Fusion Via Linear Combination for the Routing Problem

Christopher C. Vogt, Garrison W. Cottrell
University of California, San Diego
Computer Science and Engineering 0114
La Jolla, CA 92093
{vogt,gary}@cs.ucsd.edu

February 6, 1998

Abstract

A linear combination of scores from two different IR systems is used for the routing task, with one combination model being trained for each query. Despite a poor selection of component systems, the combination model performs on par with the better of the two systems, learning to ignore the worse system.

1 INTRODUCTION

Our work this year followed up on our TREC5 fusion approach – a linear combination of relevance scores (a.k.a. RSVs) from different IR systems. Last year's *adhoc* entry successfully improved performance over all three component systems. However, our *routing* entry did not show an improvement, but rather a degradation in performance when compared to the best individual system. We attributed these disappointing results to three possible factors: overfitting, a weak combination model, or the fact that we used a single set of model parameters for all routing queries rather than training a separate model to each query. This year's entry addressed the last factor by using the same linear model and similar training method, but customizing an individual model for each query.

Our participation was in the Category A, routing task.

2 METHOD

A linear combination model is used to compute the weighted sum of scores from two IR systems. Since we are in the routing context, we can effectively ignore the query as an input, thus the score R for a particular document d on a particular query q is computed as:

$$R_q(w,d) = R_{q,1}(d) + wR_{q,2}(d) \quad (1)$$

A single weight w is used instead of two, because all that matters is the *ranking* of documents, and thus only the ratio of weights. w is scanned from 20 to $\frac{1}{20}$ in multiplicative increments of 0.95, with scores from each system pre-normalized by dividing by the respective averages. This normalization allows the above technique of scanning the weight to effectively cover all possible different combinations even though it only covers a small interval of possible weights. Negative weights were not examined. The w which maximizes average precision on the training set is selected for each query. Our entry into last year's TREC optimized a different criterion, J, which measures how close the combined system's ranking is to the user's, and is highly correlated with average precision. Since we have only one parameter in our model this year, it was computationally feasible to optimize average precision directly.

The two "systems" used were both variants based on Verity Inc.'s routing submissions (due to the first author's affiliation with Verity over the summer). These systems make use of Verity's "Query By Example" (QBE) functionality, which generates a query in Verity's rich VQL query language based on positive and negative document examples. The first system used was a version trained primarily on documents from the same source as the test set (FBIS) which varied how many of the top-ranked documents and terms from these documents were used to construct the query, choosing different numbers of examples and terms for each query. The second system used a constant number of terms (15) and used all possible positive training examples (regardless of which collection they came from) and no negative examples, regardless of the query. We will refer to first system as the source-specific system because of its emphasis on FBIS documents, and the second as the fixed system, because it did not optimize the parameters (number of documents and terms) of the QBE query generator.

3 RESULTS and DISCUSSION

Figure 1 shows the precision/recall graph for each individual system and the combined system. Verity's run is included for comparison because it also used a fusion approach. Verity's approach was to choose whichever of two different systems performed better on the training set on a per query basis. One of the two systems Verity used was the same as our source-specific system. Also shown are results for the best possible fusion using the linear model, found by optimizing the weight using the *test* data. This indicates an upper bound on achievable performance.

The interesting points to note about the graph are:

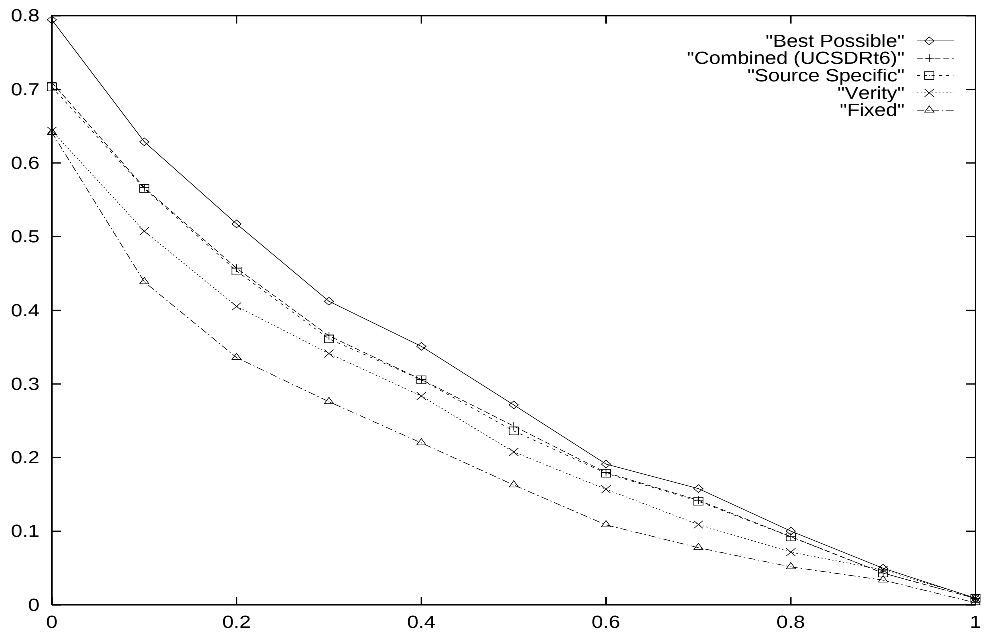

Figure 1: Precision/Recall Graph of Test Set Performance

1. The combination system has performance virtually identical to the better (source-specific) of the two systems.

2. The combination system does not achieve the best possible linear performance (given knowledge of the test set).

3. Verity's fusion did not fare as well as ours, underperforming the better of its two component systems (the source-specific system).

The first point is disappointing, since previous work ([Belkin et al., 1995], [Lee, 1997], [Bartell et al., 1994], etc.) has shown that the combination should *outperform* the best individual system. Closer examination of the model reveals why. Examination of the weight used in the combination for each query shows that in 29 of the 47 queries (62%), the fixed system received *zero* weight. Furthermore, in 76% of the queries, the fixed system's contribution was less than 10%, and for all but one of the queries, the source-specific system was weighted more heavily. Considering this, the similar performance of the source-specific system and the combined system is not surprising, but raises the question of why one system is always weighted more heavily than the other.

Work which we have done concurrently ([Vogt, 1997]) may shed some light on this – it appears that the problem is which systems we chose to combine. Our work indicates that the best time to linearly combine systems is when they

a) both have performance of similar magnitudes and b) rank relevant documents differently. For our training data, nearly 80% of the queries exhibited a difference in magnitude of performance (average precision) of 0.1 or more, with the average being 0.2. Of those 10 queries in which both systems did exhibit similar performance, only 2 ranked relevant documents differently[1]. Thus, it appears we were attempting to combine systems which had little potential for improvement.

The best possible combination line shown in Figure 1 shows that a linear model could indeed significantly outperform the source-specific system, with an average precision of 0.32 versus 0.28. However, it's doubtful that this combination is achievable using the available training data. In fact, on the training set, the source-specific system had average precision of 0.40 and the mixture weighed in at 0.41, a very small difference.

However, we note that our model and training technique apparently were able to generate a combination which was at least as good as the better system. In fact, on all but one query, the combined system's performance on the test set was identical to the source-specific system's. Thus, our training technique was able to recognize that the fixed system generally could not contribute much and therefore to ignore it.

The precision/recall graph shows that our technique is better than the system selection approach used by Verity, which achieved performance somewhere between its two component systems (Verity's second system is not shown, but performs slightly better than the fixed system). However, as Verity points out in its TREC report, their fusion approach was one which has not generally done well in the past (system selection), so perhaps this comparison is not very informative.

4 CONCLUSIONS and FUTURE WORK

Our results show that, unlike last year's entry, training one model per query results in a system at least as good as the best expert. However, no major improvement over the best expert was obtained. Again, we believe this was due to combining a good expert with a poor one, and since this technique has generally proven effective for other IR researchers, we maintain interest in pursuing this approach.

The linear combination model is theoretically capable of performing much better than our entry did (about 14% at low recall levels). This may be due to insufficient training data, outliers, or an inadequate training method. For example, we optimized performance on the training set, rather than using a hold–out set to stop training, a technique which should give us better generalization. This approach, and training on only the top-ranked documents to

[1]As measured by GPA_r, the Guttman's Point Alienation calculated using only relevant documents – see [Vogt, 1997]

avoid outliers, are two techniques we are currently investigating. Several other issues, such as the use of negative weights and score normalization are also part of ongoing research on the linear model.

As noted above, the inability to improve on the better system may be due to the particular systems we chose to combine. Our ongoing work is investigating this by looking at combinations of a broad spectrum of different IR systems (the actual entries from past TRECs), thus allowing more serendipitous combinations to be found. In addition to the linear model, we are investigating neural network models which are capable of implementing a broader range of combination functions and taking query and document representations into account.

Perhaps the most interesting conclusion from this work is that our training technique this year, which included per-query training, was robust to a poor selection of component systems. Given a pair of experts which were unlikely to be combinable, the training process was able to identify the "bad" system and ignore it.

5 Acknowledgments

The authors would like to thank Dominic Lobbia for all the hard work he did in the initial stages of our work for TREC.
This research was supported by NSF grant IRI 92-21276.

References

[Bartell et al., 1994] Bartell, B. T., Cottrell, G. W., and Belew, R. K. (1994). Automatic combination of multiple ranked retrieval systems. In Croft, W. B. and van Rijsbergen, C., editors, *SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin. Springer-Verlag.

[Belkin et al., 1995] 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.

[Belkin et al., 1997] Belkin, N. J., Narasimhalu, A. D., and Willett, P., editors (1997). *SIGIR 97: Proceedings of the Twentieth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia. ACM Press.

[Lee, 1997] Lee, J. H. (1997). Analyses of multiple evidence combination. In [Belkin et al., 1997], pages 267–276.

[Vogt, 1997] Vogt, C. C. (1997). When does it make sense to linearly combine relevance scores? In [Belkin et al., 1997]. Poster Session. UCSD CSE Tech Report CS97-556.