0.012, diverse is 0.011, and random is 0.006. This means that on average, training using $\bar{d}$ gives worse generalization when $\bar{\mathbf{P}}$ is the measure of performance, just the opposite effect as observed for routing. Once again, although these differences are statistically significant, it is doubtful that they are practically different.

Table 6 confirms that training the adhoc problem using $\bar{\mathbf{P}}$ is better than using $\bar{d}$, since more pairs of systems are able to successfully generalize and overall degradation is less, although the difference is slight. The table also confirms the hypothesis that adhoc is harder to train than routing, since on average, there is a degradation in performance. This is as

Table 6. Training results for adhoc.

Subset	Training method	# Combos showing some improvement on the training set	# Combos showing improvement on both training and test sets	% of improved training combos which also improve on test set	Average degradation or improvement on test set $((\bar{\mathbf{P}}/\bar{\mathbf{P}}_\mathbf{a}) - 1)$ (%)
Chorus	$\bar{\mathbf{P}}$	29	5	17	−13
Diverse	$\bar{\mathbf{P}}$	30	10	33	−9
Random	$\bar{\mathbf{P}}$	33	16	48	−3
Total/average		92	31	34	−8
Chorus	$\bar{d}$	19	5	26	−15
Diverse	$\bar{d}$	18	3	17	−12
Random	$\bar{d}$	24	13	54	−2
Total/average		61	21	34	−10

Number of system pairs (out of 45) which achieved better performance (as measured by $\bar{\mathbf{P}}$) than the better component system ($\bar{\mathbf{P}}_\mathbf{a}$). Average degradation on the test set is over all combinations counted in the third column.

expected, since it seems unlikely that a single weighting would apply across multiple queries. However, for a significant number of combinations (34%) there was an improvement. In future work, we would like to apply techniques similar to those we used in our empirical work for routing in order to predict when improvement is achievable for the adhoc task.

5. Conclusions

Our analysis of the linear combination of scores fusion model has revealed a number of important points. As a technique for combining information retrieval systems, the LC model has limitations in terms of its scope and power. Its effective use appears to be limited to situations where the systems involved have high overlap of relevant documents and low overlap of nonrelevant documents. However, we have also developed a technique which can accurately predict when improvement is possible (for the routing problem), so this limitation can be readily identified.

Our analysis has also revealed several advantages to using d as a performance criterion. Its simple algebraic form has allowed us to mathematically support conclusions about the model which previously were only empirically based, and to derive theoretically optimal weightings for pairs of IR systems. Furthermore, when used as an optimization criterion for the routing problem, d selects weights which generalize better than those chosen by optimizing average precision. On the adhoc problem, training d works about as well (or rather, about as poorly) as training average precision. Because d is differentiable and cheap to calculate, it is also a likely candidate for use in gradient based optimization procedures, which will be necessary for combinations of more than two systems or for other parameterized models (e.g., neural nets).

Our future work will involve exploring both of these possibilities. We intend to apply the LC model to combinations of more than two systems, using d or D_c to choose the weightings. We hope to be able to characterize the amount of improvement gained by adding more

systems. We will also use more sophisticated, neural network models which will select which of the component IR systems to listen to based on the document being scored. This model will be a simple version of the mixture of experts model (Jordan and Jacobs, 1994). In this way, we hope to be able to exploit not only the Chorus Effect, but the Dark Horse and Skimming Effects as well.

Appendix

An Equation for d

Our first step is to derive an equation for d when combining two systems. We use the following notation: the letters a and b are used to indicate the two systems being combined. The set of relevant documents returned only by system a are indicated $\mathcal{P}_a$ ($\mathcal{P}$ for positive example), those by b are $\mathcal{P}_b$ and those returned by both are $\mathcal{P}_{ab}$, with their sizes shown as P, so that $P_a = |\mathcal{P}_a|$, etc. The corresponding sets of nonrelevent documents and their sizes are indicated using $\mathcal{N}$ and N.

We proceed by deriving an expression for $\bar{p}$ in terms of $\bar{p}_a$ and $\bar{p}_b$, and then likewise for $\bar{n}$. As in our empirical study, we are trying to predict the performance of the combined system based on the performance (and other variables) of the component systems.

By definition,

$$\bar{p} = \frac{\sum_{x\in\mathcal{P}_{ab}} \sin(w)\rho_a(x) + \cos(w)\rho_b(x)}{P_{ab} + P_a + P_b} + \frac{\sum_{x\in\mathcal{P}_a} \sin(w)\rho_a(x)}{P_{ab} + P_a + P_b} + \frac{\sum_{x\in\mathcal{P}_b} \cos(w)\rho_b(x)}{P_{ab} + P_a + P_b}$$

or,

$$\bar{p} = \frac{\sin(w)\left[\sum_{x\in\mathcal{P}_{ab}} \rho_a(x) + \sum_{x\in\mathcal{P}_a} \rho_a(x)\right]}{P_{ab} + P_a + P_b} + \frac{\cos(w)\left[\sum_{x\in\mathcal{P}_{ab}} \rho_b(x) + \sum_{x\in\mathcal{P}_b} \rho_b(x)\right]}{P_{ab} + P_a + P_b}$$

Now, note that for system a

$$\bar{p}_a = \frac{\sum_{x\in\mathcal{P}_{ab}} \rho_a(x) + \sum_{x\in\mathcal{P}_a} \rho_a(x)}{P_{ab} + P_a}$$

or,

$$\bar{p}_a(P_{ab} + P_a) = \sum_{x\in\mathcal{P}_{ab}} \rho_a(x) + \sum_{x\in\mathcal{P}_a} \rho_a(x)$$

And likewise for system b. Substituting this into the equation for $\bar{p}$ yields:

$$\bar{p} = \frac{\sin(w)\bar{p}_a(P_{ab} + P_a)}{P_{ab} + P_a + P_b} + \frac{\cos(w)\bar{p}_b(P_{ab} + P_b)}{P_{ab} + P_a + P_b}$$

By a similar series of steps, it can be shown that:

$$\bar{n} = \frac{\sin(w)\bar{n}_a(N_{ab} + N_a)}{N_{ab} + N_a + N_b} + \frac{\cos(w)\bar{n}_b(N_{ab} + N_b)}{N_{ab} + N_a + N_b}$$

After introducing the following shorthand for the relative ratios of relevant and nonrelevant documents returned by both systems:

$$\alpha_p = \frac{P_a + P_{ab}}{P_{ab} + P_a + P_b}$$
$$\beta_p = \frac{P_b + P_{ab}}{P_{ab} + P_a + P_b}$$
$$\alpha_n = \frac{N_a + N_{ab}}{N_{ab} + N_a + N_b}$$
$$\beta_n = \frac{N_b + N_{ab}}{N_{ab} + N_a + N_b}$$

We have:

$$\begin{aligned} d &= \bar{p} - \bar{n} \\ &= \sin(w)(\alpha_p\bar{p}_a - \alpha_n\bar{n}_a) + \cos(w)(\beta_p\bar{p}_b - \beta_n\bar{n}_b) \end{aligned}$$

Note that technically, the scores need to be normalized to the $[0, 1]$ range before d is calculated. However, since we will not use this d for comparisons, and only for optimization, the scaling will not affect our conclusions and we leave it out to simplify the math.

Maximizing d

Because we have a formula for d (the performance of the combined system), we can find its maximum via calculus. Differentiating with respect to w and setting it equal to 0 yields:

$$0 = \cos(w_{\text{opt}})(\alpha_p\bar{p}_a - \alpha_n\bar{n}_a) - \sin(w_{\text{opt}})(\beta_p\bar{p}_b - \beta_n\bar{n}_b)$$

or,

$$\tan(w_{\text{opt}}) = \frac{\alpha_p\bar{p}_a - \alpha_n\bar{n}_a}{\beta_p\bar{p}_b - \beta_n\bar{n}_b}$$

For simplicity, define δ_a as the numerator of this expression and δ_b as the denominator so that $\tan(w_{\text{opt}}) = \delta_a/\delta_b$. Then,

$$\sin(w_{\text{opt}}) \propto \delta_a$$

and,

$$\cos(w_{\text{opt}}) \propto \delta_b$$

Substituting this back into the equation for d yields:

$$d_{\text{opt}} \propto \delta_a^2 + \delta_b^2 \tag{3}$$

It is important to note that this analysis applies only to the LC model when characterized using trignometric functions, and only when d is used as a performance metric. The value of the optimal weight will most likely change if either of these conditions is not met. For example, a single parameter characterization which weights the two IR systems using w and $1 - w$ would lead to a different optimal weight. This would not, however, change the fundamental nature of the equation for the optimal weight, and the same qualitative conclusions reached in Section 3.2.2 would hold.

Notes

1. Research was supported by UC Senate Bridge Grant #RW252G/B.
2. The work presented in this subsection is a summary and extension of (Vogt and Cottrell 1998a).
3. Throughout this paper, we will make reference to r^2 values. These values are a measure of the correlation of two variables, as measured by a linear regression (least-squares, straight-line fit). r^2 has a direct, simple interpretation: it is the percentage of the variance accounted for by the fitted line.

References

Bartell BT, Cottrell GW and Belew RK (1994) Automatic combination of multiple ranked retrieval systems. In: Croft WB and van Rijsbergen C, eds. SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag, Dublin, pp. 173–181.

Belkin N, Kantor P, Fox E and Shaw J (1995) Combining evidence of multiple query representations for information retrieval. Information Processing and Management, 31(3):431–448.

Boughanem M, Layaida R and Caron A (1993) A neural network model for documentary base self-organising and querying. In Proceedings of the Fifth International Conference on Computing and Information, Sudbury, Ontario, pp. 512–518.

Crestani F (1994) Comparing neural and probabilistic relevance feedback in an interactive information retrieval system. In 1994 IEEE International Conference on Neural Networks. Vol. 5, pp. 3426–3430.

Deerwester S, Dumais ST, Furnas GW, Landauer TK and Harshman R (1990) Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6):391–407.

Diamond T (1998) Information retrieval using dynamic evidence combination. Unpublished Ph.D. Thesis proposal, School of Information Studies, Syracuse University.

Egan JP (1975) Signal Detection Theory and ROC-Analysis. Academic Press.

Guttman L (1978) What is not what in statistics. The Statistician, 26:81–107.
Harman D, ed. (1995) The Third Text REtrieval Conference (TREC-3), Gaithersberg, MD. National Institute of Standards and Technology. NIST Special Publication 500-226.
Harman DK, ed. (1997) The Fifth Text REtrieval Conference (TREC-5), Gaithersberg, MD. National Institute of Standards and Technology. NIST Special Publication 500-238.
Hertz J, Krogh A and Palmer RG (1991) Introduction to the Theory of Neural Computation. Addison-Wesley, Redwood City, CA.
Jordan MI and Jacobs RA (1994) Hierarchical mixtures of experts and the EM algorithm. Neural Computation, 6:181–214.
Kantor PB (1995) Decision level data fusion for routing of documents in the TREC3 context: A best case analysis of worst case results. In: Harman, 1995.
Knaus D, Mittendorf E and Schäuble P (1995) Improving a basic retrieval method by links and passage level evidence. In: Harman, 1995.
Lee JH (1997) Analyses of multiple evidence combination. In: Belkin NJ, Narasimhalu AD and Willett P, eds. SIGIR 97: Proceedings of the Twentieth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, ACM Press, Philadelphia, pp. 267–276.
Ng KB (1998) An Investigation of the Conditions for Effective Data Fusion in Information Retrieval. Ph.D. Thesis, School of Communication, Information, and Library Studies, Rutgers University.
Press WH, Teukolsky SA, Vettering WT and Flannery BP (1995) Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press.
Selberg E and Etzioni O (1996) Multi-service search and comparison using the MetaCrawler. In Proceedings of the 4th International World Wide Web Conference, pp. 195–208.
Shaw J and Fox E (1995) Combination of multiple searches. In: Harman, 1995.
van Rijsbergen C (1979) Information Retrieval. Butterworths, London.
Vogt C, Cottrell G, Belew R and Bartell B (1997) Using relevance to train a linear mixture of experts. In: Harman, 1997, pp. 503–515.
Vogt C, Cottrell G, Belew R and Bartell B (1999) User lenses—achieving 100% precision on frequently asked questions. In Seventh International Conference on User Modeling, Banff, Canada, pp. 87–96.
Vogt CC and Cottrell GW (1998a) Predicting the performance of linearly combined IR systems. In: Croft WB, van Rijsbergen K and Wilkinson R, eds. SIGIR 98: Proceedings of the Twenty First Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, ACM Press, Melbourne, pp. 190–196.
Vogt CC and Cottrell GW (1998b). Using d' to optimize rankings. Technical Report, CS98-601, U.C. San Diego, CSE Department.
Wong S, Cai Y and Yao Y (1993) Computation of term associations by a neural network. In: Korfhage R, Rasmussen E and Willett P, eds. SIGIR 93: Proceedings of the Sixteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, ACM Press, Pittsburgh, PA, pp. 107–115.