Predicting the Performance of Linearly Combined IR Systems

Christopher C. Vogt
Computer Science and Engineering 0114
University of California, San Diego
www.cs.ucsd.edu/~vogt/

Garrison W. Cottrell
Computer Science and Engineering 0114
University of California, San Diego
www.cs.ucsd.edu/~gary/

Abstract We introduce a new technique for analyzing combination models. The technique allows us to make qualitative conclusions about which IR systems should be combined. We achieve this by using a linear regression to accurately ($r^2 = 0.98$) predict the performance of the combined system based on quantitative measurements of individual component systems taken from TREC5. When applied to a linear model (weighted sum of relevance scores), the technique supports several previously suggested hypotheses: one should maximize both the individual systems' performances and the overlap of relevant documents between systems, while minimizing the overlap of nonrelevant documents. It also suggests new conclusions: both systems should distribute scores similarly, but not rank relevant documents similarly. It furthermore suggests that the linear model is only able to exploit a fraction of the benefit possible from combination. The technique is general in nature and capable of pointing out the strengths and weaknesses of any given combination approach.

1 Introduction

Many Information Retrieval researchers have tried to improve the performance of individual systems by combining the results of multiple IR systems or queries, a technique commonly referred to as fusion. In so doing, they hope to exploit one or more of three effects enumerated by Diamond [4]:

- **The Skimming Effect** happens when "retrieval approaches that represent their collection items differently may retrieve different relevant items, so that a combination method that takes the top-ranked items from each of the retrieval approaches will push non-relevant items down in the ranking."
- **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 a single approach doing so."
- **The Dark Horse Effect** in which "a retrieval approach may produce unusually accurate (or inaccurate) estimates of relevance for at least some items, relative to the other retrieval approaches."

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. Likewise, a large Chorus Effect cuts into the possible gain from the Skimming Effect. These phenomena argue for a sophisticated combination model which is able to predict when these effects will occur and take advantage of them.

However, most current research on IR system combination focuses on much simpler combination models. One of the simplest ways to combine multiple IR systems is to merely take a linear combination of their relevance scores (also known as *RSV*'s – retrieval status values). In other words, the real-valued relevance ρ of a document d to a query q depends on the weights $\vec{w} = (w_1, w_2..)$ given to each individual IR system:

$$\rho(\vec{w}, d, q) = \sum_{systems} w_i \rho_i(d, q)$$

Or, for only two IR systems:

$$\rho(w_1, w_2, d, q) = w_1 \rho_1(d, q) + w_2 \rho_2(d, q) \qquad (1)$$

This straightforward approach can obviously take advantage of the Skimming Effect, as long as both systems are equally weighted (and assuming ρ_1 and ρ_2 have similar distributions). The Chorus Effect may be exploited in many more situations – if both systems rank relevant documents highly and both are given positive weights, or if a poorly performing system is given negative weight. The Dark Horse Effect, however, is unlikely to be exploited because the combination model does not take into account *which* document is being scored. Thus, even though one system may produce accurate scores for some documents, the optimal linear combination would not be able to take advantage of this. The weight on that system would have to be low in order to account for the remainder (and presumably the majority) of the documents – the ones for which the score was inaccurate.

Nevertheless, this approach has been used with varying degrees of success by a number of researchers (e.g., [2], [8], [9], [13], [14], and [15]). However, consistent, significant improvement has been elusive. An interesting question is: when is it even possible to improve the performance of two IR systems by linearly combining their estimates of relevance?

One study by Lee [10] has attempted to answer this question. Lee used five different combination functions with five entries on all 50 queries to the TREC3 "adhoc" task. Three of the five combination techniques which he examined are a subset of the linear combination model, but others (Min and Max) cannot be simulated by simple linear combination. However, he found that the three linear methods were generally superior to the others. Lee's hypothesis (also suggested in whole or part in [3], [12], and [2]) is that: *combination is warranted when the systems return similar sets of relevant documents but different sets of nonrelevant documents.* This basically asserts that the Chorus Effect is the primary source of potential for improvement. Lee defines two measures of the amount of overlap of relevant and nonrelevant documents (O_{rel} and O_{nonrel}, defined below). Although he does not formally analyze whether his hypothesis is correct, we have been able to verify that for the data presented in his paper, performance of the best mixture is in fact negatively correlated with O_{nonrel} (correlation of -0.73), which supports the second half of his hypothesis. Unfortunately, there is no correlation between O_{rel} and performance of the mixture, so his data does not support the first half of the hypothesis (although, as our study will show, there is reason to believe it is true).

The research presented here outlines a more comprehensive answer to the question of when it pays to combine. We describe a technique for analyzing the individual IR systems and use this analysis to predict the performance of the combined system. The technique involves measuring various properties of the individual IR systems (including the use of Lee's two measures), and using them in a linear regression to predict the average precision of the combination. By examining how the measures are weighted, we gain an intuitive feel for when combination pays off.

2 Method

Our overall approach is to examine a large number of pairs of actual IR systems, find the best possible linear combination for each, and then use various measures of the pairs to predict performance of their combinations. As used in this study, an "IR system" is actually a list of up to 1000 documents and their relevance scores for a single query. The 61 entries from TREC5's [7] "adhoc" track are used. For queries #251–#270, every pairwise combination is examined (1830 pairs per query, or 36,600 total). For each pair of IR systems *and each query*, the best linear combination is estimated by finding the weights which result in the highest average precision (precision averaged over all recall levels). Because systems are combined on a per-query basis, this experimental setup most accurately simulates the routing (fixed query, changing document collection) task, as opposed to the adhoc (any query, fixed document collection) task. Furthermore, since all of the individual systems draw upon the same document collection, this simulates the *data* fusion problem (as opposed to *collection* fusion, where each system indexes a different collection).

Since all that really matters is the *ranking* given by the combined system, only the ratio of the two weights and the relationship of the signs on the weights are important. Thus, equation (1) can be replaced by:

$$\rho(w,d,q) = \{-1,0,1\}\rho_1(d,q) + w\rho_2(d,q) \qquad (2)$$

Before combination, scores from all systems are normalized by dividing them by their respective means. Then for each query, each pair of systems, and each sign (plus, minus, or zero) on the first system, the single weight w is optimized using golden section search [11] starting with points [-50,0,50], and the best w is used to generate a combined system (of 1000 documents) according to equation (2).

One subtle issue arises when combining lists of top-ranked documents – what score should be given to documents returned by one system but not the other? We assumed that for such documents, the system which did not return them gave them a score of zero. This assumption has two unwanted side-effects. First, for systems which give negative scores, the unreturned documents get ranked above those with negative score. Luckily, of the 1220 system/query pairs, only 14 had significant numbers of negative scores. The second unwanted side-effect caused by this assumption is that it tends to amplify the Chorus Effect. This occurs because the combined list will mostly contain those documents returned by the system with higher "weight" (after taking into account the ranges of scores from the individual systems). Thus, the lower-weighted system effectively only contributes to the scores of documents in the intersection, and (for positive weights) the combination only boosts those documents. Despite these side-effects, we maintain 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.

2.1 Individual Measures

Because it seems likely that the combined performance could depend on the component systems' performances, we made two measures of the performance of each IR system individually: average precision (p_1, p_2), and a different measure of system performance (J_1, J_2). By convention, system #1 is always the one with higher average precision. J is defined as:

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

where $d \succ_q d'$ indicates the user prefers document d 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. J is a rank order statistic that measures how close an individual IR system is to the user's ranking and is correlated with average precision ([1], [15]). Note that J is simply the Guttman's Point Alienation (GPA, defined below) between an IR system and a user's relevance judgments.

2.2 Pairwise Measures

Additionally, we make a number of measures which are meant to reveal how similar the two systems are to each other, to test the hypothesis that the systems should be "different" in order to maximize the improvement in performance. The first of these is Guttman's Point Alienation (GPA) [5]. GPA is a measure of how similar two rankings are to each other, and can be calculated for any two systems ρ_1, ρ_2 and query q as:

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

The second measure we calculate is the number of documents in the intersection of the two lists of returned documents ($\cap$). The third measure is the correlation coefficient from a linear regression of the scores of documents in the intersection of the two systems (C). Note that C is actually just the r^2 value from a regression which uses one system's scores to predict the other's, but we use C to avoid notational confusion later. We

Qry 251..270	System	p 0..1	J −1..1	U 0..1
254	Cor5A2cr	0.150	0.561	0.351
254	genrl1	0.029	-0.084	0.026
264	colm1	0.064	-0.053	0.934
264	fsclt3	0.010	0.376	0.650
251	anu5aut1	0.002	-0.767	0.792
251	anu5man6	0.022	0.428	0.889

Table 1: Examples of Measures Associated with Individual Systems

Qry	Sys 1	Sys 2	GPA	GPA_{ni}	GPA_{rel}	$\cap$	$\cap_{rel}$	C	C_{rel}	O_{rel}	O_{nonrel}
251..270			−1..1	−1..1	−1..1	0..1000	0..1000	0..1	0..1	0..1	0..1
254	Cor5A2cr	genrl1	0.896	0.859	0.977	450	37	0.320	0.541	0.779	0.434
264	colm1	fsclt3	0.504	0.645	0.803	84	7	0.072	0.306	0.111	0.117
251	anu5aut1	anu5man6	-0.778	-0.307	0.700	34	5	0.110	0.218	0.145	0.030

Table 2: Examples of Pairwise Measures

also measure the number of unique relevant documents contributed by each system (U_1, U_2). U is the number of relevant documents returned by one system but not the other ($R_{i,unique}$) divided by the number total number of relevant documents returned by that system (R_i):

$$U = \frac{R_{i,unique}}{R_i}$$

As such, U is a pairwise measure, but is associated with one of the two systems in the pair. U was included because Lee's hypothesis would indicate that it should be low for systems which combine well. We also calculate Lee's overlap measures:

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

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

where R_i is the number of relevant documents returned by system i, and N_i is the number of nonrelevant. Finally, because it seems likely that measuring the similarity of the two systems on nonrelevant documents is less important than on relevant ones, the first three measures are also calculated using only relevant documents, and are denoted: $GPA_{rel}, \cap_{rel}, C_{rel}$. One last measure, GPA_{ni} (for "not irrelevant") is the GPA using pairs of documents where at least one is relevant. All of these measures indicate in varying ways how similar the two systems are to each other. Although some of these measures will be correlated with others, it is hoped that the variety is sufficient to allow prediction of the combination's performance.

2.3 Examples

Tables 1 and 2 show both the individual measures and pairwise measures for three randomly selected pairs of systems. The first pair, {Cor5A2cr, genrl1} on query 254, are similar to each other by most measures. Table 1 indicates that Cor5A2cr exhibits decent performance, as measured by both average precision and J. Also, about 35% of its relevant documents are not returned by genrl1. On the other hand, genrl1 shows relatively poor performance, and also retrieves very few unique relevant documents. One would guess from Table 1 alone that the two systems are dissimilar. However, Table 2 shows the opposite is true. All three variants of GPA are very high, indicating that both systems rank documents (both relevant and nonrelevant) very similarly. Furthermore, there are a lot of common documents in their intersection, as well as relevant documents in the intersection (query 254 has 85 total relevant documents). Their scores are somewhat correlated (C), and even more so on relevant documents. As might be guessed by the size of their intersections, they have both high overlap of relevant and nonrelevant documents.

The example tables also demonstrate a number of other points. Table 1 shows that although p and J are roughly correlated, they do measure performance in slightly different ways. In Table 2, the values of GPA and GPA_{rel} for the {anu5aut1, anu5man6} pair show that two systems can disagree strongly on most scores, and yet still agree on relevant documents (this is also somewhat discernible by examining C and C_{rel}). Finally, we note that large U values correspond to low O_{rel} – more on this later.

2.4 Analysis

We made the aforementioned measures for all 36,600 pairs of systems/queries. We then performed a multiple linear regression (using the UNIX*STAT "regress" program), using the 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. A linear regression attempts to fit a linear model to data. With one predictor variable, it fits a line, with multiple variables, a hyperplane. The coefficients of the resulting regression equation can be interpreted as indicating how much each predictor contributes to the overall estimate of the target. Thus, a large positive coefficient indicates that the corresponding predictor should be maximized in order to maximize the target. Conversely, a negative coefficient indicates the predictor should be minimized in order to maximize the target. In addition to producing a linear equation, the regression also gives an indication of how significant the correlation is between each predictor and target in the form of an F value (larger means more significant). It also reports on how good the fit is using the r^2 measure, a value in [0,1] which measures the percentage of the

Measure	Normalized Regression Coefficient	F
p_1	0.8993	129141.5501
U_1	-0.1202	405.5097
U_2	-0.0401	393.1853
J_2	0.0431	346.1357
J_1	0.0308	241.5460
GPA_{rel}	-0.0359	220.1937
p_2	-0.0232	99.0202
O_{rel}	-0.0519	55.8835
C_{rel}	0.0125	35.8910
GPA	0.0137	22.6715
O_{nonrel}	-0.0427	20.9289
$\cap_{rel}$	0.0088	17.5199
GPA_{ni}	-0.0099	8.9850
$\cap$	-0.0149	2.3284
C	0.0023	1.2025

Table 3: Results of Linear Regression for Predicting Combination's Average Precision (r^2=0.94)

variance in the data accounted for by the linear model. Finally, the actual coefficients of the regression equation are normalized based on the distributions of the individual predictors, so that their magnitude can also be compared.

3 Results and Discussion

Table 3 presents the results of the multiple 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). 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, $r^2 = 0.98$. This can be seen graphically in Figure 1, which plots the actual average precision versus the predicted value on the test set. The clustering of points around the line $y = x$ indicates a good fit.

p_1, J_1, p_2, J_2 : The upper part of Table 3 indicates that in order to maximize average precision, one IR system must be very good (the normalized coefficient for p_1 is positive and much larger than any other, and J_1's coefficient is also positive) but the second IR system should also be good (J_2's coefficient is positive). Examining the actual combined systems supports this conclusion: the six best mixtures of IR systems (when averaged over all 20 queries) are all comprised of systems which are ranked among the top 10 individually. However, this rule is not hard and fast: mixtures {ibms96b, uwgcx0}, and {ibmgel, ETHme1} are ranked 7th and 12th (out of 1830 possible mixtures) and yet they make use of relatively poor systems (ibmgel is ranked 32nd and ibms96b is ranked 46th out of 61 systems). Thus, even with one poor IR system, we can get significant improvement. In fact, the negative coefficient on p_2 indicates that we may *want* the second system to perform poorly, when performance is defined by average precision. This conclusion is also supported by another linear regression presented in the next paragraph.

Measure	Normalized Regression Coefficient	F
p_1	0.9366	191543.1029
O_{rel}	0.1021	2249.4031
O_{nonrel}	-0.0581	975.4101
p_2	-0.0228	119.1705

Table 4: Results of Linear Regression for Predicting Combination's Average Precision (r^2=0.94)

$U_1, U_2, O_{rel}, O_{nonrel}$: Another interesting conclusion from Table 3 is that maximal precision can be achieved by minimizing the percentage of relevant documents which are unique to each system (U_1 and U_2 have negative coefficients). This indicates exploitation of the Chorus Effect. One would also expect that Lee's O_{rel} should be maximized, a conclusion not supported by the table. A simple analysis explains this anomaly. It can be shown that:

$$\frac{1}{2O_{rel}} = \frac{1}{1-U_1} + \frac{1}{1-U_2}$$

Thus, minimizing U_1 and U_2 maximizes O_{rel}. In fact, if the regression is repeated without U_1 and U_2 as predictors, the resulting model is just as good ($r^2 = 0.94$), and O_{rel} has the second highest F value (after p_1), with a large, positive coefficient. Thus, the original regression mislabelled O_{rel} simply because it was redundant information. The large contribution of O_{rel} provides support for the first part of Lee's hypothesis, that both systems should return the same relevant documents. The second part of Lee's hypothesis – that the two systems should retrieve different sets of nonrelevant documents – would suggest a negative coefficient on O_{nonrel}, which is indeed what is observed. In fact, by repeating the regression using only p_1, p_2, 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 4).

$GPA, GPA_{rel}, GPA_{ni}, C_{rel}, \cap_{rel}$: The positive coefficient on GPA indicates that this measure should be maximized. In other words, 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. Knaus, et al. [9] have indicated that problems may occur when linearly combining systems that distribute RSV's differently, so this result is not surprising, and is also supported by the positive coefficient on C_{rel}. On the other hand, the negative coefficients on GPA_{rel} and GPA_{ni} indicate that these measures should be minimized. In other words, each system should rank *relevant* documents differently than the other system. Here again, we see the Chorus and Skimming Effects in play. By preferring those systems that mix nonrelevant documents in with relevant documents differently (low GPA_{ni}), nonrelevant documents will get pushed down the list. By preferring systems that rank relevant documents differently (low GPA_{rel}), we can be assured that no relevant document has a low score from both experts, a situation which would allow it to get lost in the noise of nonrelevant documents. Finally, we note that the positive coefficient for $\cap_{rel}$ also supports the first

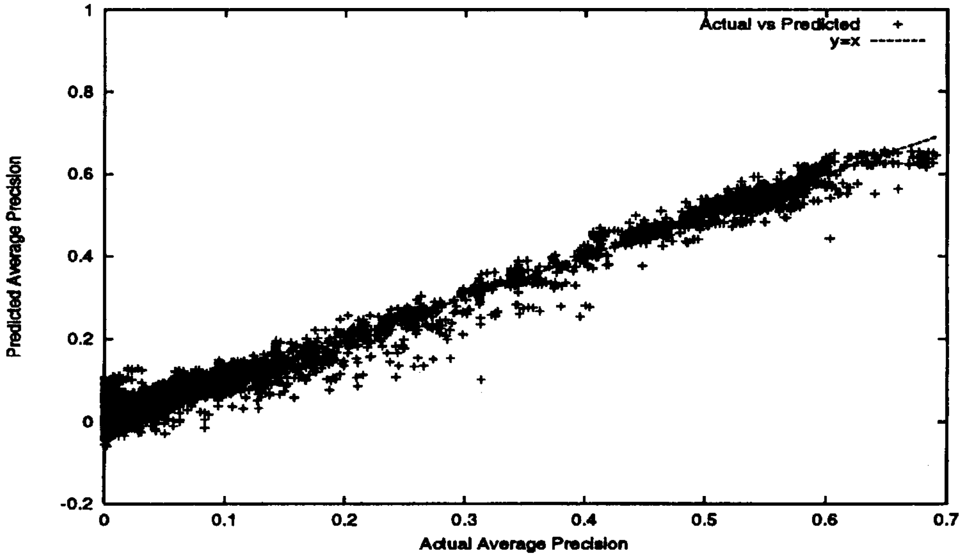

Figure 1: Actual Average Precision versus Average Precision Predicted by the Regression Model on the Test Data

part of Lee's hypothesis, that the number of relevant documents returned by both systems should be maximized.

In summary, the best time to linearly combine two IR systems is when:

- at least one exhibits good performance,
- both return similar sets of relevant documents,
- both return dissimilar sets of nonrelevant documents,
- both distribute scores similarly, but
- both do not rank relevant documents in a similar fashion,

with the first three points being the most important.

The discussion of U_1, U_2 vs. O_{rel} above, and the inclusion of a second regression (in Table 4), 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.

Other Observations

The methodology used in the above experiment provides us with a large set of systems which are nearly optimally combined (using the linear model). A quick analysis of these systems leads to a number of conclusions which have implications for the fusion problem in general.

1. **Even with this simple model, we can often achieve improvement.** For 88% of the pairs, some improvement is possible. For 50% of the pairs, improvement is at least 5% over the better of the two systems, and for 11% of the pairs, improvement is at least 50%. The median improvement is 5%.
2. **By choosing wisely, we can beat any system.** On a per-query basis, 5% of the pairs beat the best individual IR system for that query. Most of these (60%) did not include the best system in the combination.
3. **Queries with fewer relevant documents may have a greater chance for large improvement.** Figure 2 shows that when there are few relevant documents, the ratio of improvement (the average precision of the combination divided by the average precision of the better system: $\frac{p_{combo}}{p_1}$) is much more varied, and capable of being much higher than for queries with many relevant documents. This seems logical, since with small numbers of relevant documents, changes in document rankings can greatly affect the value of average precision.
4. **Expertise is often query dependent.** Only 53% of the pairs of IR systems have one IR system which is consistently better than the other (i.e., better on over 70% of the queries).
5. **Weights are often predicted by performance.** About 88% of the time, p_1 and p_2 are ordered the same as the best weights.
6. **Negative weights are common.** About 34% of the 36,600 query/system triplets have an optimal weighting scheme where at least one system is negatively weighted. 1% have *both* systems negatively weighted.

The first two points provide support for the use of fusion models in general – even when the underlying systems have very high performance. The third point illustrates the influence of the characteristics of each individual query on the ability to combine systems and, along

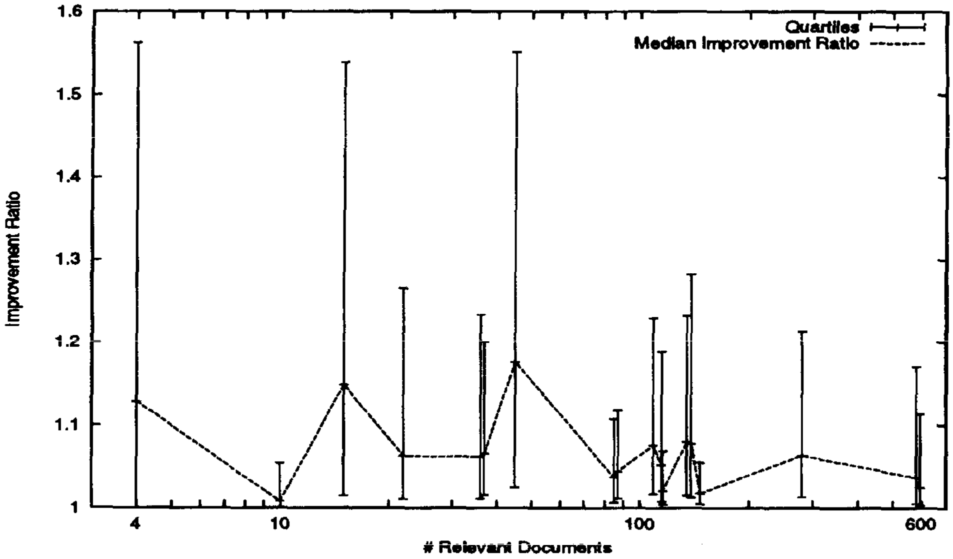

Figure 2: The Median Improvement Ratio ($\frac{p_{combo}}{p_1}$) as a Function of the Number of Relevant Documents. Error bars are one quartile.

with the fourth point, argues that solving the routing problem with a fusion system may be easier than solving the adhoc problem. The fifth point indicates that the simple heuristic of weighting the better system higher may be sufficient for achieving some improvement. Furthermore, the last point indicates that it is often the case that a system's judgement of which documents are relevant is exactly wrong. In light of the fifth point, it seems likely that the negatively weighted systems are poor performers, so subtracting their scores effectively downweights the nonrelevant documents near the head of the list which are also in the intersection, thus improving combined performance. The last point also illustrates the importance of negative weights – previous implementations of the linear model have generally ignored the possibility of negative weights, but here we see that they are clearly desirable.

Limitations of Linear Combination

The above analysis hints that the linear model primarily exploits only the Chorus Effect. Perhaps this effect is the only one worth exploiting, and thus we need not consider more complicated models. One way to illustrate the effectiveness of a combination model is to compare it to the theoretically optimal fusion. The optimal fusion would rank all of the relevant documents in the union of the two systems above all nonrelevant documents. As it turns out, such a system would have an average precision equal to its total recall. If we compare the ratio of each combined system's average precision to its optimal ($\frac{p_{combo}}{p_{opt}}$), we find that the average value for this ratio over all query/system-pair combinations is 0.34, with standard deviation 0.27, and median 0.28. Thus, despite the occasional impressive gains in performance reported above, the linear model only achieves about one third of the theoretically optimal performance. Although it is improbable that the optimal performance is actually consistently achievable, there is nevertheless much room for improvement, presumably by using a more sophisticated combination model.

4 Conclusions

We have introduced a general technique for analyzing the behavior of combination models which allows us to predict with extraordinary accuracy the performance of a combined system based on measurable characteristics of the component systems. We have applied this technique to a linear combination model. The main conclusions – maximize individual performance and the overlap of relevant documents while minimizing the overlap of nonrelevant documents – are in agreement with previous theories, specifically those put forth in [10]. The analysis also elicits two other hypotheses for explaining when it makes sense to linearly combine systems: both systems should distribute scores similarly, but not rank relevant documents similarly.

It should be noted that these conclusions are for the linear model only. Other, more sophisticated combination techniques would most likely take advantage of more than just the Chorus Effect, and thus require a new analysis. However, we stress that our analysis technique (computing a regression using measurements of the component systems) is not limited to analyzing a linear combination model – it is capable of pointing out the strengths and weaknesses of any given combination approach. Obviously, the particular measures used as input to the regression may include ones besides those used here, and should be chosen based on some knowledge of how the combination model works. Finally, we note again that our experiment was limited to the routing, data fusion setting.

5 Future Work

Our analysis begs the question of how to determine the best weights for a linear model on two fronts. First of all, the golden section optimization technique only works for models with a single parameter. If we were to combine more than two systems, we would have to use another technique. More importantly, we have no idea whether our "optimal" weights would generalize to new data. Our current work involves exploring techniques for training the linear model in order to guarantee generalization to new data, as well as to be able handle multiple-parameter models. These techniques range from direct optimization of average precision, to gradient-based techniques based on optimizing J, which is correlated with average precision.

As the above analysis indicates, the linear model primarily takes advantage of the Chorus Effect. Our future work will involve more sophisticated (neural network) models which can exploit the Skimming and Dark Horse Effects by using representations of the documents and queries as inputs to the combination model. Our hope is that the training techniques we develop for the linear model will generalize to these more complicated models.

Our current work focuses on combining two systems. Previous studies [3] have found that "more is better" when it comes to the number of systems. We intend to verify this conclusion using a methodology similar to that used here. Ultimately, we hope our work will shed some light on why combination works, and when it works best.

Acknowledgements This research was supported by NSF grant IRI 92-21276 and UC Senate Bridge Grant #RW252G/B. Thanks go to Matt Dailey and the rest of GURU for their useful comments.

References

[1] Brian T. Bartell. *Optimizing Ranking Functions: A Connectionist Approach to Adaptive Information Retrieval.* thesis, Department of Computer Science and Engineering, The University of California, San Diego, CSE 0114, La Jolla, CA 92093, 1994.

[2] Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew. Automatic combination of multiple ranked retrieval systems. In W. Bruce Croft and C.J. van Rijsbergen, editors, *SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin, 1994. Springer-Verlag.

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

[4] Ted Diamond. Information retrieval using dynamic evidence combination. PhD Dissertation Proposal. http://www.nwlink.com/~tgdiamon, Oct 1996.

[5] L. Guttman. What is not what in statistics. *The Statistician*, 26:81–107, 1978.

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

[7] D.K. Harman, editor. *The Fifth Text REtrieval Conference (TREC-5)*, Gaithersberg, MD, 1997. National Institute of Standards and Technology. NIST Special Publication.

[8] Paul B. Kantor. Decision level data fusion for routing of documents in the TREC3 context: A best case analysis of worst case results. In Harman [6]. NIST Special Publication.

[9] Daniel Knaus, Elke Mittendorf, and Peter Schäuble. Improving a basic retrieval method by links and passage level evidence. In Harman [6]. NIST Special Publication.

[10] Joon Ho Lee. Analyses of multiple evidence combination. In Nicholas J. Belkin, A. Desai Narasimhalu, and Peter Willett, editors, *SIGIR 97: Proceedings of the Twentieth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 267–276, Philadelphia, 1997. ACM Press.

[11] William H. Press, Saul A. Teukolsky, William T. Vettering, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1995.

[12] T. Saracevic and P. Kantor. A study of information seeking and retrieving III. *Journal of the American Society for Information Science*, 39(3):197–218, 1988.

[13] Erik Selberg and Oren Etzioni. Multi-service search and comparison using the MetaCrawler. In *Proceedings of the 4th International World Wide Web Conference*, 1996.

[14] J.A. Shaw and E.A. Fox. Combination of multiple searches. In Harman [6]. NIST Special Publication.

[15] C.C. Vogt, G.W. Cottrell, R.K. Belew, and B.T. Bartell. Using relevance to train a linear mixture of experts. In Harman [7]. NIST Special Publication.