Artif Intell Rev (2006) 26:23–34
DOI 10.1007/s10462-007-9037-2

Probabilistic data fusion on a large document collection

David Lillis · Fergus Toolan · Rem Collier · John Dunnion

Published online: 14 September 2007

Abstract Data fusion is the process of combining the output of a number of Information Retrieval (IR) algorithms into a single result set, to achieve greater retrieval performance. *ProbFuse* is a data fusion algorithm that uses the history of the underlying IR algorithms to estimate the probability that subsequent result sets include relevant documents in particular positions. It has been shown to out-perform CombMNZ, the standard data fusion algorithm against which to compare performance, in a number of previous experiments. This paper builds upon this previous work and applies *probFuse* to the much larger Web Track document collection from the 2004 Text REtreival Conference. The performance of *probFuse* is compared against that of CombMNZ using a number of evaluation measures and is shown to achieve substantial performance improvements.

Keywords Data fusion · Information retrieval · *ProbFuse*

1 Introduction

In the past, various solutions have been proposed to solve the Information Retrieval (IR) problem of identifying documents that have relevance to particular user queries. No such solution

D. Lillis (✉) · R. Collier · J. Dunnion
School of Computer Science and Informatics,
University College Dublin,
Dublin 4, Ireland
e-mail: david.lillis@ucd.ie

F. Toolan
Faculty of Computing Science, Griffith College Dublin
Dublin 8, Ireland
e-mail: fergus.toolan@gcd.ie

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

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

has been proven to achieve superior performance over all others for all user queries. Indeed, even when the retrieval performance of the individual systems is similar, it has been shown that the documents retrieved by the individual IR systems are typically different (Harman 1993).

As a result of this, much research has been undertaken into combining the outputs of a number of different IR systems into a single result set, in order to achieve greater retrieval performance than any individual system. This is known as "data fusion" when the underlying systems have access to the same document collection (Aslam and Montague 2000).

ProbFuse is a novel data fusion algorithm that uses the probability that particular documents are relevant to a given query in order to produce a fused result set. In the past, *probFuse* has been shown to outperform the common CombMNZ algorithm on small document collections (Lillis et al. 2006b) and also on data taken from Text REtrieval Conferences (TREC) (Lillis et al. 2006a). In order to run *probFuse*, it is necessary to determine values for two variables, t and x. The meaning of these variables is discussed in Sect. 4. In previous work, these values have been set empirically in the course of our experiments. This paper, however, uses the variables selected during experiments on the TREC-3 and TREC-5 document collections (Lillis et al. 2006a). The aim is to demonstrate that these values are not collection-specific and that it is possible to determine values that will achieve superior performance on many document collections without the necessity of calculating different values for each.

In addition, the document collection used (the Web Track from the TREC-2004 conference) is much larger than those used in previous work. This has the effect that the relevance judgments are far less complete (i.e. only a small fraction of the documents in the collection have been judged to be relevant or nonrelevant to each query, leaving most documents unjudged). Because the *probFuse* algorithm relies on training data to estimate the probability that particular documents are relevant to given user queries, we aim to investigate the effects of this lower level of completeness on fusion performance.

Section 2 is a general description of the data fusion problem. In Sect. 3 we outline work that has been carried out by other researchers in the past to perform data fusion. Section 4 describes the *probFuse* data fusion algorithm. In Sect. 5 we outline an experiment to demonstrate the effectiveness of *probFuse* on the Web Track collection from the 2004 TREC Conference, including a comparison with the standard CombMNZ algorithm (Fox and Shaw 1994). Finally, Sect. 6 closes with our conclusions and intended future work.

2 Problem description

When performing data fusion, there are three "effects" that may be leveraged in order to achieve greater retrieval performance (Vogt and Cottrell 1999). The "Chorus Effect" describes a situation where a document is regarded to be relevant by a number of the underlying systems whose results are being fused. Fusion techniques that rank documents higher based on this agreement will tend to achieve better performance in this situation. The Chorus Effect is the key difference between the treatment of the data fusion and collection fusion tasks. In data fusion, the presence of a document in numerous result sets can be used as evidence of relevance, as each of the underlying systems has access to the same document collection. Where the document collections are disjoint, documents will only be returned by, at most, a single input result set. In situations where the document collections only partially overlap, it is not possible to draw conclusions about the relevance of documents based on their presence in or absence from multiple result sets. This is because a document returned in only a single result set may be present in only one document collection, or may be present in other

document collections but not be considered relevant by the other IR algorithms. Experiments by Lee (1997) have shown that the Chorus Effect is a significant factor when performing data fusion.

The most relevant documents are likely to be returned near the beginning of each result set. Where this is the case, fusion algorithms that "skim" the top documents from each of its input result sets and combine these to form the fused output will perform well. This is known as the "Skimming Effect".

The third "effect" to be taken into account is the "Dark Horse" effect. This describes a situation where one underlying system produces results that are of an unusually high (or low) standard. There is an apparent contradiction between this and the Chorus Effect. Whereas the Chorus Effect argues in favour of taking as many input result sets into account as possible, the Dark Horse Effect favours the identification of a single result set that is of a higher quality to the others.

3 Background research

Previous approaches to data fusion tend to fall into two broad categories. Some approaches use the rank in which a document appears in each result set to produce the fused output. Others use the scores assigned to each document by the underlying IR systems, generally utilising a normalisation step to map these scores into a common range. Two common score-based techniques were proposed by Fox and Shaw (1994). CombSum ranks documents according to the sum of the normalised scores assigned to them by the underlying systems. CombMNZ multiplies the CombSum score by the number of result sets in which the document is returned. A study by Lee (1997) achieved positive results by applying CombMNZ to the TREC-3 dataset. Real-world meta search engines such as MetaCrawler (Selberg and Etzioni 1997) have used CombSum as their fusion algorithms. The Linear Combination model is another common approach to score-based fusion (Vogt and Cottrell 1999), in which the scores assigned to documents by the underlying systems are multiplied by weights associated with each system.

An early rank-based technique is interleaving (Voorhees et al. 1994), which takes a document from each result set in turn in a round-robin fashion. Weighted variations have also been proposed (Voorhees et al. 1995). Other rank-based approaches include a variation of CombMNZ proposed by Lee (1997) and the Borda-fuse (Aslam and Montague 2001) and Condorcet-fuse (Montague and Aslam 2001) voting algorithms developed by Aslam and Montague. Manmatha et al. were able to calculate the probability that a document was relevant based on its position within the result set using Bayes' Rule (Manmatha et al. 2001).

Other approaches that use data other than ranks or scores have also been proposed. For example, some use the contents of the documents returned (Craswell et al. 1999; Lawrence and Giles 1998). Others require the underlying systems to provide metadata about the documents being returned, rather than merely assigning a ranking score (Gravano et al. 1997).

4 Probability-based fusion

This section describes *probFuse*, a data fusion algorithm that uses the probability that documents are relevant to a query to produce the final fused result set. The inputs to *probFuse* are the result sets returned by a number of different input systems in response to particular queries. Each input system has access to the same document collection.

The *probFuse* algorithm divides the input result sets into segments. The number of segments into which to divide each result set, x, is determined empirically. For a result set containing 1000 documents, if $x = 100$, each segment will contain 10 documents. An example of the segmentation of documents can be found in (Lillis et al. 2006b).

The first stage of applying *probFuse* is a training phase to determine the probability that a document returned in a particular segment by a particular input system is relevant. 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. Because this formula takes all of the documents in each segment into account, this is known as *probFuseAll*.

In previous work, a variation of this formula, known as *probFuseJudged* achieved slightly better retrieval performance where relevance judgments were incomplete (Lillis et al. 2006a). When using *probFuseJudged*, $P(d_k|m)$ is given by

$$P(d_k|m) = \frac{\sum_{q=1}^{Q} \frac{|R_{k,q}|}{|R_{k,q}|+|N_{k,q}|}}{Q} \tag{2}$$

where $|N_{k,q}|$ is the number of documents in segment k that are judged to be nonrelevant to query q.

ProbFuseJudged only takes documents in each segment that have been judged either relevant or nonrelevant into account when estimating the probability of relevance. In contrast, *probFuseAll* takes all of the documents in a segment into account, assuming unjudged documents to be nonrelevant.

Once these probabilities have been estimated for each segment number for each input system, fusion can take place. The ranking score S_d for each document d is given by

$$S_d = \sum_{m=1}^{M} \frac{P(d_k|m)}{k} \tag{3}$$

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 by retrieval model m, and k is the segment that d appears in (1 for the first segment, 2 for the second, etc.). If a particular document d is not included in a result set at all, $P(d_k|m)$ is considered to be zero.

Using the sum of the probabilities to generate the final ranking score for each document makes use of the Chorus Effect. The division by k gives greater weight to documents appearing in early segments and so aims to utilise the Skimming Effect.

5 Experiments and evaluation

This section describes an experiment to evaluate the performance of *probFuse* on the Web Track collection from the TREC-2004 conference.

In previous research on *probFuse*, the values for x, the number of segments into which to divide each result set and t, the percentage of queries to use for training purposes, have been calculated empirically using the document collection that was being evaluated

(Lillis et al. 2005,2006a). The aim of these experiments was to demonstrate that values could be identified that would cause *probFuse* to achieve superior performance to that of CombMNZ. For this experiment, we have taken the values that were shown to perform best on the data from the TREC-3 and TREC-5 conferences. Each result set was divided into 25 segments and 50% of the available queries were used as training data. This means that the values chosen are independent of the document collection being used in this case.

The goal of this experiment is to demonstrate that values for the number of segments and the training set size that have been calculated by using one document collection can be used to achieve high performance on other document collections. We demonstrate this by using the popular CombMNZ fusion algorithm as a baseline. CombMNZ has been shown to achieve high performance on data fusion tasks (Lee 1997) and has become the standard data fusion technique against which to compare new algorithms (Aslam and Montague 2001).

An additional goal is to investigate the effects of using a document collection where relevance judgments are extremely incomplete.

Section 5.1 describes the setup of the experiment that was carried out on the TREC-2004 Web Track data. The results of the evaluation using MAP and bpref are presented in Sect. 5.2. These results are then analysed in detail in Sect. 5.3.

5.1 Experiment setup

The TREC-2004 Web Track data includes 74 topfiles. Each of these topfiles contains result sets returned by one IR system for each of 225 queries. Five runs were performed and the evaluation results below are the average of the scores over all five runs. For each run, six random topfiles were selected, ensuring that no topfile was used in multiple runs.

For each experimental run, the order of the queries was randomised and fusion was then performed using *probFuseAll*, *probFuseJudged* and CombMNZ. This was done five times for each run and the evaluation scores associated with each run is the average of these. Doing this ensured that performance was not influenced by the order of the queries.

Initial evaluation of the fused result sets was carried out using the Mean Average Precision (MAP) and bpref measures. MAP is the mean of the precision scores obtained after each relevant document has been retrieved. Relevant documents that are not included in the result set are given a precision of zero (Buckley and Voorhees 2004). MAP assumes that documents that have not been judged are nonrelevant. The bpref measure evaluates the relative position of relevant and nonrelevant documents, ignoring documents that are unjudged. It was proposed by Buckley and Voorhees to cater for situations where relevance judgments are incomplete (Buckley and Voorhees 2004).

5.2 Evaluation

Table 1 shows the results for the *probFuseAll* and *probFuseJudged* algorithms for each of the five experimental runs, when evaluated using the MAP and bpref measures. The values in parentheses for *probFuseAll* and *probFuseJudged* are the percentage difference from the corresponding values for CombMNZ. The information presented in this table is also illustrated graphically in Fig. 1.

The bpref scores do not show a consistent pattern when comparing *probFuse* and CombMNZ. Both *probFuseAll* and *probFuseJudged* achieve higher bpref scores on the "first" and "fifth" runs, whereas CombMNZ performs better on the others. The degree by which one technique outperforms the other varies also. Whereas for the "fourth" run, CombMNZ's bpref

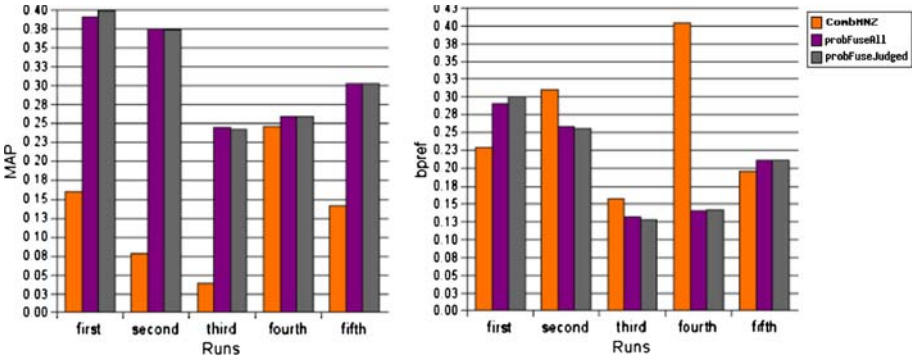

Fig. 1 TREC-2004 MAP and bpref scores for $t = 50\%$ and $x = 25$

score is over 60% higher than either *probFuse* variant, its score for the "first" run is lower by over 25%.

The MAP data contrasts sharply with this. Using this measure, the scores for *probFuse* are higher than those of CombMNZ in all cases, and drops below a 100% increase only for the "fourth" run. The average improvement in MAP score over CombMNZ is over 230% for both *probFuseAll* and *probFuseJudged*.

In previous experiments on smaller TREC datasets, *probFuseJudged* achieved slightly better performance than *probFuseAll*. One of the aims of this experiment is to investigate whether the scores for the two variations of *probFuse* diverge as the relevance judgments' level of completeness decreases. From Table 1, we can see that there is no significant difference between the scores for *probFuseAll* and *probFuseJudged* using either evaluation measure. In fact, the difference between the scores does not exceed 4% in any case. Thus we can conclude that the decision to base the probability scores on all documents or just judged documents does not have a significant effect on fusion performance, even in cases where the level of completeness of the relevance judgments is very low.

The contrast between the evaluation results for MAP and bpref is of interest. Whereas the MAP scores clearly indicate that the performance of *probFuse* is superior to that of CombMNZ, the bpref scores are inconclusive. In order to explain this, it is useful to examine the distribution of relevant, nonrelevant and unjudged documents in the result sets created by *probFuseAll*, *probFuseJudged* and CombMNZ. This is done in Sect. 5.3.

5.3 Analysis of experimental results

Figures. 2–4 illustrate the distribution of judged relevant, judged nonrelevant and unjudged documents respectively in the fused result sets output by *probFuseAll*, *probFuseJudged* and CombMNZ. These figures show the percentage of documents that are judged relevant, judged nonrelevant and unjudged at various positions in the result sets. Each data point represents this percentage for a group of 10 documents. For example, data points for position 0 on the x-axis represent documents from position 0 to position 9. All of the result sets produced by *probFuseAll*, *probFuseJudged* and CombMNZ during our experiments were used for this analysis.

Table 1 TREC-2004 performance of five individual runs for $t = 50\%$ and $x = 25$ using *probFuseAll* and *probFuseJudged*

	CombMNZ		*probFuseAll*		CombMNZ		*probFuseJudged*	
	MAP	bpref	MAP	bpref	MAP	bpref	MAP	bpref
First	0.16042	0.22868	0.39154 (+144.07%)	0.29016 (+26.88%)	0.16042	0.22868	0.39920 (+148.85%)	0.30082 (+31.55%)
Second	0.07808	0.31030	0.37536 (+380.74%)	0.25848 (−16.7%)	0.07808	0.31030	0.37340 (+378.23%)	0.25558 (−17.63%)
Third	0.03846	0.15788	0.24418(+534.89%)	0.13236 (−16.16%)	0.03846	0.15788	0.24132 (+527.46%)	0.12748 (−19.26%)
Fourth	0.24544	0.40436	0.25862 (+5.37%)	0.14048 (−65.26%)	0.24544	0.40436	0.25924 (+5.62%)	0.14204 (−64.87%)
Fifth	0.14130	0.19550	0.30278 (+114.28%)	0.21084 (+7.85%)	0.14130	0.19550	0.30284 (+114.32%)	0.21120 (+8.03%)
Average	0.13274	0.25934	0.31450 (+235.87%)	0.20646 (−12.68%)	0.13274	0.25934	0.31520 (+234.9%)	0.20742 (−12.44%)

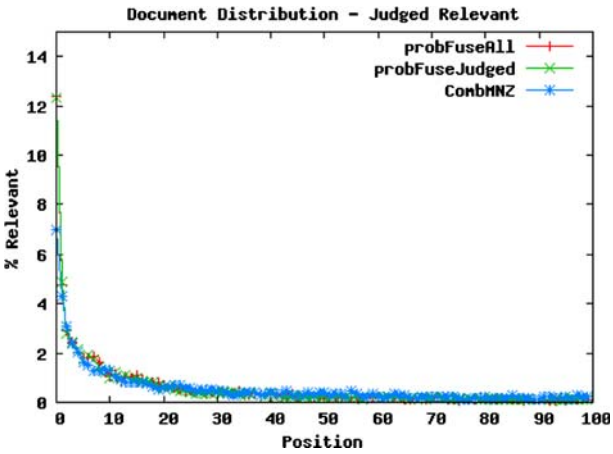

Fig. 2 Distribution of judged relevant documents

Both variations of *probFuse* return more judged relevant documents than CombMNZ in early positions. This will have a positive effect on their MAP scores, as MAP rewards highly-placed relevant documents more than those that are returned further down the result set or not returned at all. CombMNZ returns a greater number of unjudged documents towards the top of the result sets it produces. This has a detrimental effect on its MAP score, as these documents are assumed to be nonrelevant for this measure. However, bpref ignores these documents. This means that they have no effect on the bpref score despite the fact that relevant documents are then pushed further down the fused result set. The tendency to return nonrelevant documents in early positions is higher for *probFuse*. When evaluated using MAP, this is no different to returning unjudged documents. However, it does have a detrimental effect on bpref scores to rank nonrelevant documents above those that have been judged relevant.

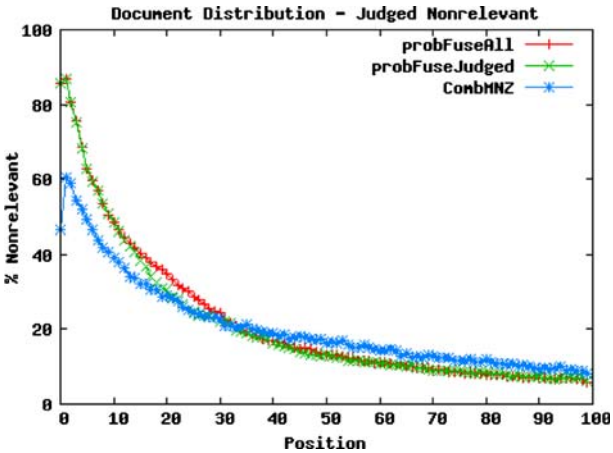

Fig. 3 Distribution of judged nonrelevant documents

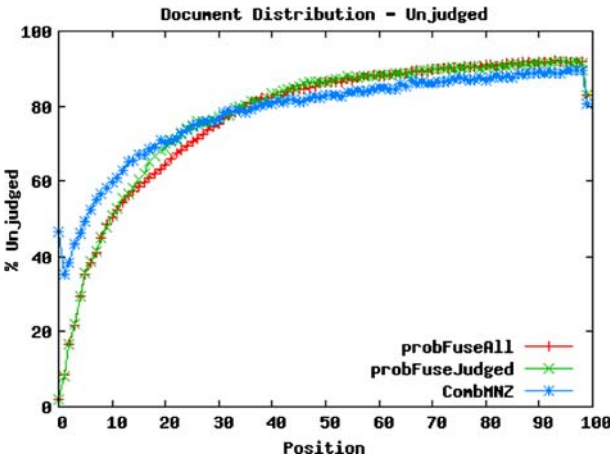

Fig. 4 Distribution of unjudged documents

Table 2 Average number of relevant documents returned

Average relevant documents	882.68
probFuseAll	690.84 (78.27%)
probFuseJudged	670.98 (76.01%)
CombMNZ	661.96 (74.99%)

Table 2 shows the average number of relevant documents that were returned by each fusion technique over all the runs. The "Average Relevant Documents" shows the average number of relevant documents that were available for retrieval. This is an average because the order of the queries were changed, meaning that a different set of queries was being used for fusion each time. From this figure, we can see that overall recall was higher for *probFuse* than for CombMNZ, meaning that more relevant documents were retrieved in total.

Because 50% of the 225 available queries were used for training, this means that 113 were used for fusion each time. Given that the average number of available relevant documents is 882.68, this means that an average of only 7.81 relevant documents was available to be retrieved for each query. This data is significant when interpreting the results returned by bpref. If R relevant documents are available for a query, bpref only considers the first R non-relevant documents. Consider a query for which there are eight relevant documents available. If a fusion technique returns eight nonrelevant documents in its first 20 results, returning a relevant document in 21st place is the same as returning the same document in 1,000th place or not at all. In contrast, MAP always considers a higher-placed relevant document to be superior, and also prefers a relevant document that has been returned in a result set to one that was not returned.

Additionally, since bpref ignores unjudged documents, this allows CombMNZ to return relevant documents further down its result sets without negatively impacting its bpref scores. However, this is reflected in the inferior MAP scores.

The tendency of *probFuseAll* and *probFuseJudged* to return a greater number of relevant documents in early positions motivates the introduction of a third evaluation measure, in order to observe the extent of this tendency. The P10 evaluation metric measures the precision after

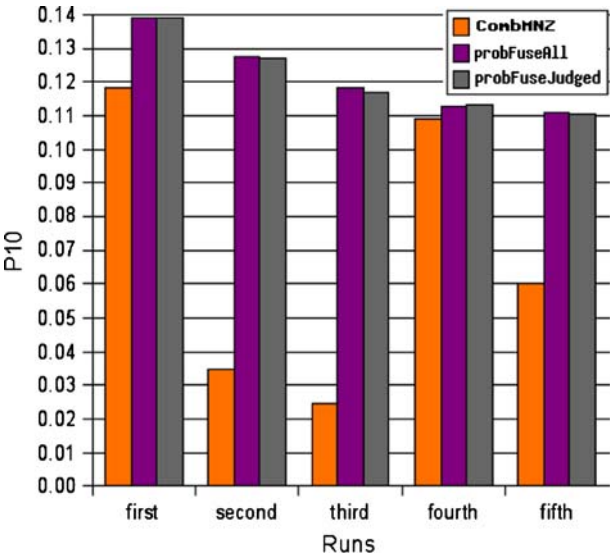

Fig. 5 TREC-2004 P10 scores for $t = 50\%$ and $x = 25$

Table 3 TREC-2004 performance of five individual runs for $t = 50\%$ and $x = 25$ using *probFuseAll* and *probFuseJudged*, evaluated with P10

	CombMNZ	*probFuseAll*	*probFuseJudged*
First	0.11858	0.13930 (+17.47%)	0.13910 (+17.31%)
Second	0.03486	0.12780 (+266.61%)	0.12708 (+264.54%)
Third	0.02480	0.11824 (+378.77%)	0.11700 (+371.78%)
Fourth	0.10902	0.11292 (+3.58%)	0.11310 (+3.74%)
Fifth	0.06038	0.11116 (+84.1%)	0.11046 (+82.94%)
Average	0.06953	0.12188 (+149.71%)	0.12135 (+148.06%)

ten documents have been retrieved, i.e. the fraction of the first 10 documents in a result set that are relevant to the given query.

The results of evaluating the outputs of *probFuse* and CombMNZ with the P10 measure are presented in Table 3 and Fig. 5. Again, there is little difference between the performance of *probFuseAll* and *probFuseJudged*. Both variations of *probFuse* achieve substantial performance increases over CombMNZ, however, with an average increase of almost 150%. As with the results under MAP, *probFuse* achieves higher scores for all five experimental runs, although the improvement for the "fourth" run is slight.

Having examined the output of the MAP, bpref and P10 evaluation measures, it is important to consider the behaviour of human users of IR systems. In general, a typical user expects to find relevant documents at the top of result sets. One study found that 85.2% of surveyed users examined only the top 10 results presented to them (Silverstein et al. 1998). This suggests that the P10 results can be considered to be particularly important, as they indicate the success of a data fusion algorithm to return relevant documents where users expect to find them.

This indicates that the tendency of CombMNZ to return relevant documents in lower positions, which is not penalised by bpref, is not desirable behaviour in a real-world system.

The vastly superior MAP and P10 scores achieved by probFuse over those of CombMNZ, together with the higher overall recall of *probFuse* support the conclusion that the *probFuse* algorithms display superior performance to CombMNZ.

6 Conclusions and future work

This paper presents the results of an experiment to apply the *probFuse* data fusion algorithm to the large Web Track collection from the TREC-2004 conference. Evaluating the performance of *probFuseAll* and *probFuseJudged* using the MAP evaluation measure showed the *probFuse* algorithms achieving improvements of over 230% over CombMNZ. The P10 evaluation measure showed the performance of *probFuse* to be almost 150% higher than CombMNZ. Using each of these measures, the performance of *probFuse* was higher than that of CombMNZ on all five experimental runs. A third evaluation metric, bpref, failed to show a consistent pattern. Additionally, *probFuse* achieved higher recall overall and tended to return more relevant documents at the top of the fused result sets produced.

The completeness of the relevance judgments for the Web Track collection is lower than that of others that have been used in previous experiments, meaning that the relevance of a greater number of documents is unknown. Despite this, no significant difference was observed between using all documents in the calculation of the probability of relevance and using just those documents that have been judged to be either relevant or nonrelevant, regardless of the evaluation measure used.

All the experiments to evaluate the effectiveness of *probFuse* that have been carried out to date have used training data from the same document collection as is used for fusion. In the future, we intend to investigate the results of using training data from one document collection and using this to perform fusion on another. In order to carryout this work, it will be necessary to use result sets that have been returned by a number of different IR systems in response to queries on each of two document collections. For the probability estimates to be meaningful, the systems used for the document collection used for fusion and evaluation must be the same as those used for the training phase. The data currently available from TREC does not provide this, as systems change from conference to conference.

Additionally, the result sets provided in the TREC data are of a fixed length of 1,000 documents. In order for *probFuse* to be applicable in real-world situations, it must be capable of performing well on variable length result sets, as some queries will result in more documents being returned by the underlying IR systems than others.

We also intend to investigate methods of inferring the probability of relevance without the necessity for relevance judgments to be available, thereby eliminating the need for a training phase. Previous work by Manmatha et al. (2001) in estimating the probability that a document is relevant to a particular query may be useful in this regard. Another possible approach would be to adjust the probabilities associated with each underlying input system in response to users' behaviour at query time.

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. New York, NY, USA. ACM Press, 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. New York, NY, USA. ACM Press, pp 276–284

Buckley C, Voorhees EM (2004) Retrieval evaluation with incomplete information. In: SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA. ACM Press, pp 25–32

Craswell N, Hawking D, Thistlewaite PB (1999) Merging results from isolated search engines. In: Australasian database conference. Auckland, New Zealand, pp 189–200

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

Gravano L, Chang K, Garcia-Molina H, Paepcke A (1997) STARTS: Stanford Protocol Proposal for Internet Retrieval and Search. Technical report, Stanford, CA, USA

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. New York, NY, USA. ACM press, pp 36–47

Lawrence S, Giles CL (1998) Inquirus, The NECI meta search engine. In: Seventh international world wide web conference. Brisbane, Australia. Elsevier Science, pp 95–105.

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

Lillis D, Toolan F, Mur A, Peng L, Collier R, Dunnion J (2005) Probability-based fusion of information retrieval result sets. In: Proceedings of the 16th Irish conference on artificial intelligence and cognitive science (AICS 2005). Portstewart, Northern Ireland. University of Ulster, pp 147–156

Lillis D, Toolan F, Collier R, Dunnion J (2006a) ProbFuse: a probabilistic approach to data fusion. In: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval. New York, USA. ACM Press, pp 139–146

Lillis D, Toolan F, Mur A, Peng L, Collier R, Dunnion J (2006b) Probability-based fusion of information retrieval result sets. Artif Intell Rev 25(1–2), doi:10.1007/s10462-007-9021-x

Manmatha R, Rath T, Feng F (2001) Modeling score distributions for combining the outputs of search engines. 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, pp 267–275

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

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

Silverstein C, Henzinger M, Marais H, Moricz M (1998) Analysis of a very large altavista query log. Technical Report 1998-014, Digital SRC http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/abstracts/src-tn-1998-014.html

Vogt CC, Cottrell GW (1999) Fusion via a linear combination of scores. Inform Retriev 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. New York, NY, USA. ACM Press, pp 172–179