Artif Intell Rev (2006) 25:179–191
DOI 10.1007/s10462-007-9021-x

Probability-based fusion of information retrieval result sets

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

Published online: 21 August 2007
© Springer Science+Business Media B.V. 2007

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. Combining the results produced by different algorithms has resulted in superior retrieval performance and this has become an important research area. This paper introduces a probability-based fusion technique *probFuse* that shows initial promise in addressing this question. It also compares *probFuse* with the common CombMNZ data fusion technique.

Keywords Data fusion · Information retrieval · *ProbFuse*

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. These IR techniques typically

D. Lillis (✉) · F. Toolan · A. Mur · L. Peng · R. Collier · J. Dunnion
School of Computer Science and Informatics, University College Dublin, Dublin, Ireland
e-mail: david.lillis@ucd.ie

F. Toolan
e-mail: fergus.toolan@ucd.ie

A. Mur
e-mail: mur.angel@ucd.ie

L. Peng
e-mail: peng.liu@ucd.ie

R. Collier
e-mail: rem.collier@ucd.ie

J. Dunnion
e-mail: john.dunnion@ucd.ie

assign a score to each document in a collection. This score is a judgment of that document's relevance to the given query. A list of documents, ranked according to this relevance score, is then returned. These ranked lists are known by numerous names in the relevant literature. For consistency, we will refer to them as *result sets*, as in (Beitzel et al. 2004).

No single approach to IR has been demonstrated to achieve superior performance to all others in all situations. Individual IR systems will retrieve different documents from the same document collection in response to the same query (Das-Gupta and Katzer 1983). This may be a result of differing methods of representing documents and user queries, different document and query preprocessing steps and different algorithms used to rank documents. This is illustrated by the entries for the TREC-1 conference, many of which achieved approximately the same performance level despite returning substantially different documents in their result sets (Harman 1993).

The combination of result sets produced by a number of different IR algorithms has been shown to improve retrieval performance (Bartell et al. 1994). This has become known as "data fusion" (Aslam and Montague 2000). Because some IR algorithms will return documents that are not returned by others, combining numerous result sets in the correct way will lead to an increase in recall by presenting a greater number of relevant documents to the user. Additionally, it has been shown that the presence of a document in the results returned by a number of IR systems reflects an increased probability that it is relevant to the given query (Saracevic and Kantor 1988). Thus, returning documents that are returned by multiple systems can lead to increased precision.

The term "data fusion" specifically refers to the use of a number of different IR systems to retrieve documents from the same document collection and the combination of their result sets using any information that is available in order to achieve superior results. This is in contrast to related tasks such as "collection fusion", where the document collections being searched are distinct (Voorhees et al. 1994) and situations where there is only partial overlap between the document collections (Wu and Crestani 2004), although research in these areas is also relevant to data fusion.

In order to perform data fusion, a number of solutions have been proposed. 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 weightings to create a bias to favour some sources over others.

In many cases, such research has been in the context of meta search engines (Voorhees et al. 1994), which involve the fusion of result sets produced by distinct, autonomous IR systems. Because these search engines are designed to act in a standalone fashion, rather than being specifically designed for use by a meta search engine, relevance scores are not necessarily made available to the meta search algorithm. Similarly, information about the contents of the database each engine has access to is typically unavailable.

Our research is aimed at combining results from multiple IR algorithms running on the same document collection within the same system. The HOTAIR (Highly Organised Teams of Agents for Information Retrieval) architecture is an extensible and scalable multi-agent architecture for the discovery, retrieval and indexing of documents from multiple information sources (Mur et al. 2005) (Peng et al. 2005). Our research on fusion is intended for use within this system.

This paper is organised as follows: an overview of some approaches that have been taken by others in solving the fusion problem in the past is provided in Sect. 2. Section 3 details the problem in question. In Sect. 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 (Lee 1997). We conclude in Sect. 6 and discuss possible directions for further study.

2 Prior work

An early, simple method of merging distinct result sets is to interleave the results in round-robin fashion (Voorhees et al. 1994), 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 the result sets are of equal quality. An empirical study (Voorhees et al. 1995) demonstrates a 40% degradation in effectiveness when compared to the performance of a single centralised collection.

Voorhees et al. suggested two variations to interleaving (Voorhees et al. 1995; Voorhees and Tong 1997): *Modeling Relevant Document Distributions* and *Query Clustering*. The key focus of both was to use training data to predict which input systems were most likely to return the best results. A greater proportion of higher-ranked systems' result sets were used in the fused result set. Once the documents to be fused had been identified, they were fused in a weighted fashion. For each position in the fused result set, a system was first chosen by rolling a C-sided die, biased by the number of documents remaining in each of the C result sets. Once a system was chosen, the first document remaining in its set was inserted into the fused set. This had the effect of preserving the rankings returned by each individual system and giving priority to those systems judged most likely to return relevant documents. The two algorithms only differ in their methods of ranking the input systems. Both rely on comparing the given query to training queries. In the *Modeling Relevant Document Distributions* algorithm, the query is compared (using cosine similarity) with each training query and the number of documents to be taken from each result set is based on the performance of each information source for the queries that are most similar to the given query. The *Query clustering* algorithm involves creating centroid clusters with the training queries, based on the number of retrieved documents that queries have in common. In this case, the number of documents to take from each result set is based on the performance of each system for the queries in the cluster that the given query is closest to.

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. 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 Fox and Shaw (Fox and Shaw 1994). 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 that are judged relevant by a higher number of individual techniques. CombMNZ has become the standard data fusion technique (Beitzel et al. 2004; Montague and Aslam 2002), as it has been shown to outperform the other techniques proposed by Fox and Shaw. In particular, Lee (Lee 1997) was able to achieve significant improvements by using CombMNZ.

The fusion technique used by the *MetaCrawler* meta search engine (Selberg and Etzioni 1997) is the same as CombSUM, with a slightly different normalisation scheme to that used by Fox and Shaw. SavvySearch (Howe and Dreilinger 1997) also uses CombSUM, although some adjustments are made for documents for which relevance scores are not available.

CombMNZ has also been used with alternative methods of normalising scores (Montague and Aslam 2001) and also using document ranks to produce normalised scores, where relevance scores are not available (Lee 1997).

Wu and Crestani (Wu and Crestani 2004) proposed "Shadow Document" methods of fusion. The focus of their work was partially overlapping databases. In that scenario, algorithms that use a document's presence in multiple result sets as evidence of relevance cannot be used. The score assigned to each document for fusion purposes is the sum of its normalised scores in each result set. If a document appears in one result set but not another, the score assigned to it for that result set is based on its normalised score in the result set it does appear in and also on a coefficient. This coefficient can either be determined empirically or as a function of the degree of overlap of the result sets.

The Linear Combination model involves a weight being calculated for each input system. That weight then is multiplied by each document score in the relevant result set, with the final fused score for a document being the sum of these weighted scores.

A Linear Combination model was used in (Callan et al. 1995), where the weight of each input system was a function of the score calculated by the CORI algorithm (an algorithm for ranking IR systems for particular queries to ascertain which are most likely to produce the best results). This was also used in (Powell et al. 2000) and a variation that also used normalised relevance scores was used in (Si and Callan 2002) and (Larkey et al. 2000).

A different weighting system was proposed in (Rasolofo et al. 2001). Named LMS (using result Length to calculate Merging Score), it relied on the number of documents returned by each input system. This was based on the hypothesis that systems returning more documents are more likely to be providing better results. Under LMS, the weight given to each input system is the number of documents returned by it, relative to the number returned by the other systems. Its principal advantage lies in its simplicity since no prior knowledge of the input systems is necessary.

In (Wu and Crestani 2002), the *WSUM* (Weighted SUM) technique calculates three possible weights to be assigned to input systems: 'good', 'fair' and 'poor'. The appropriate weight for each is calculated by testing how much agreement there is between systems. For each system, they take the top N documents in its result set and sum the number of occurrences of these documents in all other result sets. The categorisation of a system as 'good', 'fair' or 'poor' is based on how this score compares with the average for all systems. The score used for each document for fusion was a linear combination of the document's normalised score and the weights of the appropriate input systems. A variation of this for situations where scores were not available was also proposed. Here, a score was calculated based on the ranking of the document in the result sets and the linear combination was performed based on that.

Aslam and Montague have proposed two fusion techniques that are based on algorithms designed to identify successful candidates in democratic elections where there are more than two candidates. Borda-fuse (Aslam and Montague 2001) is based on an election model designed for a situation where there are many candidates, but few voters. They use the analogy that the voters are equivalent to the input systems being used and the candidates are represented by the documents retrieved.

With Borda-fuse, each voter ranks a set of c candidates in order of preference. The top ranked candidate is awarded c points, and the score for each candidate decreases by one as we progress down the list. The total score for any one candidate is the sum of the points awarded to it by all the voters.

The other voting model they proposed is *Condorcet-fuse* (Montague and Aslam 2002). Under the traditional Condorcet algorithm, the winner is the candidate that beats or ties with

every other candidate in a pairwise comparison. It must be adapted slightly for fusion, since the goal is not merely to identify the top-ranked document but the desire to include all documents in a ranked list. Condorcet-fuse firstly creates a list of all documents returned by any input system. It then uses the QuickSort algorithm with the comparison function being the relative ranking of documents in each system. A document will be ranked above another if it is ranked higher in more of the input result sets. Weighted versions of each of these algorithms were also proposed.

3 Problem description

The characteristics of fusion are outlined by Vogt and Cottrell (Vogt and Cottrell 1999). If the individual result sets contain 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 that attach a greater significance to documents common to multiple sources will perform well. Research involving the CombMNZ algorithm has shown that the Chorus Effect is very significant in data fusion tasks.

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 results. It is noted that the Chorus and Dark Horse effects are somewhat contradictory in nature, with the former encouraging fusion techniques to take as many techniques as possible into account when fusing and the latter suggesting that a single technique may provide the best performance.

The degree to which any of these effects is important is dependent on the type of input system producing the result sets. Fusion techniques that attempt to make use of the Chorus Effect can only do so when the databases available to the individual input systems contain the same documents. The reason for this is the treatment of documents that are returned in one result set but not another. If there is only partial overlap between the databases, such a situation can be explained by one of the following:

1. The document is contained in both databases but is not considered to be relevant to the query by one system.
2. The document is only contained in one database and so can only possibly be returned by one system.

Clearly, in this situation, the presence of a document in multiple result sets cannot be reliably used as evidence of relevance. The Chorus Effect will also have no influence on fusion where the databases are disjoint, as no document will appear in multiple result sets.

Our research focuses on different IR models running on identical document collections, and so it is desirable for our fusion techniques to leverage the Chorus Effect. This has become known as *data fusion* (Aslam and Montague 2000), as distinct from *collection fusion* (Voorhees et al. 1994), which relates to fusion of result sets from disjoint databases.

If we have a system in which we use multiple IR models, the relevant documents returned by each model are likely to be different (Das-Gupta and Katzer 1983; Harman 1993). 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 Sect. 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. This technique is motivated by work in the classifier ensemble domain in machine learning. In this area, the results of multiple classification techniques are combined to form one consistent classification scheme that should outperform all individual members. Similar to such ensemble schemes as weighted voting, the *probFuse* algorithm generates a representation of the past performance of individual techniques and uses this information to combine results from these techniques in future occurrences. For a discussion of ensemble techniques in the domain of machine learning, see (Dietterich 2000; Giacinto and Roli 2001).

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.

Running *probFuse* requires that relevance judgments are available. These are lists of documents compiled by human judges that indicate which documents in the collection are relevant to the given queries. Using these judgments, the performance of each individual IR model on a number of training queries can be analysed and the probability that a document returned by a particular model is relevant can be calculated.

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.

Figure 1 shows an example of segmenting a result set using three different values of x. This result set can be considered to have been produced by a single IR model that has ranked twelve documents from the document with the highest relevance score (*d12*) to the document with the lowest relevance score (*d27*). Taking document *d215* as an example, we can see that for a value of $x = 2$ (i.e. the result set is divided into two segments), it appears in the second segment of the result set. For $x = 3$, it is still in the second segment, moving to the third for $x = 4$.

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 judged to be relevant to query q, and K is the total number of documents in segment k.

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

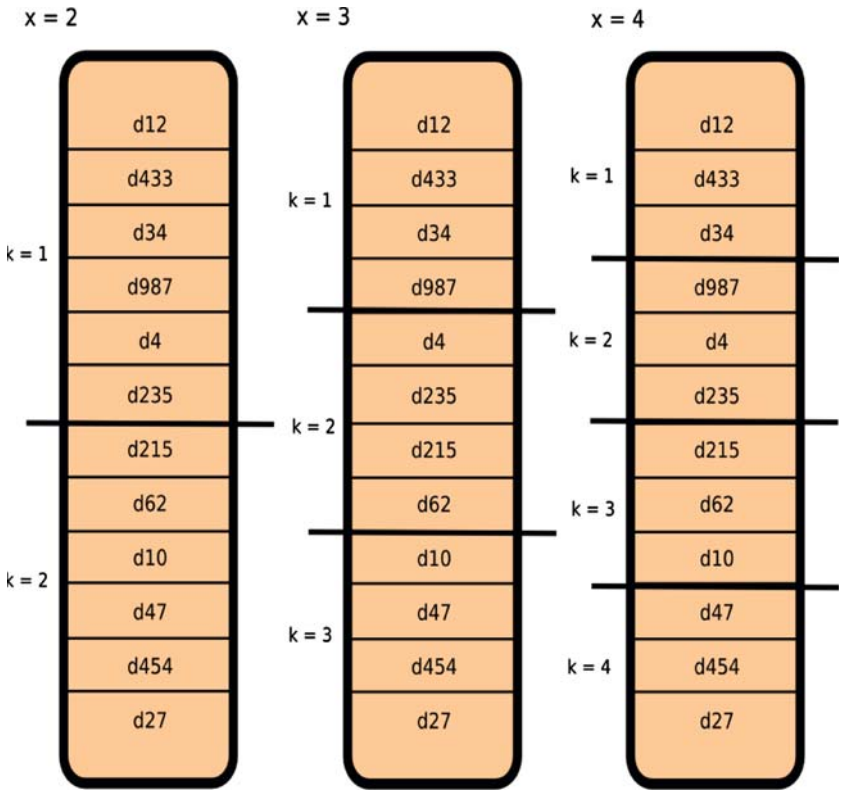

Fig. 1 Segmenting a result set for different values of x

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 that 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 produce 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 (Vogt and Cottrell 1999). 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. Secondly, by using the sum of the scores from each individual technique, rather than the maximum, we make use of the Chorus effect in a

Table 1 Characteristics of document collections used

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

similar way to CombMNZ. 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. One reason for selecting these collections is that complete relevance judgments are available for them, making it easier to calculate the probabilities for each segment in the training phase. It is likely that the generation of useful probabilities would be more difficult with larger collections where the relevance judgements are incomplete and we leave such experiments for future work.

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 arising from 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 (Salton and Lesk 1968), the Extended Boolean Model (Salton et al. 1983) and the Fuzzy Set Model (Baeza-Yates and Ribeiro-Neto 1999). We then ran *probFuse* on each, using various training set sizes and x values.

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 (Baeza-Yates and Ribeiro-Neto 1999) (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 Figs. 2 and 3 is the average $\overline{\Delta P_c}$ across all four collections.

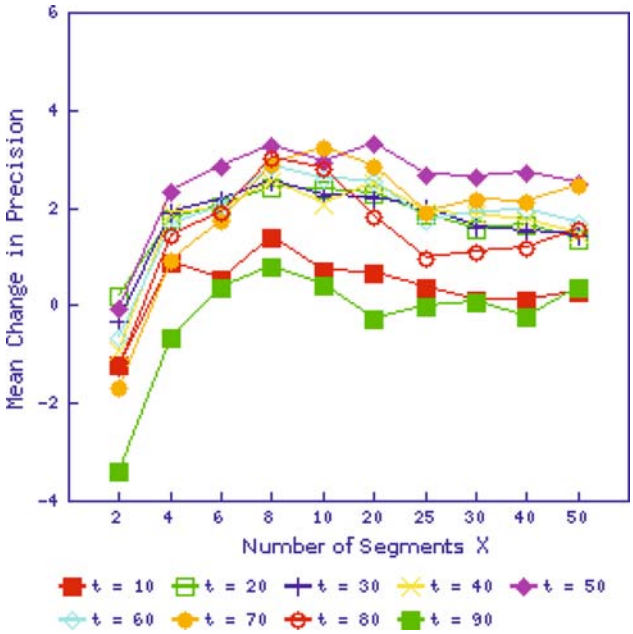

Fig. 2 Mean difference in precision for different training set sizes

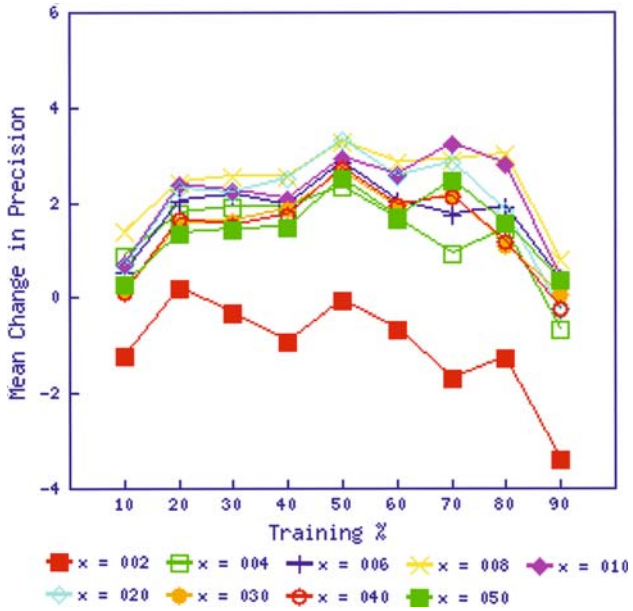

Fig. 3 Mean difference in precision for different values of x

Using this evaluation measure ensures that fused result sets are compared with the best precision at each recall level, rather than a single, overall measure. For example, we may have one technique that achieves high levels of precision at low recall but which drops significantly as recall increases. Another technique may initially achieve lower levels of precision but suffers less degradation as recall increases. This may lead to the second technique having greater precision than the first at higher levels of recall. In this situation, both techniques will be taken into account when calculating the mean change in precision, as we consider the technique that has the highest level of precision for each level of recall.

Figure 2 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% yields consistently superior performance to the other training set sizes and results in the best performance for almost all values of x.

In Fig. 3, 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. This is clearly too coarse a measure, as the results show. Increasing values for x produce superior results, to a point, with x values of 10 and 20 showing the highest mean precision increases.

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-performing combination of x and t values, we then performed a comparison of those results and the CombMNZ algorithm. CombMNZ was selected as the technique for comparison, as it has been shown to perform well on data fusion tasks by exploiting the Chorus Effect (Lee 1997).

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 to a common scale. Numerous methods of normalising relevance scores have been proposed. Here, we follow the method used by Lee for calculating normalised scores, which has been described by Montague and Aslam as "Standard Normalisation" (Montague and Aslam 2001). Lee's implementation of CombMNZ normalised scores using

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

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

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

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.

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 available queries for each collection are being used solely as training data by *probFuse*,

Table 2 Comparison of the mean difference in precision achieved by the *probFuse* and CombMNZ algorithms for each collection

	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

Entries marked with "*" are statistically significant for a significance level of 5%. Entries marked with "**" are statistically significant for a significance level of 1%, as calculated by the Wilcoxon test

we have ignored them 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. We can see 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 achieves significant improvements for the NPL collection alone.

Figure 4 illustrates the performance of *probFuse* and CombMNZ on the Cranfield collection. It shows the interpolated precision at the standard recall levels for each of the three individual techniques, as well as for each of the two fusion techniques.

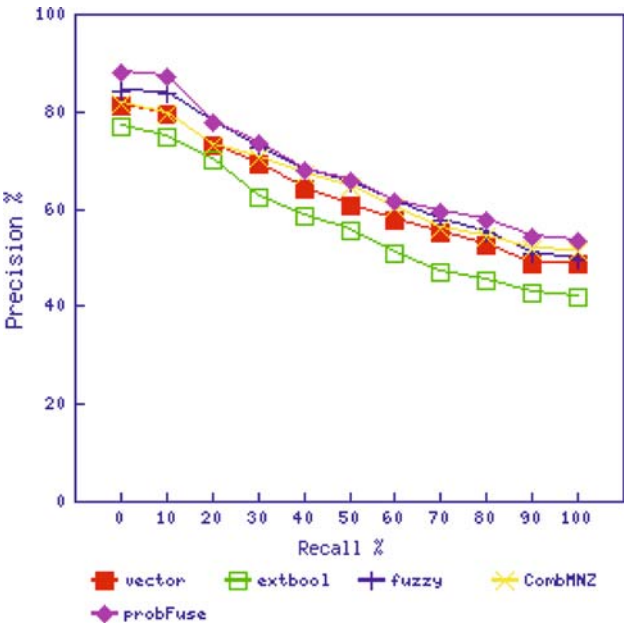

Fig. 4 Interpolated precision graph for the Cranfield collection

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. This probability is calculated based on the performance of the underlying IR models on a number of training queries. In experiments on small collections, *probFuse* shows initial promise, outperforming 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. The lack of availability of full relevance judgements, particularly for the larger TREC collections, may require adjustments to the methods that we are currently using to calculate segment probabilities. In that situation, most evaluation measures assume documents for which a judgment is unavailable to be non-relevant. However, for the purposes of calculating probabilities based on a training set, this may not be a reasonable assumption to make. For this reason, it may be necessary to calculate the probabilities based on the number of judged documents in each segment, rather than the total number of documents appearing in each segment.

The next stage in our research is to compare the performance of *probFuse* against that of other data fusion techniques using the larger TREC document collections. A successful outcome to this comparison would allow the investigation of the effects of training *probFuse* on one document collection and performing fusion on another. Ultimately, the aim of this research is to train *probFuse* on existing document collections for which relevance judgments are available (such as the TREC Web Track) and apply it to large-scale domains such as web search and corporate intranets

Acknowledgement We gratefully acknowledge the support of Enterprise Ireland through grant No. PC/2004/377

References

Aslam JA, Montague M (2000) Bayes optimal metasearch: a probabilistic model for combining the results of multiple retrieval systems. In: SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM Press. New York, NY, USA, pp 379–381

Aslam JA, Montague M (2001) Models for metasearch. In: SIGIR '01: proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM Press. New York, NY, USA, pp 276–284

Baeza-Yates RA, Ribeiro-Neto B (1999) Modern information retrieval. Addison-Wesley Longman Publishing Co, Inc, Boston, MA, USA

Bartell BT, Cottrell GW, Belew RK (1994) 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. Springer-Verlag, New York, New York Inc., NY, USA, pp 173–181

Beitzel SM., Jensen EC, Chowdhury A, Grossman D, Frieder O, Goharian N (2004) Fusion of effective retrieval strategies in the same information retrieval system. J Am Soc Inf Sci Technol 55(10): 859–868

Callan JP, Lu Z, Croft WB (1995) Searching distributed collections with inference networks. In: SIGIR '95: proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. ACM Press. New York, NY, USA, pp 21–28

Das-Gupta P, Katzer J (1983) A study of the overlap among document representations. In: SIGIR '83: Proceedings of the 6th annual international ACM SIGIR conference on Research and development in information retrieval. ACM Press. New York, NY, USA, pp 106–114

Dietterich TG (2000) Ensemble methods in machine learning. Lecture Notes Comput Sci 1857:1–15

Fox EA, Shaw JA (1994) Combination of multiple searches. In: Proceedings of the 2nd text Retrieval conference (TREC-2), national institute of standards and technology special publication 500-215. pp 243–252

Giacinto G, Roli F (2001) Dynamic classifier selection based on multiple classifier behaviour. Pattern Recogn 34(9):1879–1881

Harman D (1993) Overview of the first text retrieval conference (TREC-1). In: SIGIR '93: proceedings of the 16th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press. New York, NY, USA, pp 36–47

Howe AE, Dreilinger D (1997) SavvySearch: a metasearch engine that learns which search engines to query.. AI Mag 18(2):19–25

Larkey LS, Connell ME, Callan J (2000) Collection selection and results merging with topically organized U.S. patents and TREC data. In: CIKM '00: proceedings of the ninth international conference on Information and knowledge management. ACM Press. New York, NY, USA, pp 282–289

Lee JH (1997) Analyses of multiple evidence combination. SIGIR Forum 31(SI):267–276

Montague M, Aslam JA (2001) Relevance score normalization for metasearch. In: CIKM '01: proceedings of the tenth international conference on Information and knowledge management. ACM Press. New York, NY, USA, pp 427–433

Montague M, Aslam JA (2002) Condorcet fusion for improved retrieval. In: CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management. ACM Press. New York, NY, USA, pp 538–548

Mur A, Peng L, Collier R, Lillis D, Toolan F, Dunnion J (2005) A HOTAIR scalability model. In: Proceedings of the 16th irish conference on artificial intelligence and cognitive science (AICS 2005). University of Ulster. Portstewart, Northern Ireland, pp 359–368

Peng L, Collier R, Mur A, Lillis D, Toolan F, Dunnion J (2005) A self-configuring agent-based document indexing system. In: Proceedings of the 4th international central and eastern european conference on multi-agent systems (CEEMAS 2005). Springer-Verlag GmbH, Budapest, Hungary,

Powell AL, French JC, Callan J, Connell M, Viles CL (2000) The impact of database selection on distributed searching. In: SIGIR '00: proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval. ACM Press. New York, NY, USA, pp 232–239

Rasolofo Y, Abbaci F, Savoy J (2001) Approaches to collection selection and results merging for distributed information retrieval. In: CIKM '01: proceedings of the tenth international conference on Information and knowledge management. ACM Press. New York, NY, USA, pp 191–198

Salton G, Fox EA, Wu H (1983) Extended boolean information retrieval. Commun ACM 26(11):1022–1036

Salton G, Lesk ME (1968) Computer evaluation of indexing and text processing. J ACM 15(1):8–36

Saracevic T, Kantor P (1988) A study of information seeking and retrieving. III. Searchers, searches, and overlap. J Am Soc Inform Sci 39(3):197–216

Selberg E, Etzioni O (1997) The metacrawler architecture for resource aggregation on the web. IEEE Expert (January–February): 11–14

Si L, Callan J (2002) Using sampled data and regression to merge search engine results. In: SIGIR '02: proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press. New York, NY, USA, pp 19–26

Vogt CC, Cottrell GW (1999) Fusion via a linear combination of scores. Inform Retrieval 1(3):151–173

Voorhees EM, Gupta NK, Johnson-Laird B (1994) The collection fusion problem. In: Proceedings of the third text retrieval conference (TREC-3). pp 95–104

Voorhees EM, Gupta NK, Johnson-Laird B (1995) Learning collection fusion strategies. In: SIGIR '95: proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval. ACM Press. New York, NY, USA, pp 172–179

Voorhees EM, Tong RM (1997) Multiple search engines in database merging. In: Proceedings of the second ACM international conference on digital libraries. ACM Press, Philadelphia, Pa, New York, pp 93–102

Wu S, Crestani F (2002) Data fusion with estimated weights. In: CIKM '02: Proceedings of the eleventh international conference on information and knowledge management. ACM Press. New York, NY, USA, pp 648–651

Wu S, Crestani F (2004) Shadow document methods of results merging. In: SAC '04: proceedings of the 2004 ACM symposium on applied computing. ACM Press. New York, NY, USA, pp 1067–1072