UCD SIFT in the TREC 2010 Web Track: Notebook Paper

David Leonard, Lusheng Zhang, David Lillis,
Fergus Toolan, Rem W. Collier, and John Dunnion

School of Computer Science and Informatics
University College Dublin, Ireland
{david.leonard, lu-sheng.zhang, david.lillis,
fergus.toolan, rem.collier, john.dunnion}@ucd.ie

Abstract. The SIFT (Segmented Information Fusion Techniques) group in UCD is dedicated to researching Data Fusion in Information Retrieval. This area of research involves the merging of multiple sets of results into a single result set that is presented to the user. As a means of both evaluating the effectiveness of this work and comparing it against other retrieval systems, the group entered Category B of the TREC 2010 Web Track. This involved the use of freely-available Information Retrieval tools to provide inputs to the data fusion process.
This paper outlines the strategies of the 3 candidate fusion algorithms entered in the ad-hoc task, discusses the methodology employed for the runs and presents a preliminary analysis of the provisional results issued by TREC.

1 Introduction

This is the second year of the SIFT (Segmented Information Fusion Techniques) project's participation in the TREC Web Track. In an effort to build on the experience gained in last year's competition and test some of the modifications made to our approach, it was once again decided to enter Category B. The principal aim of the SIFT group is to develop data fusion algorithms that combine the outputs of multiple Information Retrieval (IR) systems or algorithms in order to produce a single result-set that is of a superior quality. It should therefore be emphasised that the motivation behind our entry was not to evaluate novel IR systems or algorithms, but rather to investigate methods that may be used to combine these. In order to achieve this, the method employed uses implementations of standard, off-the-shelf IR algorithms (available as open source software) as the base systems for fusion and subsequently layers the fusion algorithms on top of these. This year's entry comprised three runs submitted to the ad-hoc task, with the result sets for each generated using a different fusion technique developed within the group.

The paper is organised as follows: Section 2 gives a short introduction to the area of Data Fusion. Sections 3, 4 and 5 provide implementation details for the three data fusion techniques that each constituted our entry for one of

the runs. The procedures used to tune the parameters of these algorithms for the submitted runs, in addition to details of the component IR systems, are described in Section 6. Preliminary results are presented in Section 7. Possible directions for future entries are discussed in Section 8.

2 Data Fusion

Data Fusion is an IR technique for combining the ranked lists returned by different component IR systems in response to a query. The goal is to produce an aggregate ranked list with improved performance over each of the individual lists. An inherent assumption within the data fusion context (as distinct from the related concept of *collection fusion*) is that each system retrieves from the same document collection. Techniques for fusion may be decomposed into two broad categories based on the level at which they access information:

1. **Rank-based:** the fusion algorithm is restricted to accessing the linearly scaled ranked lists output by the component systems and is not privy to the degrees of confidence underpinning these rankings. Such algorithms include approaches based on interleaving [1] and voting-based techniques [2, 3]
2. **Score-based:** the fusion algorithm may also take into account the relevance scores of the documents in the ranked list. These are the internally generated real numbers used by each IR system as a basis for calculating the rankings. Linear combination [4, 5] and the popular CombSum and CombMNZ algorithms [6, 7] are examples of methods based on relevance scores. These categories may be further sub-divided in accordance with whether they require training data to tune the parameters of the algorithm.

When choosing how to fuse the ranked lists returned by multiple IR systems, Vogt and Cottrell proposed certain intuitive "effects" that may be taken into consideration [4]. The first of these, 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. Secondly, 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 non-relevant ones [7]. Fusion algorithms that attach greater importance to documents that are returned by multiple input systems attempt to exploit this effect.

The fusion algorithms which were used to generate the results sets for the runs submitted to the ad-hoc task are part of a family of rank-based fusion techniques that may be termed "probabilistic". They are probabilistic in the sense that they attempt to build a model of the ranking behaviour of each component system, which may subsequently be used to estimate the probability that a document returned by that system at a particular rank will be relevant.

A training phase is utilised to gather statistics about the past performance of each system from which such a probability distribution may be approximated. At the fusion stage this probability information is used as a means to combine and re-rank the documents returned by each system in response to a query. The primary difference between the fusion strategies relates to the nature of the approximation of the probability distribution i.e. the degree to which it accurately reflects the true distribution.

3 ProbFuse

Probfuse is a rank-based data fusion algorithm that attempts to model the characteristic ranking behaviour of each component system using a probability distribution [8]. It adopts a coarse-grained approach to the estimation of such a distribution, which is based on the notion of segmentation. The key idea is to divide a ranked list, often referred to as a result set, into a series of consecutive equal-sized segments spanning a range of rank positions. After segmentation, a training phase is undertaken to calculate the probability that a document returned at a position lying within a particular segment is relevant. These probability values are later used during the fusion phase to produce a single aggregated ranked list.

3.1 Training Phase

The training phase is undertaken using a dataset consisting of a collection of result sets for which relevance judgments are available. The model of probability used by ProbFuse requires that each result set be divided into x segments of equal size. The division of a simple 12-document result set into segments is illustrated in Figure 1. Here, the leftmost result set is seen to be divided into two segments, with half of the documents appearing in each. Examples are also shown for increasing values for x, resulting in greater numbers of segments being created.

The objective of the training phase is to ascribe probabilities to each segment that will represent the likelihood that a document appearing in that segment will be relevant to any given topic. These probabilities may be calculated using the following formula:

$$P(d_k|S) = \frac{\sum_{q \in Q} \frac{R_{k,q}}{K}}{Q} \tag{1}$$

where $P(d_k|S)$ represents the probability that a document d returned by the system S in segment k is relevant, $R_{k,q}$ is the number of documents in segment k that are judged to be relevant to topic q, K is the total number of documents in segment k and Q is the set of training topics. The outcome of the training phase is a set of probability values associated with the segments belonging to each of the systems.

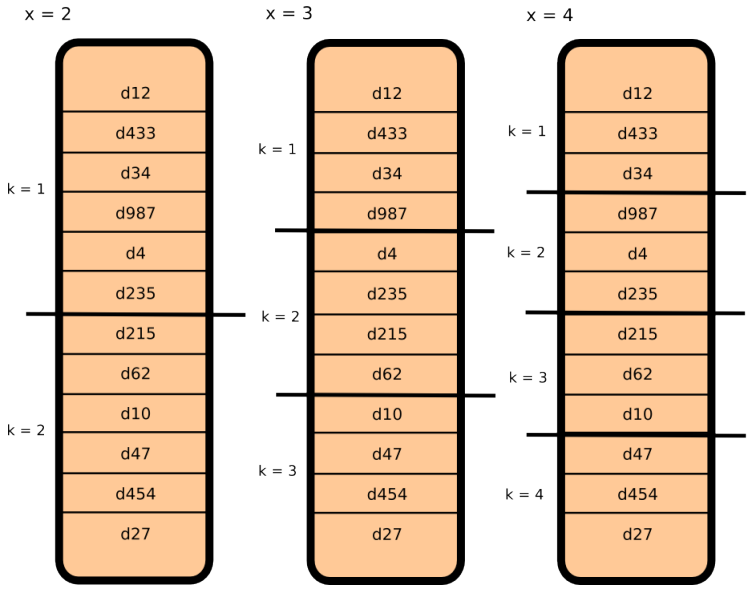

Fig. 1. Segmenting a result set for different values of x

3.2 Fusion Phase

Having obtained, for each system, a set of values estimating the probability that a document returned in each segment is relevant, the next step of the process is fusion. In this phase, each document is examined and its position in each of the result sets to be fused is noted. Depending on the segment the document is returned in, each system may then contribute towards that document's final ranking score, with no contribution occurring from any system that fails to return the document. The ranking score R_d for each document d is given by the following equation:

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

where M is the number of retrieval models to be fused, $P(d_k|S)$ is as outlined above and k is the segment in which d is returned by system s (1 for the first segment, 2 for the second, etc.). Once R_d has been calculated for each document, the documents are then merged into the final result set, sorted in descending order of R_d.

4 SlideFuse

SlideFuse is a variation on the method adopted for modelling probability distributions used in ProbFuse [9]. In particular, it attempts a more fine-grained ap-

proximation of the true underlying distribution (i.e. in contrast to the segmented approach, where the probability values apply to ranges of rank positions, it attempts to calculate the probability that a document returned at each position in a result set is relevant). For a training set of topics, this may be computed using the following formula:

$$P(d_p|s) = \frac{\sum_{q \in Q} R_{d_p,q}}{Q} \tag{3}$$

where, $P(d_p|s)$ is the probability that a document d returned by input system s in position p of a result set is relevant, $R_{d_p,q}$ is the relevance of the document d, at position p, to the topic q (1 if the document is relevant, 0 if not) and Q is the set of training topics. In practice, however, a problem arises when using the above formula to calculate such probabilities, due to the presence of un-judged documents in the result sets i.e. documents for which no relevance information is available. During the training procedure, it is quite likely that there may be many positions at which only judged non-relevant or un-judged documents are returned. Unfortunately, this leads to a zero value for the probabilities of relevance calculated for these positions.

In order to address this problem and obtain a smoother, more representative, probability distribution the concept of a sliding window is introduced. Instead of focusing on individual positions, as above, the probability values for the surrounding positions are also taken into consideration and an average value calculated as follows:

$$P(d_{p,w}|s) = \frac{\sum_{i=a}^{b} P(d_i|s)}{b - a + 1} \tag{4}$$

In the above equation, $P(d_{p,w}|s)$ is the probability of relevance of a document d returned in position p using a window of size w either side of p, $P(d_i|s)$ is calculated using Equation refeqn:rank and a and b are, respectively, the beginning and end positions that delimit the window. The size of the window, or number of neighbouring positions that are taken into account on each side of a position, is fixed for each ranked list with a suitable value for this parameter being determined empirically. An illustration of the smoothing effect of the sliding window is shown in Figure 2 for a sample input system.

The primary difference between the method adopted in ProbFuse lies in the fact that for SlideFuse the window or segment used to associate a probability value with each position is now always centred about the position. The combination strategy used to calculate the final ranking scores, R_d, at the fusion stage is very similar to that given in equation 2, with the exception that $P(d_{p,w}|s)$ is now substituted for $P(d_k|s)$ and the scaling parameter k is no longer required. This function is presented in Equation 5.

$$R_d = \sum_{s=1}^{M} P(d_{p,w}|s) \tag{5}$$

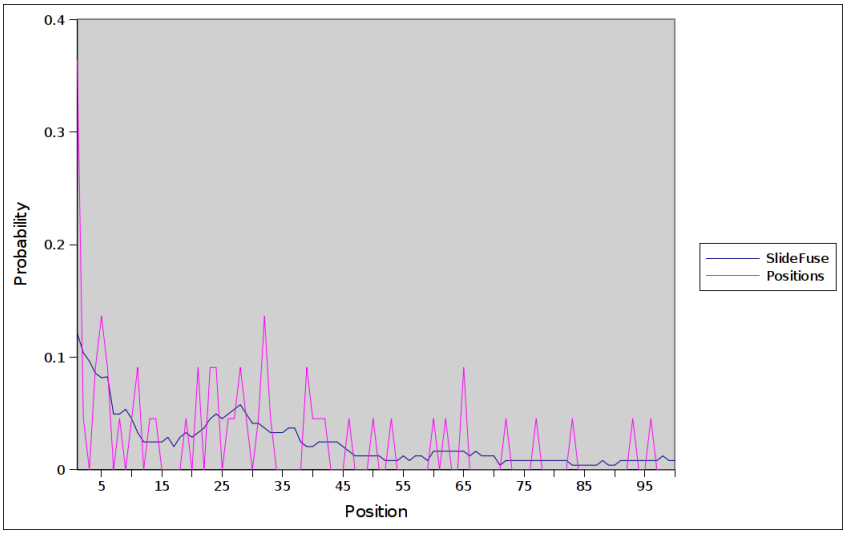

Fig. 2. Probability Distribution using SlideFuse

5 MAPFuse

MAPFuse is a fusion technique designed to address the extensive data/training demands of earlier probabilistic algorithms such as ProbFuse and SlideFuse [10]. It attempts to formulate a universal probabilistic model that may be used to characterise the ranking behaviour of IR systems. The aim is to then estimate the parameters of this model using less data while simultaneously preserving fusion performance. At its core, it postulates a hyperbolic approximation of relationship between the position of a document in a ranked list returned by an IR system and the probability of relevance of the document to a query. In effect, a weight is associated with each input system based on past performance (similar in many respects to the score-based technique of Linear Combination [4]) which is then used in conjunction with the rank of each document to scale the contribution of the documents returned by that system to the fused result set.

In initial experiments carried out to explore MAPFuse, the MAP score achieved for training queries (MAP_s) was used as the weight associated with each system. The probability of relevance at a given position p was then estimated by

$$P(d_p|s) = \frac{MAP_s}{p} \tag{6}$$

This was found to be correlated with a curve fitted to the probability of relevance when estimated at each individual position in the result set (as calculated using Equation 3). As such, the final ranking score R_d of document d could be calculated as follows:

$$R_d = \sum_{s \in S} \frac{MAP_s}{p_s(d)} \tag{7}$$

where $p_s(d)$ is the position in which input system s returned document d.

6 TREC 2010 Experiments

In order to prepare for entry into the competition a number of decisions needed to be taken in relation to the experimental setup. In particular it was necessary to select both a suitable training dataset and also the input systems to be used during fusion.

6.1 Training Data

As discussed above, each fusion algorithm requires a training phase to tune the parameters of the models that are built of the input systems. Ideally, for fusion to be successful, the data on which this training occurs should provide a representative sample that will be sufficient to capture the ranking behaviour of the models on future queries. In an effort to fulfil this requirement, the training strategy adopted was to use the ClueWeb09 Category B document collection in conjunction with the topics and relevance judgments available from TREC Web Track 2009 [11]. The parameters for segment and window size required respectively by ProbFuse and SlideFuse were chosen based on successful performance in previous empirical work [8, 9] i.e. the segment size used was 25 and the window size for was 5.

6.2 Input Systems

In order to focus development work on the design of fusion techniques the philosophy of the group is to use freely available open source IR software as a means for generating inputs to the fusion process. Two such packages provided the backbone for this year's entry.

- **Terrier**: Terrier (TERabyte RetrIEveR) is an open-source search engine developed at the University of Glasgow and released under the Mozilla Public License [12]. Terrier is specifically designed to be capable of handling large-scale document collections, on the order of terabytes. This, coupled with the fact that it offers implementations of a variety of document ranking models, made it an attractive choice for providing the inputs to the fusion process.
- **Lemur**: The Lemur Project[1] was started in 2000 by the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts, Amherst, and the Language Technologies Institute (LTI) at Carnegie Mellon University. Indri is an open-source search engine, released as part of this project, which provides state-of-the-art text search and a rich structured query language for text collections of up to 50 million documents.

[1] http://lemurproject.org

It was required that a subset of these IR algorithms be selected to generate the inputs to the fusion process. In order to accomplish this, each of the techniques provided by Terrier was run on topics from TREC Web Track 2009 and the 3 best performing systems chosen. These were DFR_BM25, PL2 and TF_IDF. In addition to these, the stand-alone Indri search engine was also selected. The same four systems were used as inputs to each of fusion algorithms.

7 Results

Based on the preliminary results issued by TREC, computed over 36 of the 50 topics, a provisional measure of the performance of the 3 fusion algorithms may be gauged. Table 1 displays the average values for a selection of the evaluation metrics computed for the ad-hoc task, with the highest value for each highlighted in bold type. With reference to this table it may be seen that on average SlideFuse performs best according to 5 out of 6 of the metrics. Slidefuse was also generally the best performing technique across the range of average statistics calculated by the trec_eval tool.

Table 1. Evaluation Results, with the highest score for each metric in bold

	ProbFuse	SlideFuse	MAPFuse
ERR@10	0.135	**0.156**	0.129
nDCG@10	0.074	**0.085**	0.075
P5	0.328	**0.333**	0.244
P10	0.300	**0.331**	0.250
bpref	**0.211**	0.208	0.204
MAP	0.108	**0.115**	0.108

The performance of our runs with respect to the other entrants is illustrated in Table 2, which shows the number of queries for which a run did better than or was equal to the median value of the two primary metrics ERR@10 and nDCG@10 (It should be noted that that there were 5 queries for which the median value was 0 for both measures). On a per query basis ProbFuse performed marginally better than SlideFuse with both techniques recording results better than or equal to the median on half the queries. To put these results into some further perspective, it was observed that the average difference between SlideFuse and the best performing system across the 36 queries was 0.421 for ERR@10 and 0.248 for nDCG@10.

The relatively strong performance of ProbFuse was surprising, given that it is the least smooth attempt to approximate the probability distribution of the constituent systems to be fused. On the other hand the comparatively poor performance of MAPFuse, which has shown promise in previous experimental work [10], was disappointing. A possible explanation for this may be its reliance

Table 2. The percentage of queries which did better than or was equal to the median value of the metrics

	ProbFuse	SlideFuse	MAPFuse
ERR@10	53%	50%	33%
nDCG@10	53%	50%	39%

on only a single summary statistic (MAP score) to characterise the behaviour of an IR system at individual rank level. It is also not clear whether the MAP score is the appropriate measure to use for parameterisation of the probability distribution on such large datasets. In contrast, SlideFuse exploits more detailed information/statistics about the behaviour of the system at each rank position and is therefore perhaps a more stable and accurate approximation of the underlying probability distribution. However, it should also be pointed out that the primary motivational scenario behind MAPFuse is for situations where such detailed information is not available.

Although the three fusion algorithms are not explicitly designed to optimise the criteria for the diversity task, Table 3 presents the results of our runs for the nERR-IA@10, α-nDCG@10 and P-IA@10 metrics. As above, the figures represent the number of queries for which our algorithms were better than or equal to the median value (these statistics are computed over the 88 runs submitted for both the ad-hoc and diversity tasks).

Table 3. Diversity task, % of queries better than or equal to median on each run

	ProbFuse	SlideFuse	MAPFuse
nERR-IA@10	53%	33%	44%
α-nDCG@10	47%	31%	42%
P-IA@10	50%	47%	44%

8 Future Work

The selection procedure used to determine the inputs to be used in the fusion phase of the runs relied solely on the individual performance of systems with little attention paid to the relationship between interactions amongst the systems and combined performance. One area of interest to the group is the formulation of metrics to capture complementary characteristics of the input systems stemming from their methodological differences. Such metrics would form the basis of more sophisticated selection strategies and perhaps more intelligent fusion techniques. Similarly, it would also be interesting to investigate whether such metrics could be leveraged effectively in fusion algorithms tailored to the diversity task.

References

1. Voorhees, E.M., Gupta, N.K., Johnson-Laird, B.: The Collection Fusion Problem. In: Proceedings of the Third Text REtrieval Conference (TREC-3). (1994) 95–104
2. Aslam, J.A., Montague, M.: Models for metasearch. In: SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, ACM Press (2001) 276–284
3. Montague, M., Aslam, J.A.: Condorcet fusion for improved retrieval. In: CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management, New York, NY, USA, ACM Press (2002) 538–548
4. Vogt, C.C., Cottrell, G.W.: Fusion Via a Linear Combination of Scores. Information Retrieval **1**(3) (1999) 151–173
5. Callan, J.P., Lu, Z., Croft, W.B.: 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, New York, NY, USA, ACM Press (1995) 21–28
6. Fox, E.A., Shaw, J.A.: Combination of Multiple Searches. In: Proceedings of the 2nd Text REtrieval Conference (TREC-2), National Institute of Standards and Technology Special Publication 500-215. (1994) 243–252
7. Lee, J.H.: Analyses of multiple evidence combination. SIGIR Forum **31**(SI) (1997) 267–276
8. Lillis, D., Toolan, F., Collier, R., Dunnion, J.: 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 (2006) 139–146
9. Lillis, D., Toolan, F., Collier, R., Dunnion, J.: Extending Probabilistic Data Fusion Using Sliding Windows. In: Proceedings of the 30th European Conference on Information Retrieval (ECIR '08). (31st March - 2nd April 2008) 358–369
10. Lillis, D., Zhang, L., Toolan, F., Collier, R.W., Leonard, D., Dunnion, J.: Estimating Probabilities for Effective Data Fusion. In: Proceedings of the 33rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland (2010)
11. Clarke, C., Craswell, N., Soboroff, I.: Overview of the TREC-2009 Web Track. In: In TREC2009: Proceedings of the 18th Text Retrieval Conference, 2009, Gaithersburg, United States (2009)
12. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Johnson, D.: Terrier information retrieval platform. In: Proceedings of the 27th European Conference on Information Retrieval (ECIR 05), Springer (2005) 517–519