Probability-Based Fusion of Information Retrieval Result Sets

D. Lillis, F. Toolan, A. Mur, L. Peng, R. Collier, and J. Dunnion

Department of Computer Science
University College Dublin
Ireland
{david.lillis,fergus.toolan,mur.angel,
peng.liu,rem.collier,john.dunnion}@ucd.ie

Abstract. Information Retrieval (IR) forms the basis of many information management tasks. Information management itself has become an extremely important area as the amount of electronically available information increases dramatically. There are numerous methods of performing the IR task both by utilising different techniques and through using different representations of the information available to us. It has been shown that some algorithms outperform others on certain tasks. Very little progress has been made in fusing various techniques to improve the overall retrieval performance of a system. This paper introduces a Probability-Based Fusion technique *probFuse* which shows initial promise in addressing this question. It also compares *probFuse* with the common CombMNZ data fusion technique.

1 Introduction

Numerous Information Retrieval models have been proposed to solve the problem of identifying documents in a collection that are relevant to given queries. In recent years, much research has been conducted into what has become known as *data fusion* or *collection fusion* [1]. Data fusion involves the combination of results from different sources, using any information that is available, in order to obtain results which are superior to those of any of the individual sources.

In order to achieve this, a number of solutions have been proposed to achieve high-performance data fusion. Some of these rely on the relevance scores provided by the individual retrieval sources, some make use of the ranking of the individual result sets alone and others introduce weighting to create a bias to favour some sources over others. In many cases, such research has been in the context of metasearch engines [1], which involve the fusion of result sets produced by distinct, autonomous IR systems.

This paper is organised as follows: in section 2, we provide a brief overview of some of the approaches that have been taken by others in solving the data fusion problem in the past. Section 3 details the problem in question. In section 4 we

introduce the *probFuse* algorithm, a probability-based approach to data fusion. Section 5 describes the results of running *probFuse* on a number of collections, along with a comparison with the popular CombMNZ fusion technique. Finally, section 6 deals with conclusions and future work.

2 Prior Work

An early, simple method of merging distinct result sets is to interleave the results in round-robin fashion [1], whereby the first-ranked documents are placed at the beginning of the merged set, followed by the second-ranked documents and so on. The effectiveness of this method is largely dependent on the rather naive assumption that each server returns results of equal quality and an empirical study [2] demonstrates a 40% degradation in effectiveness when compared to the performance of a single centralised collection.

A number of later approaches rely on the relevance scores assigned by each retrieval technique to each document in order to rank those documents appropriately [3] [4]. The relevance scores returned by each IR model are not necessarily comparable in their raw form, since each will typically return scores in different ranges. In order to compare these scores in a meaningful way, it is necessary to normalise them, so that they lie within a common range.

A number of fusion techniques based on normalised scores were proposed by Shaw and Fox [5]. These included CombSUM, in which the ranking score for each document is the sum of the normalised scores returned by the individual techniques, and its variant CombMNZ, which introduces a bias in favour of documents which are judged relevant by a higher number of individual techniques. CombMNZ has become the standard data fusion technique [6] [7], as it has been shown to outperform the other techniques they proposed. In particular, Lee [8] was able to achieve significant improvements by using CombMNZ.

A Linear Combination model has been used in a number of studies [9] [10]. Here, each individual source is assigned a weight, based on past performance. Each document's ranking score is then calculated based both on this weight and the estimation of relevance it receives from each source. Vogt and Cottrell made use of training methods to find optimal values for these weights.

Another training-based technique is proposed by Voorhees et. al. in [2]. For each query, they assigned a weight to each seperate collection based on the prior performance of clusters of similar queries. This allowed them to select more documents from the result set returned by the collection with the highest weighting.

Montague and Aslam have developed the *Borda* [7] and *Condorcet* [11] voting-based fusion techniques. They make use of two algorithms that were developed in the 18th century to address shortcomings in the straight vote system for elections in where there were more than two candidates. Applying these algorithms

to fusion they were able to achieve improved results using the document rankings alone, ignoring estimations of relevance returned by the underlying sources. They also produced a weighted variation of each technique, which, like other weighted techniques, uses training data on past performance to calculate the appropriate weights.

Beitzel et. al. [6] argue that the task of fusing result sets from different techniques within the same system is different to the meta search task. They claim that CombMNZ's effectiveness is largely attributable to differences between the autonomous IR systems, such as different stopword lists, different stemming algorithms and relevance feedback. In addition, they argued that Lee's improvements were likely to have arisen because of an increase in overall recall, given that his approach was specifically designed to retrieve documents of different types. Therefore, they claim that CombMNZ's use for fusing result sets produced by the same IR system is limited.

3 Problem Description

The characteristics of fusion are outlined by Vogt and Cottrell [9]. If the individual sources are retrieving different documents, this is likely to increase recall (the fraction of total relevant documents that have been retrieved). They describe this as the "Skimming Effect", as a fusion technique would "skim" the top-ranked documents from each result set, since the highest density of relevant documents is most likely to appear there. They also describe the "Chorus Effect", in which several retrieval sources are in agreement that a document is relevant. In situations where this agreement is correct, fusion techniques which attach a greater significance to documents which are common to multiple sources will perform well. This has been shown to have a significant effect by the research involving the CombMNZ algorithm.

They also identify a "Dark Horse Effect", in which one retrieval approach returns results of a much different quality than the others. This may either be the returning of unusually accurate or inaccurate relevance judgments. Vogt and Cottrell note that the Chorus and Dark Horse effects are somewhat contradictory in nature, with the former encouraging fusion techniques to take as many sources into account when fusing and the latter suggesting that a single technique may provide the best performance.

If we have a system in which we use multiple IR models, it is likely that different models will perform better on different queries. In addition, it is unlikely to be possible to identify which technique will produce the best performance on any specific query. For these reasons, it is desirable to be able to combine the results returned by each model in order to achieve results that are superior to any of the individual techniques. An acceptable minimum performance level would be to match the best performing technique for each query. When evaluating our *probFuse* algorithm in section 5, we use the maximum precision achieved by any

single technique at each point of recall as the benchmark to be improved upon. An ability to improve upon this benchmark supports the case in favour of fusion, rather than merely creating an algorithm to attempt to select the best individual technique for a given query.

4 Probability-Based Fusion

In this section, we describe *probFuse*, a probability-based approach to fusing results from different Information Retrieval models within the same system. Using this approach, each document contained in any of the individual result sets to be fused is assigned a score, based on its probability of relevance, which is used in ranking the documents in the final, fused result set.

In order to calculate this probability, each result set is divided into x segments. Using a training set comprising $t\%$ of the queries available, the probability of relevance for each segment must be calculated.

In a training set of Q queries, $P(d_k|m)$, the probability that a document d returned in segment k is relevant, given that it has been returned by retrieval model m, is given by:

$$P(d_k|m) = \frac{\sum_{q=1}^{Q} \frac{|R_{k,q}|}{|k|}}{Q} \tag{1}$$

where $|R_{k,q}|$ is the number of documents in segment k that are relevant to query q, and $|k|$ is the total number of documents in segment k.

This probabilty should be calculated for each segment in each retrieval model.

The ranking score S_d for each document d is given by

$$S_d = \sum_{m=1}^{M} \frac{P(d_k|m)}{k} \tag{2}$$

where M is the number of retrieval models being used, $P(d_k, m)$ is the probability of relevance for a document d_k that has been returned in segment k in retrieval model m, and k is the segment that d appears in (1 for the first segment, 2 for the second, etc.). For any technique that does not return document d in its result set at all, $P(d_k|m)$ is considered to be zero, in order to ensure that documents do not receive any boost to their ranking scores from techniques which do not return them as being judged relevant.

Using the segment a document is returned in, rather than the specific rank, recognises that different queries will likely result in result sets of varying lengths, depending on how common the terms in the query are. For example, a document ranked 10th in a 20-document result set is less likely to be relevant than the 10th in a 200-document result set.

This approach strives to balance the three effects identified by Vogt and Cottrell. Firstly, by considering the probability of relevance, we make use of the Dark Horse effect, by attaching a greater importance to techniques which are more likely to return relevant documents in particular segments. By using the sum of the scores from each individual technique, rather than the maximum, we make make use of the Chorus effect. Finally, the division by k attaches a greater weight to documents returned near the beginning of the result set, where retrieval techniques will typically have their highest density of relevant documents (Skimming Effect).

5 Experiment and Evaluation

In this section, we describe a number of experiments which were run in order to test the effectiveness of the *probFuse* algorithm. Firstly, we use various training set sizes and x values (the number of segments each result set should be divided into) in order to find optimal values for each. Once these have been identified, we compare the results with that of Shaw and Fox's CombMNZ algorithm.

The experiments were run over four document collections: Cranfield, LISA, NPL and Med. The characteristics of each collection are outlined in Table 1. Initially, the queries for each collection were arranged in a random order. Once this was done, this order was maintained for each experimental run, in order to eliminate inconsistencies of results due to a change in the ordering of the queries. We then obtained the result sets to be fused using three Information Retrieval models: the Vector Space Model [12], the Extended Boolean Model [13] and the Fuzzy Set Model [14]. We then ran *probFuse* on each, using various training set sizes and x values.

Collection	Documents	Queries
Cranfield	1,400	225
LISA	5,872	35
Med	1,033	30
NPL	11,429	93

Table 1. Characteristics of Document Collections Used

The training set sizes used ranged from 10% to 90% inclusive, in intervals of 10 percentage points. For each of those training set sizes, we ran *probFuse* with x values of 2, 4, 6, 8, 10, 20, 30, 40 and 50.

In order to evaluate the performance of our experiments, we firstly calculated the interpolated precision at the 11 standard recall levels [14] (0% to 100% inclusive, at intervals of 10 percentage points) for the result set returned for each document collection by each individual retrieval model and also for the

fused result set. Once this is done, $\overline{\Delta P_c}$, the mean difference in precision for collection c is given by

$$\overline{\Delta P_c} = \frac{\sum_{r=1}^{R} P_{f,r} - MAX(P_{c,r})}{R} \tag{3}$$

where R is the number of standard recall levels, $P_{f,r}$ is the precision of the fused result set at recall level r and $MAX(P_{c,r})$ is the maximum precision obtained by any single retrieval model on collection c at recall level r. The single value used in Figures 1 and 2 is the average $\overline{\Delta P_c}$ across all four collections.

Figure 1 shows the change in average precision for the various values of x and t with each line representing a particular training set size. The poorest-performing training set sizes are 10% and 90%, demonstrating that training set sizes that are either very large or very small will lead to poor performance. Using a training set size of 50% results in the best performance for all but one value of x.

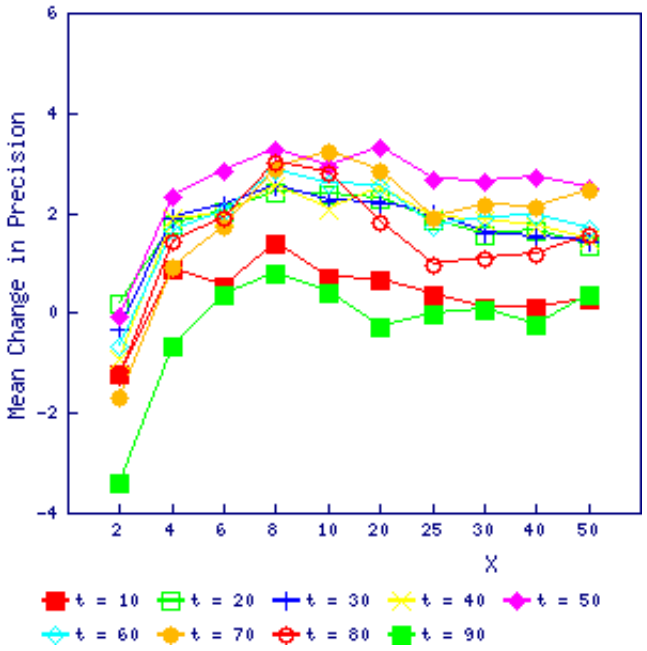

Fig. 1. Mean difference in precision for different training set sizes

In Figure 2, each line represents the change in average precision for a particular value of x. The worst-performing x value is 2. At this value, probability of relevance is assigned to a document based on whether it appears in the first

half or the second half of a result set. Increasing values for x produce superior results, to a point, with x values of 10 and 20 showing the highest mean precision increase.

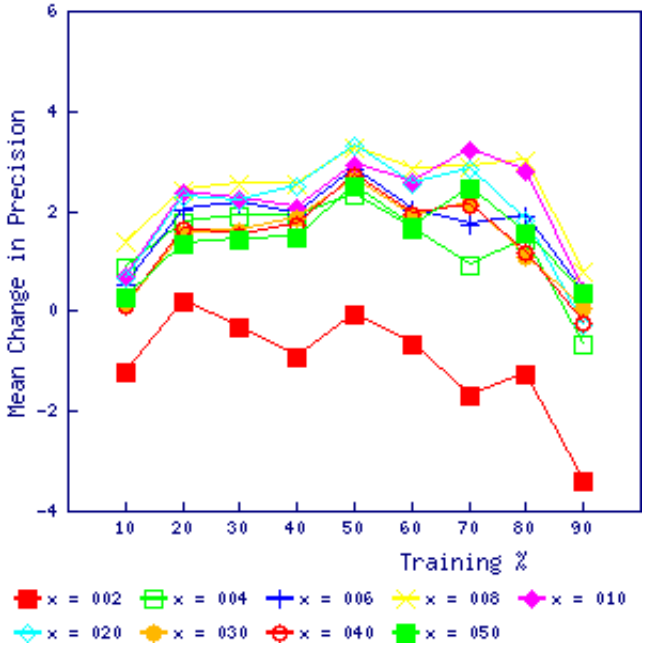

Fig. 2. Mean difference in precision for different values of x

From these two graphs, we can see that the best performance is achieved using a training set size of 50% and dividing each result set into 20 segments.

Having identified the best perfoming combination of x and t values, we then performed a comparison of those results and the CombMNZ algorithm. The CombMNZ algorithm is based on the relevance scores assigned to each document by each retrieval model. However, the raw scores returned by each model are not necessarily directly comparable, so it is necessary to normalise them. Lee's implementation of CombMNZ normalised scores using

$$normalised\_sim = \frac{unnormalised\_sim - min\_sim}{max\_sim - min\_sim} \tag{4}$$

where max_sim and min_sim are the maximum and minimum score, respectively, that are actually seen in the retrieval result. Once the scores have been normalised, the $CombMNZ_d$, the CombMNZ ranking score for any document d is given by

$$CombMNZ_d = \sum_{s=1}^{S} N_{s,d} * |N_d > 0| \quad (5)$$

where S is the number of result sets to be fused, $N_{s,d}$ is the normalised score of document d in result set s and $|N_d > 0|$ is the number of non-zero normalised scores given to d by any result set.

	probFuse	CombMNZ
Cranfield	+1.92**	-1.48*
LISA	+3.09**	+2.24
Med	+3.48	+3.07
NPL	+4.80**	+4.13**
Max	+4.80	+4.13
Min	+1.92	-1.48
Avg	+3.32	+1.99

Table 2. Comparison of the mean difference in precision achieved by the *probMerge* and CombMNZ algorithms for each collection. Entries with a "*" are significant for a significance level of 5%. Entries with a "**" are significant for a significance level of 1%, as calculated by the Wilcoxon test

Table 2 shows a comparison in the mean difference in precision for *probFuse* and CombMNZ, where *probFuse* uses a training set of 50% and an x value of 20. As the first half of the collection is being used solely as training data by *probFuse*, we have ignored it for the purposes of CombMNZ, so that we are comparing the two algorithms' performance over the same queries. The table shows the mean difference in precision both for each collection individually and as an overall average. The table shows us that *probFuse* outperforms CombMNZ on each collection, and that the use of CombMNZ actually causes a significant reduction in performance when applied to the Cranfield collection. For all collections except Med, *probFuse* shows highly significant improvements over the maximum precision values of the individual techniques. In contrast, CombMNZ only achieves significant improvements for the NPL collection.

Figure 3 illustrates the performance of *probFuse* and CombMNZ on the Cranfield collection. It shows the interpolated precision at the standard recall levels for each individual technique, as well as for each fusion technique.

6 Conclusions and Future Work

In this paper, we have proposed a new data fusion technique, *probFuse*. Using this algorithm, documents are ranked based on their probability of relevance. In experiments on small collections, *probFuse* shows initial promise, outperforming

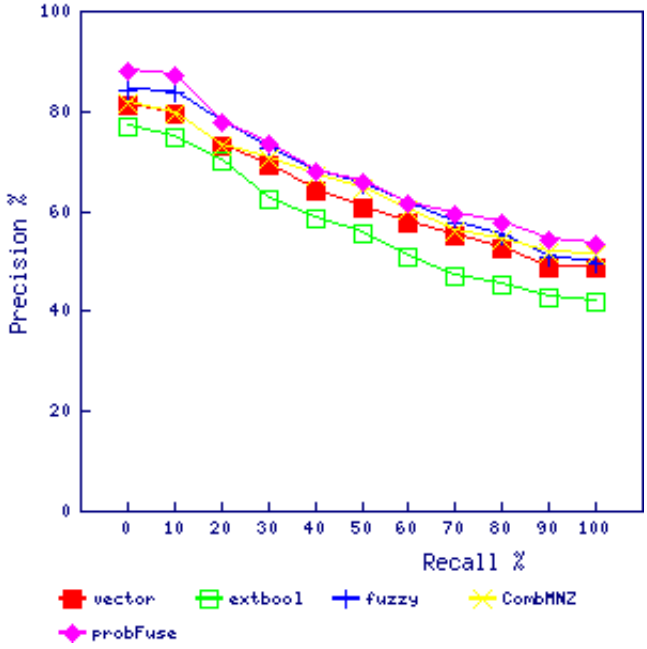

Fig. 3. Interpolated Precision graph for the Cranfield Collection

the best performance of any of the individual retrieval models that we used, namely the Vector Space Model, the Fuzzy Set Model and the Extended Boolean Model. It also was shown to produce superior results to the popular CombMNZ algorithm.

While *probFuse* shows promise on these small collections, it remains to be seen whether the increase in retrieval effectiveness achieved on small collections can be replicated on larger document collections, such as data from the Text REtrieval Conferences (TREC), which is widely used to evaluate fusion techniques.

References

1. Voorhees, E.M., Gupta, N.K., Johnson-Laird, B.: The collection fusion problem. In: Proceedings of the Third Text REtrieval Conference. (1994)
2. Voorhees, E.M., Gupta, N.K., Johnson-Laird, B.: Learning collection fusion strategies. In: SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1995) 172–179
3. Selberg, E., Etzioni, O.: The MetaCrawler architecture for resource aggregation on the Web. IEEE Expert (1997) 11–14

4. Howe, A.E., Dreilinger, D.: SAVVYSEARCH: A metasearch engine that learns which search engines to query. AI Magazine **18** (1997) 19–25
5. Shaw, J.A., Fox, E.A.: Combination of multiple searches. In: Proceedings of the 2nd Text REtrieval Conference. (1994) 243–252
6. Beitzel, S.M., Jensen, E.C., Chowdhury, A., Grossman, D., Frieder, O., Goharian, N.: Fusion of effective retrieval strategies in the same information retrieval system. J. Am. Soc. Inf. Sci. Technol. **55** (2004) 859–868
7. Montague, M., Aslam, J.A.: Condorcet fusion for improved retrieval. In: CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management, New York, NY, USA, ACM Press (2002) 538–548
8. Lee, J.H.: Analyses of multiple evidence combination. In: SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1997) 267–276
9. Vogt, C.C., Cottrell, G.W.: Fusion via a linear combination of scores. Inf. Retr. **1** (1999) 151–173
10. Bartell, B.T., Cottrell, G.W., Belew, R.K.: Automatic combination of multiple ranked retrieval systems. In: SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, Springer-Verlag New York, Inc. (1994) 173–181
11. Aslam, J.A., Montague, M.: Models for metasearch. In: SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2001) 276–284
12. Salton, G., Lesk, M.E.: Computer evaluation of indexing and text processing. J. ACM **15** (1968) 8–36
13. Salton, G., Fox, E.A., Wu, H.: Extended boolean information retrieval. Commun. ACM **26** (1983) 1022–1036
14. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)