UCD SIFT in the TREC 2011 Web Track

David Leonard, Doychin Doychev, David Lillis,
Fergus Toolan, Rem W. Collier, and John Dunnion

School of Computer Science and Informatics
University College Dublin, Ireland
{david.leonard, david.lillis, fergus.toolan,
rem.collier, john.dunnion}@ucd.ie,
doychin.doychev@ucdconnect.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 2011 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 entries submitted to compete in the ad-hoc task, discusses the methodology employed by them and presents a preliminary analysis of the results issued by TREC.

1 Introduction

This is the third 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, 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. The result sets for the 3 runs were generated using the fusion technique, SlideFuse, which was our best performing entry at TREC 2010. The entries differ, however, in respect of the number and type component systems fused as well as the amount of training data used in the construction of the fusion model.

The paper is organised as follows: Section 2 gives a short introduction to the area of Data Fusion. Section 3 provides implementation details for the data

fusion technique SlideFuse. 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 4. Preliminary results are presented in Section 5. Possible directions for future entries are discussed in Section 6.

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] and the popular CombMNZ algorithm [5] 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.

The fusion algorithm which was used to generate the results sets for the runs submitted to the ad-hoc task is 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.

3 SlideFuse

SlideFuse is a rank-based probabilistic data fusion algorithm that attempts to model the characteristic ranking behaviour of each component system using a probability distribution [6].

3.1 Training Phase

The input to the training phase is a dataset consisting of a collection of result sets for which relevance judgments are available and a set of component systems. For such a training set of topics and a component system, the objective of the training phase is to ascribe probabilities to each ranked position, in a results list, that will represent the likelihood that a document appearing at this position will be relevant to any given topic. These probabilities may be calculated using the following formula:

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

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 training 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. 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 1 for a sample input system.

3.2 Fusion Phase

The output of the training phase is, for each component system, a set of values estimating the probability that a document returned at each rank position in the result list 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 position at which the document is returned, 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 equation 2, with M representing the number of input systems and $P(d_{p,w})$ the probability value based on the positions in the surrounding window, w:

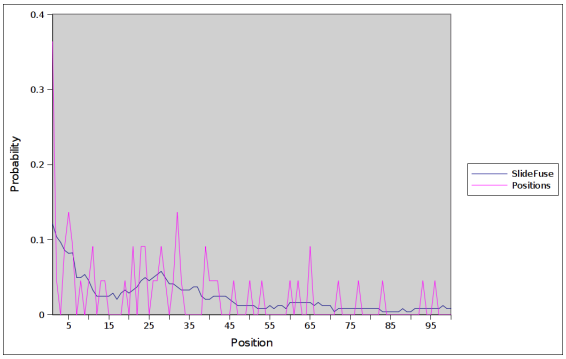

Fig. 1. Probability Distribution using SlideFuse

$$R_d = \sum_{s=1}^{M} P(d_{p,w}|s) \tag{2}$$

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 TREC 2011 Entries

In order to prepare for entry into the competition it was necessary to select both a suitable training dataset and also the input systems to be used during fusion.

4.1 Training Data

As discussed above, the 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 and 2010 [7, 8]. The value for the window size parameter required by SlideFuse, 5, was chosen based on successful performance in previous empirical work [6].

4.2 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, Indri[1] and Terrier [9], provided the backbone for this year's entry.

It was required that a subset of the IR algorithms, available in these packages, be selected to generate the inputs to the fusion process. Last year's entry focused on fusing together the output of algorithms from Terrier with the output generated by submitting "free text" queries to Indri. This year it was decided to take a different approach.

For the first entry, R1, the inputs to SlideFuse were generated by formulating the query separately on each of 5 fields associated with a document and querying Indri to return a ranked list for each e.g. by querying only on the *title* field, Indri returns documents that are restricted to contain the query terms in the title of the document. The fields chosen were *document*, *heading*, *inlinks*, *title* and *url*. In effect, this constitutes one potentially strong information source, the *document* field, and 4 weaker but, possibly more focussed information sources. The 100 queries from TREC 2009 and 2010, along with their associated relevance judgements, were used as training data for SlideFuse in R1. The second entry, R2, is the same as R1 but the model generated by SlideFuse is based on only 50 training queries, those taken from TREC 2009. The third entry, R3, fuses 3 algorithms from Terrier, In_expB2, PL2 and DFR_BM25, and is based on the same training data as R1. The constant in this years entry is the fusion technique, SlideFuse, and the variables are 1.) number of input systems 2.) type of component system and 3.) amount of training data used.

5 Results

5.1 Ad-hoc task

Baseline To put our results for the R1 and R2 entries on the ad-hoc task in context, we required a baseline. For this purpose, following the release of the relevance judgements for TREC 2011, we retrospectively ran this year's 50 TREC queries as "free text" queries on the Indri search engine. It is our understanding that this method ranks the documents by utilising information from all sources associated with a document. It therefore has access to the same information base that was used by R1 and R2. It should be noted, however, that due to the pooling of competition entries employed by TREC, direct comparison against a non-competiton entry is not correct e.g. across the 50 queries, 10% of the documents in the top 20 of the result lists returned for the "free text" queries were unjudged. Bearing this in mind, the results presented next should be viewed with caution. Table 1 shows the percentage improvement of the R1 entry over the baseline referred to above. It may be seen from this table that on the traditional *precision* metric, the improvement, 62%, is greatest for P5. Moving further down the ranked list this advantage is lessened, culminating with a 15% performance difference at P20. One of the reasons underlying this performance difference relates to the presence of spam e.g. R1 reduces the number of judged spam documents in the top 20 by 33% over the baseline.

[1] http://lemurproject.org

Table 1. Percentage improvement of R1 over baseline for the precision metric

Metric	P5	P10	P15	P20
% Improvement	62	34	24	15

Graded Relevance Metrics Table 2 gives a summary of how all 3 entries performed against each other on the nDCG@20 metric, alongside the averages of the scores for the *best*, *median* and *worst* results for all entries. Referring to table 2 it is observed that R1 and R2 achieve scores that are better than or equal to the *median* on 31 and 30 queries respectively (60% of the total). The average values for R1 and R2 are 0.2021 and 0.1953, both of which exceed the average value for the *median*, 0.1876. The difference in performance between R1 and R2 is not as great as expected, taking into account that the R1 fusion model is based on twice the amount of training data. R3 performed comparatively poorly, getting a score better than or equal to the *median* on 20 queries. The mean value of 0.1358 is also significantly lower than the *median* average. This is somewhat surprising given that the fusion was based on 2 years training data. The interpretation of this result will depend on an analysis of the performance of the component systems that were used as inputs to the fusion process.

Table 2. Results for R1, R2 and R3 on the nDCG@20 metric

Entry	R3	R2	R1
>= median	20	30	31
nDCG@20	0.1358	0.1953	0.2021
All entries	Best	Median	Worst
nDCG@20	0.5370	0.1876	0.0106

5.2 Diversity task

The primary evaluation metrics for the diversity task are the so-called *cascade* measures. There has been much recent debate about the effectiveness/behaviour of these metrics and it has been proposed that they achieve a balance between novelty and overall precision in result lists [10]. Because our entries were not optimised to compete in the diversity task, we feel that our results may be useful as a baseline with respect to this debate. The ERR-IA metric is taken as the representative *cascade* metric for evaluation of our performance in the diversity task (the *Spearman* correlation coefficient between the results for ERR-IA@20 and alpha-nDCG@20 on our entry R1 was 0.97).

ERR-IA metric It is noted that absolute values of this metric for ambiguous queries should be viewed with respect to how it is calculated using the *ndeval* tool i.e. the scheme adopted by *ndeval* rewards systems which return documents that capture many facets of a topic and are positioned in the very top positions of the ranked lists. For an ambiguous query such as e.g. "source of the nile", it is unlikely that such an ideal is achievable.

The breakdown of our results on the ERR-IA@20 metric for our best performing entry, the R1 system, is presented in tables 3 and 4. Table 3 segments the data along 3 lines: 1.) The number of subtopics for a query – taking into account the 4 queries for which subtopics were removed, 2.) the number of queries in that category, 3.) the average value of ERR-IA@20, for R1, in this context. The similarity of the average scores across the different number of subtopics is a feature of table 3, although perhaps this should be conditioned on the number of queries available in each category. Table 4 shows the same data as table 3, with the number of subtopics for a query replaced by an indicator of whether the query was faceted (F.) or ambiguous (A.). A failure of R1 to return a document relevant to more than one subtopic in the first 2 ranking positions is a contributory factor to the low average value, 0.35, for ERR-IA@20 on the ambiguous queries.

Table 3. Breakdown of the average value of ERR-IA@20 for R1

Number of subtopics	2	3	4	5	6
ERR-IA@20	0.45	0.45	0.44	0.51	0.47
Number of queries	6	29	11	3	1

Table 4. Breakdown of the average value of ERR-IA@20 for R1

Query Type	F.	A.
ERR-IA@20	0.48	0.35
Number of queries	41	9

Figure 2 plots the value of the ERR-IA metric for the R1 entry at the first 20 rank positions. With reference to this figure it may be seen that the majority of the gain is attained in the first 5 positions. The curve flattens out after rank position 10. To study what happens in this region, figure 3 presents a zoomed in view of the graph in figure 2. Additional context is provided by plotting results on the metric for 2 hypothetical systems *Ideal* and *Worst*. Starting with the value of ERR-IA at rank position 10, the *Ideal* system maximises the value

of the metric attainable in the remaining positions, whereas the *Worst* system returns documents that are not relevant to any subtopics from ranks 11 to 20. It may be seen from this that there is still room for improvement up to rank 15, however improvements for subsequent positions are difficult to quantify.

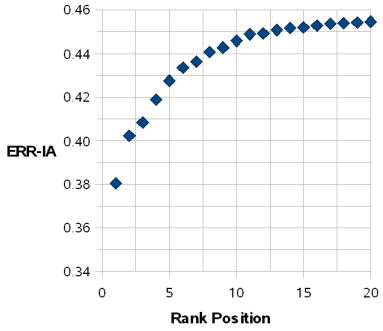

Fig. 2. Plot of ERR-IA against Rank Position for the R1 entry

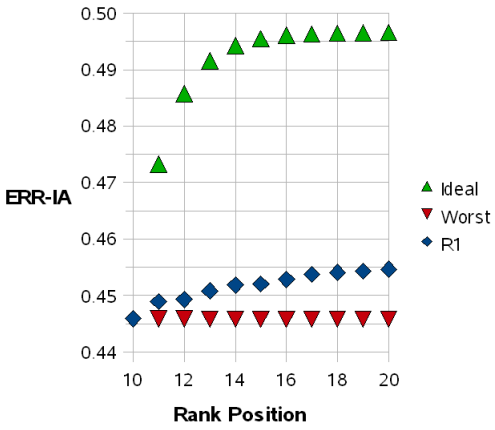

Fig. 3. Plot of ERR-IA against Rank Position for the R1 entry *vs* the hypothetical *Ideal* and *Worst* systems

Table 5 presents an overview of how all 3 entries performed relative to each other and the averages of the scores for the *best*, *median* and *worst* results on the ERR-IA@20 metric. R1 and R2 maintain their performance from the ad-hoc

task, by achieving a score above or equal to the *median* on roughly 60% of the queries. Their average values across the 50 queries, 0.4546 and 0.44, are also above the *median* average of 0.4079. Similar to the results from the *ad-hoc* task, the difference between the performance of R1 and R2 is surprisingly small. The poor performance of R3 on the diversity task is more pronounced than on the *ad-hoc* task, scoring better than or equal to the median on just 12 queries and with an average value of 0.291. It should be stressed though, that the fusion technique is not optimised for this task.

Table 5. Results for R1, R2 and R3 on the ERR-IA@20 metric

Entry	R3	R2	R1
>= median	12	27	30
ERR-IA@20	0.291	0.44	0.4546
All entries	Best	Median	Worst
ERR-IA@20	0.7441	0.4079	0.0346

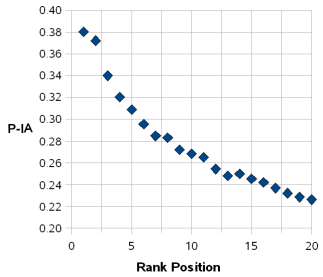

Fig. 4. Plot of P-IA against Rank Position for the R1 entry

P-IA metric In order to gain a different perspective on the behaviour of our entries in the diversity task it is instructive to assess the results of a *non-cascade* measure. Figure 4 plots the results for the R1 entry on the P-IA metric against rank position. With reference to this figure it may be seen that after the first 2 rank positions there is a noticeable drop, with a steadily decreasing trend in the values thereafter. Taken together with figure 2, above, this graph appears to support the intuition that our results in the diversity task are primarily based on the ability of the R1 and R2 entries to return documents in the first 2 rank positions that satisfy the diversity criteria encouraged by the *cascade* metrics.

6 Future Work

To date, the analysis of this year's results is preliminary and definite conclusions will require a deeper analysis of the probability models generated by each entry. There are, however, a number of possible directions for future work to take. The models of the input systems, generated from the training data by SlideFuse, do not take *graded* relevance information into account i.e. an input system returning highly relevant or key documents to a query, receives the same credit as one returning documents that are relevant but perhaps not *essential*. Intuitively, this seems to be a sub-optimal approach. A step forward in this direction would be to model separately the probability of an input system returning highly relevant documents and incorporate this into the fusion process. A second area that may lead to improved performance, would be to learn to rank the input systems prior to the training phase of the fusion process. In particular, for the simpler input systems used by the R1 and R2 entries this would be expected to yield better results.

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

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. Lee, J.H.: Analyses of multiple evidence combination. SIGIR Forum **31**(SI) (1997) 267–276
6. 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
7. 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)
8. Clarke, C., Craswell, N., Soboroff, I., Cormack, G.: Overview of the TREC-2010 Web Track. In: In TREC2010: Proceedings of the 19th Text Retrieval Conference, 2010, Gaithersburg, United States (2010)
9. 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
10. Clarke, C.L.A., Craswell, N., Soboroff, I., Ashkan, A.: A comparative analysis of cascade measures for novelty and diversity. In: WSDM'11. (2011) 75–84