Estimating Probabilities for Effective Data Fusion

David Lillis
School of Computer Science
and Informatics
University College Dublin
david.lillis@ucd.ie

Lusheng Zhang
School of Computer Science
and Informatics
University College Dublin
lu-sheng.zhang
@ucdconnect.ie

Fergus Toolan
School of Computer Science
and Informatics
University College Dublin
fergus.toolan@ucd.ie

Rem W. Collier
School of Computer Science
and Informatics
University College Dublin
rem.collier@ucd.ie

David Leonard
School of Computer Science
and Informatics
University College Dublin
david.leonard@ucd.ie

John Dunnion
School of Computer Science
and Informatics
University College Dublin
john.dunnion@ucd.ie

ABSTRACT

Data Fusion is the combination of a number of independent search results, relating to the same document collection, into a single result to be presented to the user. A number of probabilistic data fusion models have been shown to be effective in empirical studies. These typically attempt to estimate the probability that particular documents will be relevant, based on training data. However, little attempt has been made to gauge how the accuracy of these estimations affect fusion performance. The focus of this paper is twofold: firstly, that accurate estimation of the probability of relevance results in effective data fusion; and secondly, that an effective approximation of this probability can be made based on less training data that has previously been employed. This is based on the observation that the distribution of relevant documents follows a similar pattern in most high-quality result sets. Curve fitting suggests that this can be modelled by a simple function that is less complex than other models that have been proposed. The use of existing IR evaluation metrics is proposed as a substitution for probability calculations. Mean Average Precision is used to demonstrate the effectiveness of this approach, with evaluation results demonstrating competitive performance when compared with related algorithms with more onerous requirements for training data.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
Copyright 2010 ACM 978-1-60558-896-4/10/07 ...$10.00.

Keywords

information retrieval, probabilistic data fusion, results merging

1. INTRODUCTION

In the context of Information Retrieval (IR), many researchers have attempted to use data fusion to improve the quality of their results. This involves submitting a query to a number of distinct IR systems (known as "input systems", as they provide the inputs to the fusion process) that have access to the same document collection, and subsequently merging their outputs into a single result set to be presented to the user. This is related to, but distinct from, the concept of meta-search (or collection fusion), where the results being merged are from IR systems operating with disjoint (or partially overlapping) document collections [18].

Many techniques to tackle the data fusion task are available that use only the result sets that are actually being fused. These approaches vary from purely rank-based algorithms such as interleaving [18] to score-based techniques such as linear combinations [3, 14, 17] and the popular CombSum and CombMNZ algorithms [5, 6, 15]. Algorithms based on voting have also been popular [1, 13].

More recently, attempts have been made to take into account the past performance of input systems when performing fusion [1, 8, 10, 16]. These techniques make use of probabilities to calculate a score on which the final, fused result set will be ranked. Many, however, require detailed training data to be available, from which the probabilities are calculated. Typically, a number of training queries are run, with each of the input systems required to provide results for each. These results are then compared with relevance judgements so as to identify the positions in each result set where relevant documents have been returned. From this data, a model can be built up that predicts the probability of particular documents being relevant, based on the system that returned them and the positions in the respective result sets they occupy.

This position-level granularity of training data is an onerous requirement to have on a fusion process. Probabilistic data fusion with a minimal requirement for training would be preferable. The aim of this paper is to attempt to perform fusion based on probability of relevance, but without such a reliance on detailed training data. In order to do this, we must firstly demonstrate that an accurate probability model is indeed beneficial to fusion performance. Following this we attempt to show that this probability of relevance can be approximated by a function of a documents position within a result set. We also outline one candidate function to achieve effective fusion.

This paper is organised as follows: Firstly, Section 2 outlines a number of considerations that must be taken into account when developing a data fusion solution. Section 3 motivates the work by considering some pre-existing probabilistic fusion models and examines how an accurately-constructed probability model can result in effective data fusion. Following from this, Section 4 shows how such a probability model can be estimated by reference to a single-value measure of the quality of the inputs to the fusion process. Having chosen the Mean Average Precision evaluation metric as this single-value measure, a set of experiments is outlined in Section 5 that demonstrates the effectiveness of our approach when compared with others. Finally, we outline our conclusions and present some ideas for future work in Section 6.

2. CHARACTERISTICS OF DATA FUSION

When performing effective data fusion, there are a number of "effects" that may be taken into account. These were initially outlined by Vogt and Cottrell in [17].

- The *Skimming Effect* is based on the observation that relevant documents are more likely to appear at the top of result sets (where an IR system would place those documents it estimates to be most relevant). Thus favouring early-ranked documents when compiling the final result set can result in improved fusion performance.

- The *Chorus Effect* argues that if multiple input systems agree on the relevance of a document (by including it in each of their result sets) then this is increased evidence of relevance. This is also consistent with Lee's observation that IR systems tend to return the same relevant documents but different nonrelevant ones [7]. Fusion algorithms that attach greater importance to documents that are returned by multiple input systems attempt to exploit this effect.

- The *Dark Horse Effect* refers to a situation where an input system returns an unusually high- or low-quality result set. In this situation, if a fusion technique was able to identify a "dark horse", it may opt to return only the result set of that input system, rather than performing any fusion. This effect is very difficult to detect and we are not aware of any techniques that attempt to make use of it.

3. PROBABILITY AS A STRATEGY FOR DATA FUSION

A number of data fusion algorithms have been proposed that use the probability of relevance as a method of assigning scores to documents. Aslam and Montague make use of a Bayesian model that uses both the probability of relevance and the probability of non-relevance to rank documents [1]. The probabilities are calculated by examining the precision at a number of document levels. Result sets are divided into ranges between these document levels, with appropriate probability values being associated with each range. Manmatha et al. infer probabilities from the ranking scores given to documents by the various input systems [11].

Another group of probability-based fusion algorithms use training data to calculate a set of probabilities for each of the systems providing result sets to be fused. In this context, training data consists of result sets produced by the same input systems in response to queries for which relevance judgements are available. Having analysed where relevant documents tend to be returned by each inputs system, a probability model is built. For each of the input systems, this model maps a probability score on to each position in which a document may potentially be returned. For instance, System A may have a probability of 0.4 associated with position 1. This would imply that for any given document returned by System A at the top of its result set, there is an estimated probability of 0.4 that the document is relevant. Algorithms utilising this type of probability model include Lillis et al.'s ProbFuse [8] and SlideFuse [10] and the SegFuse algorithm developed by Shokouhi [16].

This approach to data fusion relies on two fundamental assumptions. Firstly, it is assumed that a system's performance in response to training queries is indicative of how it will perform when faced with different queries. It is also assumed that the construction of an accurate probability model will result in effective fusion. Although the empirical experiments presented in [9, 10, 16] demonstrate effective retrieval performance when compared against the baseline CombMNZ, the accuracy of the probability model is not tested.

The aim of this paper is to examine this second assumption in more detail. Establishing that an accurate model of the probabilities required results in effective fusion further motivates the examination of further methods of constructing such models.

In order to test this, we evaluated the effectiveness of using a perfect probability model for fusion. This perfect probability model was constructed by using the same queries (and consequently the same result sets) for training as for fusion. The consequence of this is that the probability model perfectly reflects the positions of the relevant documents in the result sets being used for fusion. Clearly, such an approach is not feasible from a practical point of view, as the relevant documents are not known at query time. However, the aim of this experiment is to demonstrate how effective fusion would be if an accurate approximation of the real probability distribution could be constructed.

The inputs for this experiment were taken from the TREC 2004 Web Track [4]. Five fusion runs were performed, using six input systems each time. The systems were chosen by their overall MAP score, with the six best systems being part of *run*1, the seventh to twelfth best systems in *run*2 etc. These inputs consisted of result sets relating to 225 distinct topics (queries).

The specific inputs used for each run are the same for all experiments presented in this paper, and are as follows:

- **run1:** MSRC04B2S, MSRC04C12, MSRC04B1S, MSRAx4, MSRAx2,MSRAmixed1

- **run2:** MSRAmixed3, MSRC04B1S2, MSRAx5, UAmsT04MSind, UAmsT04MWScb, UAmsT04MSinu

- **run3:** UAmsT04MWinu, uogWebSelAn, uogWebSelAnL, MSRC04B3S, THUIRmix045, THUIRmix041
- **run4:** uogWebCA, ICT04MNZ3, THUIRmix043, ICT04CIIS1AT, ICT04RULE, THUIRmix042
- **run5:** ICT04basic, ICT04CIILC, MeijiHILw1, uogWebSelL, UAmsT04LnuNG, MeijiHILw3

By way of comparison, the result sets were fused using the SlideFuse and CombMNZ fusion algorithms, which are described in detail in [10] and [5] respectively. SlideFuse is a probabilistic data fusion algorithm that estimates the probability of relevance at each position using training queries. It is chosen as a representative from the family of probabilistic algorithms to which it belongs (also including ProbFuse [9] and SegFuse [16]). In order to compensate for incomplete relevance judgements, where judgements of relevance or nonrelevance are not available for every document in the collection, SlideFuse smooths these probabilities using a sliding window approach. This means that the probabilities associated with each position also depends on the occurrence of relevant documents in neighbouring positions. In contrast, CombMNZ is a much simpler algorithm and has been chosen because it is frequently used as a baseline in fusion experiments. This does not use any training data, but rather uses the scores given to each document by the input systems to rank the fused result set. The details of how these were implemented are given in the following subsections.

3.1 PosFuse

The approach based on the perfect probability model is described here as “PosFuse” (as it is based on the probability at the position in which a document appears). Like SlideFuse, it is calculated in two stages: a training phase and a fusion phase.

In the training phase, $P(d_p|s)$ is calculated. This is the probability that a document d returned in position p of a result set is relevant, given that is has been returned by input system s. It is calculated by

$$P(d_p|s) = \frac{\sum_{q \in Q_p} R_{d_p,q}}{Q_p} \tag{1}$$

where Q_p is the set of all training queries for which at least p documents were returned by the input system and $R_{d_p,q}$ is the relevance of the document d_p to query q (1 if the document is relevant, 0 if not). This is calculated for each input system to be used in the fusion phase.

Following this, the fusion stage requires that a ranking score be assigned to each document (R_d). This is given by

$$R_d = \sum_{s \in S} P(d_p|s) \tag{2}$$

where S is the set of all input systems used and p is the position in which document d was returned by input system s. Although the use of probabilities would suggest that multiplication would be an obvious operator to use, the nature of data fusion makes addition more useful in this scenario. Adding the probability scores together results in a document's ranking score receiving a boost for every result set in which it appears (thus leveraging the Chorus Effect). R_d is intended as a score on which to rank documents, rather than an accurate estimation of the probability of a document's relevance.

3.2 SlideFuse

SlideFuse is a probabilistic fusion algorithm that is also based on the probability of relevance in various positions in result sets [10]. For SlideFuse, this probability calculation is the same as described above in Equation 1.

However, SlideFuse does not use this probability alone in order to calculate scores. It also employs a smoothing of these probabilities based on the notion of a sliding window. The argument in favour of this smoothing is that in certain situations, some positions may be ultimately given a probability of zero. This occurs whenever no relevant documents are returned by an input system at that exact position during the training phase. There are two principal reasons why this may happen:

1. **Few Training Queries:** If the number of queries being used for training is very small, this reduces the overall number of relevant documents being returned by each input system. Because of this, it consequently increases the chance that a particular position may not contain a relevant document for any of the training queries.
2. **Incomplete Relevance Judgements:** When relevance judgements are “incomplete”, not all documents have been judged for relevance to all the queries. This means that there three types of document: relevant, nonrelevant and unjudged. The lack of judged relevant documents appearing at any position may merely be as a result of documents being unjudged.

Whatever the reason, a probability of zero is undesirable. Firstly, it runs contrary to the Chorus Effect to neglect to take into account that a document was actually returned by an input system, regardless of its position. Secondly, it is counter-intuitive to give any document that was returned in a result set the same treatment as one that was not returned at all.

The sliding window is designed to reduce the likelihood of zero probabilities by also taking into account neighbouring positions. The start and end points (a and b respectively) of the sliding window surrounding each result set position p are given by

$$a = \begin{cases} p - w & p - w >= 0 \\ 0 & p - w < 0 \end{cases} \tag{3}$$

$$b = \begin{cases} p + w & p + w < N \\ N - 1 & p + w >= N \end{cases} \tag{4}$$

where w is a parameter that indicates how many positions on either side of p should be included in the window and N is the total number of documents in the result set. In effect, the above definitions of a and b ensure that the window cannot begin before the first document in the result set and also cannot extend beyond the last document.

Once the boundaries of the window have been set, a probability must be associated with each. $P(d_{p,w}|s)$, the probability of relevance of document d in position p using a window size of w documents either side of p, given that it has been returned by input system s is given by

$$P(d_{p,w}|s) = \frac{\sum_{i=a}^{b} P(d_i|s)}{b - a + 1} \tag{5}$$

Finally, a ranking score is given to each document using a formula very similar to Equation 2, except that the probability associated with the window is used instead of the probability at a particular rank.

$$R_d = \sum_{s \in S} P(d_{p,w}|s) \tag{6}$$

3.3 CombMNZ

Although it is not a probabilistic model, we also include the CombMNZ fusion algorithm, as it has become a standard baseline against which other fusion algorithms are compared [2, 13, 19]. Originally proposed by Fox and Shaw in [5], CombMNZ is a score-based algorithm that does not rely on training. It has gained popularity as a baseline measure principally because it is easily implemented and its retrieval performance tends to be very strong, despite its simplicity. Our implementation of CombMNZ follows that of Lee [7], who carried out a number or experiments using a variety of techniques proposed by Fox and Shaw.

CombMNZ is run in two phases. Unlike PosFuse and SlideFuse, both of these are done at fusion time, with no training required. Because CombMNZ is based on the scores attributed to each document by each of the input systems, the first requirement is that these be normalised. This is intended to scale all of the scores into the same range, so as to avoid a situation where one input system attaches greater weight to documents merely because it calculates scores from 0 to 100 rather than from 0 to 1.

The normalisation formula used by Lee is known as “standard normalisation” [12] and is given by

$$normalised\_sim = \frac{unnormalised\_sim - min\_sim}{max\_sim - min\_sim} \tag{7}$$

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

$$CombMNZ_d = \sum_{s=1}^{S} N_{s,d} \times |N_d > 0| \tag{8}$$

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.

3.4 Initial Results

Table 1 shows the MAP score for a number of data fusion algorithms. For comparison purposes, the column labelled “MaxMAP” shows the highest overall MAP score achieved by any individual input. It can be argued that this is the baseline that all fusion algorithms should aim to beat. If a fusion algorithm cannot achieve this level of performance, then a superior approach would simply be to identify which of the input systems performs best, discarding the others. In this table, the highest MAP score amongst the fusion algorithms is shown in bold.

	MaxMAP	PosFuse	SlideFuse	CombMNZ
run1	0.5389	**0.5751**	0.5697	0.3317
run2	0.5120	**0.5679**	0.5651	0.5249
run3	0.4589	**0.5375**	0.5223	0.1862
run4	0.4325	**0.4791**	0.4628	0.1740
run5	0.3976	**0.4907**	0.4640	0.4203

Table 1: MAP Scores when training on the actual result sets being fused. The highest MAP score for a fusion technique on each run is in bold.

From Table 1 it can be seen that the highest MAP scores are achieved on all runs by PosFuse. Additionally, these MAP scores are greater than the best performing individual input for each run. This is an interesting result in that it adds motivation to the pursuit of a probability distribution for the purposes of fusion.

Another interesting observation is that the PosFuse technique was able to achieve marginally greater MAP scores than SlideFuse. SlideFuse is based on probabilities that are initially calculated in the same way as for PosFuse, with the addition of the smoothing that is performed by the sliding window. It is, however, important to note that the principal motivation behind the use of sliding windows is to cater for situations where a small quantity of training queries combined with incomplete relevance judgements may cause some positions to be attributed a probability of zero. As the results shown in Table 1 are for fusion runs consisting of 225 training queries, this kind of situation does not arise to the same extent and so the motivation for this is lost. The performance of SlideFuse is still greater than the maximum individual MAP score, however.

4. MODELLING PROBABILITY

Having demonstrated the effectiveness of using accurate probability figures, we now investigate how this may be modelled, preferably without the necessity for large quantities of training data.

4.1 Curve Fitting

To do this, the probability distributions calculated for the 225-query run outlined in Section 3 were analysed. For each distribution (each one related to one input system), a curve was fitted using the gnuplot graphing utility [1]. In each case, the probability of relevance was plotted on the y-axis, with the result set position (starting at 1 for the top position in the result set) on the x-axis. In each case, gnuplot fit a curve of the form $y = \frac{a}{x}$, meaning that the probability of relevance would become a function of the result set position.

Figure 1 illustrates the process of fitting a curve for the MSRC04B32S input system. The first graph in that figure shows the results of fitting a curve only to the first 100 positions in each result set, whereas the second uses the entirety of each result set (TREC results are truncated to at most

[1] `http://gnuplot.info`

1000 documents in each result set). The fitted a-value is shown at the top-right of each graph. Despite the large difference in the lengths of the result sets being used, there is less than a 1% difference between the a-values generated.

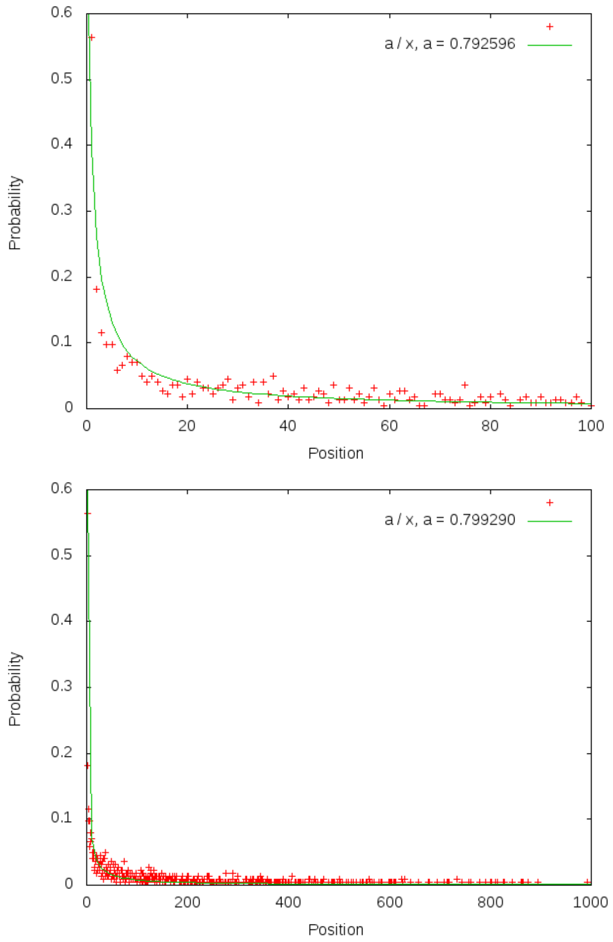

Figure 1: Curves fit to probability distribution for the MSRC04B2S input, showing 100 positions and 1000 positions respectively

Using the latter a-value (relating to 1000-document result sets), this leads to a modification of Equation 1 for calculating the probability of relevance. For the specific result sets used, the probability that a document d returned in position p in a result set created by the MSRC04B25 input system is represented by $P(d_p|MSRC04B25)$. Its value is given by

$$P(d_p|MSRC04B25) = \frac{0.79929}{p} \tag{9}$$

This is shown for illustrative purposes: similar fitting was done for all of the other input systems available, with a variety of a-values being generated. Although interesting that such a function can be generated for a range of input systems, to do so requires even more training effort than what was needed for the PosFuse algorithm used in Section 3. In addition to the training data necessary to calculate the probability of relevance at each position, the curve fitting would also have to be performed.

The shape of the fitted curves is interesting in that it supports the reasoning behind the description of the Skimming Effect. The graph shown in Figure 1 shows that documents ranked in early positions in result sets are much more likely to be relevant than those further down the result set. It also supports the idea that probability scores (or approximations thereof) can be effectively used in the calculation of fusion scores.

4.2 Evalation of curve fitted approach

To gauge how effective this is in terms of fusion performance, a comparison is made with the results obtained for the experiment outlined in Section 3. Table 3 reproduces the figures shown in Table 1, with the addition of an extra column (marked "FitFuse"), which is based on the fitted curves.

For FitFuse, rather than using the probabilities of relevance calculated on a per-position basis, we use the formula described in Equation 9. The fitted a-values used for each input system is given in Table 2.

Input	a-value
run1	
MSRAmixed1	0.806350
MSRAx4	0.803393
MSRC04B2S	0.799290
MSRAx2	0.798310
MSRC04B1S	0.786756
MSRC04C12	0.802797
run2	
MSRAmixed3	0.774764
MSRC04B1S2	0.701004
UAmsT04MSinu	0.469070
MSRAx5	0.787904
UAmsT04MSind	0.454965
UAmsTo4MWScb	0.466055
run3	
MSRC04B3S	0.649945
THUIRmix041	0.641529
THUIRmix045	0.661245
UAmsT04MWinu	0.458516
uogWebSelAn	0.469182
uogWebSelAnL	0.451617
run4	
ICT04CIIS1AT	0.685360
ICT04MNZ3	0.690526
ICT04RULE	0.653053
THUIRmix042	0.649099
THUIRmix043	0.638313
uogWebCA	0.401074
run5	
ICT04basic	0.643282
ICT04CIILC	0.660100
MeijiHILw1	0.374485
MeijiHILw3	0.371387
UAmsT04LnuNG	0.670690
uogWebSelL	0.424656

Table 2: a-values used for FitFuse the various input systems.

The results shown in Table 3 are promising. Having moved away from the perfectly accurate probability distribution

	MaxMAP	FitFuse	PosFuse	SlideFuse	CombMNZ
run1	0.5389	**0.5773**	0.5751	0.5697	0.3317
run2	0.5120	0.5640	**0.5679**	0.5651	0.5249
run3	0.4589	0.5140	**0.5375**	0.5223	0.1862
run4	0.4325	0.4704	**0.4791**	0.4628	0.1740
run5	0.3976	0.4724	**0.4907**	0.4640	0.4203

Table 3: MAP Scores when training on the actual result sets being fused. The highest MAP score for a fusion technique on each run is in bold.

used in PosFuse, FitFuse shows only a slight disimprovement in MAP score, despite it being merely an estimate of the probabilities involved. On the first run, it actually gains a marginally higher MAP score than PosFuse, which indicates that although estimating probability may not be expected to achieve the same quality results as a perfectly-modelled probability distribution, this is not necessarily the case.

4.3 Towards single-value training

Because of the onerous training needs, we are interested in finding other values that can be substituted for a fitted a-value in a $y = \frac{a}{x}$ style probability model. In order for a candidate value to be suitable for use in this way, it is required to satisfy three criteria:

- Correlation: It must be shown to correlate to the fitted a-values.
- Training: It should require less exhaustive training calculations than methods such as PosFuse and SlideFuse.
- Results: It must be competitive in terms of the evaluation of fusion performance.

The first of these criteria is particularly important for selecting what value to use. Logically, a high a-value indicates that an input system is more likely to return relevant documents than one with a lower a-value. Clearly, this a-value is linked to the overall performance of an input system. As such, this motivates the use of established IR evaluation metrics for fusion.

Evaluation metrics have been a key focus of IR research for many years. Each is designed to measure the quality of a result set in some way, with different metrics having their own emphasis, strengths and weaknesses. The metric selected for investigation in this work is the widely-used Mean Average Precision (MAP) metric.

Figure 2 shows a graph of the fitted a-values (on the x-axis) plotted against MAP (on the y-axis). Although the two values are not shown to be directly proportional, a clear upward trend can be seen. Input systems with higher fitted a-values tend to also have higher MAP scores. Intuition would dictate that this is not a surprising result: a system with a greater tendency to return relevant documents would typically achieve a higher MAP score on evaluation (although the position of the relevant documents is also important).

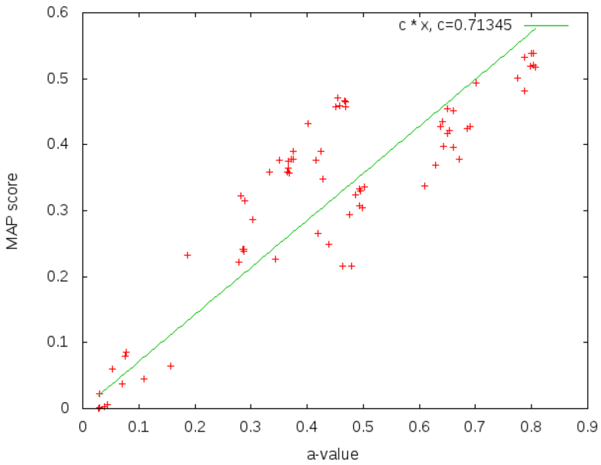

Figure 2: Correlation between MAP score and fitted a-value

A curve can be fitted for this graph also (and is indicated by the straight line in Figure 2). However, it important to bear in mind that a direct mapping from a MAP score to an a-value is not necessarily required. The aim of the fusion task is to rank the documents, rather than calculate an accurate a-value. Given the formula used below in Section 5, it can be shown that multiplying the MAP scores by a constant to better approximate a-values does not affect the ranking of the documents in the fused result set.

The second criterion required above is that the amount of training effort should be less than that of alternative techniques such as SlideFuse. SlideFuse requires knowledge of the exact positions in which relevant documents were returned in response to training queries. In contrast the proposed approach requires only a single-value estimation of the quality of the input system.

Finally, it is required to evaluate the effectiveness of using a function of MAP score and document position as a fusion strategy.

5. EVALUATION

In order to evaluate the effectiveness of the approach outlined above, it is necessary to carry out a number of experiments. These experiments involved running this new technique (which we shall call “MAPFuse”) alongside a number of alternatives.

5.1 Experiment Setup

As with the initial experiments outlined in Section 3, five separate runs were performed. These use the same inputs as in the earlier experiments. Each input was divided into a set of training result sets and a set of fusion result sets. For this, an 20%/80% split was used (i.e. 45 training queries and 180 queries used for fusion).

Dividing query sets in this way alone may cause unrepresentative results being obtained. For instance, early queries (i.e. those used for training) may be disproportionately straightforward (or indeed difficult) when compared with those used for fusion. This may mean that differences between fusion techniques' performance may be a consequence of the training data rather than the algorithms themselves.

For this reason, each of the fusion runs was performed five separate times, with the queries being shuffled into a different randomised order before each time. Thus the set of training queries was different each time. The evaluation results reported here for each run are the average of each of these five sets of shuffled inputs.

The baseline MAP score for each run is that achieved by the best-performing individual input system (denoted by MaxMAP). Training queries are ignored in this calculation, so the figures presented relate to the same query set for each technique.

Three data fusion algorithms were chosen for comparison. SlideFuse and CombMNZ are implemented as described in [10] and [5], respectively. Training queries are also ignored for CombMNZ, since that algorithm does not require a training phase. The implementation of PosFuse is as described in Section 3, with the exception that the probabilities are calculated on the training queries and then used to fuse the fusion queries at a later stage, rather than being calculated on the same result sets that are to be fused.

5.2 Defining MAPFuse

For the MAPFuse fusion algorithm, the training phase requires only that the MAP score for each input system on the training queries be calculated. This is performed by *trec_eval*, which is a tool provided by TREC to calculate evaluation metrics for IR systems. Unlike the proof-of-concept results shown in Section 3, relevance information for the actual result sets being fused is not required, as the MAP scores used for fusion are calculated using only the training queries.

Once the relevant MAP scores have been calculated, they are used in the fusion phase to calculate the scores on which the documents are ranked in the final, fused result set.

The ranking score R_d attributed to document d is given by

$$R_d = \sum_{s \in S} \frac{MAP_s}{p_s(d)} \tag{10}$$

where S is the set of the input systems that returned document d somewhere in their result sets, MAP_s is the MAP score associated with system s and $p_s(d)$ is the position in which document d was ranked by system s.

The fact that the MAP score is divided by a document's position helps to leverage the Skimming Effect, whereas the fact that the scores are added to give the document's final ranking score boosts documents that have appeared in multiple result sets and so makes use of the Chorus Effect.

5.3 Results

The results of running these experiments are presented in Table 4. Values in bold face are the highest score achieved by a fusion algorithm on a particular run. Asterisks are used to indicate a statistically significant difference to the performance of MAPFuse when measured using the t-test.

With the exception of CombMNZ, each of the fusion algorithms achieves a higher MAP score than that of the best individual input system (again shown as "MaxMAP"). MAPFuse achieves comparable results to PosFuse and SlideFuse, with the highest MAP score for three of the five runs. It also shows a statistically significant improvement in MAP score over MaxMAP and CombMNZ on all runs. As an aside, it is of note that, unlike the situation in Section 3, the scores achieved by SlideFuse are consistently higher than those of PosFuse (with the sole exception of run5). This may possibly be explained by the lower quantity of training queries being used, with the sliding window beginning to show its advantages over the strictly position-based PosFuse.

	MaxMAP	MAPFuse	PosFuse	SlideFuse	CombMNZ
run1	0.5468*	**0.5767**	0.5591*	0.5728	0.3361*
run2	0.5120*	0.5693	0.5682*	**0.5718**	0.5404*
run3	0.4555*	0.5132	0.5045*	**0.5171**	0.1842*
run4	0.4357*	**0.4711**	0.4591	0.4665	0.1785*
run5	0.4084*	**0.4858**	0.4777*	0.4681*	0.4277*

Table 4: MAP Scores From Fusion Runs. The highest MAP score for a fusion technique on each run is in bold. Asterisks indicate a statistically significant difference when compared to MAPFuse using the t-test.

The difference between the MAP scores of the three probabilistic techniques is quite small (MAPFuse's score is never more than 3% higher than PosFuse or 4% higher than that of SlideFuse). It is notable that despite this, the difference between MAPFuse and PosFuse is statistically significant in 4 of the 5 runs. Another observation is that for the two runs in which SlideFuse achieves a higher MAP score than MAPFuse, this difference is not statistically significant.

Despite these observations, the aim of the experiment is not necessarily to achieve significantly higher MAP scores. According to the third success criterion in Section 4, we merely require comparable performance with competing techniques. The principal advantage is that comparable retrieval results can be achieved by using only a single figure to represent the effectiveness of an underlying input system, rather than the detailed information about relevant documents' positions that is required by the other algorithms.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have examined the use of the probability of relevance in performing data fusion. In this context, we use "probability of relevance" to mean the probability that a document returned by a particular input system in a particular position in its result set is relevant.

Initially we showed that if a fully accurate model of the probability of relevance at each position is available, positive fusion results can be achieved using these probabilities to calculate the ranking scores for documents. Following from this, we have shown that these probabilities can be modelled by a function of the form $y = \frac{a}{x}$. Using the MAP score of each input system on a number of training queries as an substitute for a, we have shown that comparable MAP scores to alternative fusion algorithms can be achieved.

The benefits of this approach are principally in the level of training data that is required. Whereas algorithms like SlideFuse required detailed training data on the specific location of relevant documents within result sets, MAPFuse requires only a single summary metric to represent the quality of each input system being used.

For the purposes of this paper, the common Mean Average Precision (MAP) evaluation metric was used as the single-value substitution for a. However, a range of alternative metrics are available and so future work will concentrate on evaluating the impact of using alternative metrics.

Additionally, a more exhaustive study on a greater number of document collections will be necessary to demonstrate the wider applicability of this work. Such a study would also include a separation of the training and fusion phases so that each is carried out on a different document collection (although the retrieval systems generating each result set would not change). This would be an important stage in demonstrating that this type of fusion could be employed in a real-world information retrieval system.

7. ACKNOWLEDGEMENTS

This material is based upon works supported by the Science Foundation Ireland under Grant No. 08/RFP/CMS1183.

8. REFERENCES

[1] J. A. Aslam and M. Montague. Models for metasearch. In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 276–284, New York, NY, USA, 2001.

[2] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, O. Frieder, and N. Goharian. Fusion of effective retrieval strategies in the same information retrieval system. *J. Am. Soc. Inf. Sci. Technol.*, 55:859–868, 2004.

[3] J. P. Callan, Z. Lu, and W. B. Croft. 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*, pages 21–28, New York, NY, USA, 1995.

[4] N. Craswell and D. Hawking. Overview of the TREC-2004 web track. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC-2004)*, 2004.

[5] E. A. Fox and J. A. Shaw. Combination of Multiple Searches. In *Proceedings of the 2nd Text REtrieval Conference (TREC-2), National Institute of Standards and Technology Special Publication 500-215*, pages 243–252, 1994.

[6] A. E. Howe and D. Dreilinger. SavvySearch: A Metasearch Engine That Learns Which Search Engines to Query. *AI Magazine*, 18:19–25, 1997.

[7] J. H. Lee. Analyses of multiple evidence combination. *SIGIR Forum*, 31:267–276, 1997.

[8] D. Lillis. ProbFuse: Probabilistic Data Fusion. Msc, University College Dublin, UCD, February 2006.

[9] D. Lillis, F. Toolan, R. Collier, and J. Dunnion. ProbFuse: A Probabilistic Approach to Data Fusion. In *Proceedings of the 29th annual international ACM SIGIR Conference on Research and Development in information retrieval*, pages 139–146, New York, USA, 2006.

[10] D. Lillis, F. Toolan, R. Collier, and J. Dunnion. Extending Probabilistic Data Fusion Using Sliding Windows. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR '08)*, volume 4956 of *Lecture Notes in Computer Science*, pages 358–369, Berlin, 2008. Springer.

[11] R. Manmatha, T. Rath, and F. Feng. 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*, pages 267–275, New York, NY, USA, 2001.

[12] M. Montague and J. A. Aslam. Relevance score normalization for metasearch. In *CIKM '01: Proceedings of the Tenth International Conference on Information and Knowledge Management*, pages 427–433, New York, NY, USA, 2001.

[13] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 538–548, New York, NY, USA, 2002.

[14] A. L. Powell, J. C. French, J. Callan, M. Connell, and C. L. Viles. 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*, pages 232–239, New York, NY, USA, 2000.

[15] E. Selberg and O. Etzioni. The MetaCrawler Architecture for Resource Aggregation on the Web. *IEEE Expert*, pages 11–14, 1997.

[16] M. Shokouhi. Segmentation of Search Engine Results for Effective Data-Fusion. *Advances in Information Retrieval*, 4425, April 2007.

[17] C. C. Vogt and G. W. Cottrell. Fusion Via a Linear Combination of Scores. *Information Retrieval*, 1:151–173, 1999.

[18] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. The Collection Fusion Problem. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 95–104, 1994.

[19] S. Wu and F. Crestani. Data fusion with estimated weights. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 648–651, New York, NY, USA, 2002.