Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information Processing and Management 42 (2006) 899–915

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Performance prediction of data fusion for information retrieval

Shengli Wu *, Sally McClean

School of Computing and Mathematics, University of Ulster, Shore Road, Newtownabbey BT370QB, Northern Ireland, UK

Received 30 April 2005; accepted 4 August 2005
Available online 5 October 2005

Abstract

The data fusion technique has been investigated by many researchers and has been used in implementing several information retrieval systems. However, the results from data fusion vary in different situations. To find out under which condition data fusion may lead to performance improvement is an important issue. In this paper, we present an analysis of the behaviour of several well-known methods such as CombSum and CombMNZ for fusion of multiple information retrieval results. Based on this analysis, we predict the performance of the data fusion methods. Experiments are conducted with three groups of results submitted to TREC 6, TREC 2001, and TREC 2004. The experiments show that the prediction of the performance of data fusion is quite accurate, and it can be used in situations very different from the training examples. Compared with previous work, our result is more accurate and in a better position for applications since various number of component systems can be supported while only two was used previously.

Keywords: Data fusion; Multiple linear regression; Performance prediction

1. Introduction

The concept of data fusion initially occurred in multi-sensor processing. In the last 10 years or so, data fusion has been used by researchers in the information retrieval area to combine multiple document lists for the same information need. One particular situation is that we use several different information retrieval systems (or several different settings/retrieval strategies in the same system) to retrieve the same collection of documents, then merging these results into a single list for higher effectiveness. The feasibility of this solution mainly depends on whether we can obtain improvement on effectiveness by data fusion.

* Corresponding author. Tel.: +44 28 90 36 65 85.
E-mail addresses: s.wu1@ulster.ac.uk (S. Wu), si.mcclean@ulster.ac.uk (S. McClean).

0306-4573/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2005.08.004

Some early related work on data fusion is from Thompson (1990a, 1990b), Turtle and Croft (1991), Foltz and Dumais (1992), and Belkin, Cool, Croft, and Callan (1993, 1995). Thompson (1990a, 1990b) proposed a Bayesian fusion model called the Combination of Expert Opinion (CEO) model for the combination of expert opinions in probabilistic information retrieval. Turtle and Croft (1991) used independently-generated query representations to create a number of results within an inference network, and found that combining different query representations led to increased retrieval effectiveness over any single representation. Foltz and Dumais (1992) found similar improvements by combining results from multiple retrieval strategies. Belkin et al. (1993, 1995) conducted experiments with a 2GB TREC collection from TREC 1, and observed effectiveness improvement over a large number of combinations of different Boolean query representations. Later Saracevic and Kantor (1998) used independently-generated query representations to create a number of results, and found that a document was more likely to be relevant if it appeared in multiple results.

Quite a few data fusion methods such as CombSum (Fox & Shaw, 1994), CombMNZ (Fox & Shaw, 1994), linear combinations (Bartell, Cottrell, & Belew, 1994; Turtle & Croft, 1991; Vogt & Cottrell, 1998; Vogt & Cottrell, 1999; Wu & Crestani, 2002), Borda fusion (Aslam & Montague, 2001), Condorcet fusion (Montague & Aslam, 2002), Markov chain-based method (Renda & Straccia, 2003) have been proposed, and many experiments have been conducted to evaluate them. These experimental results are mixed: sometimes the fused results are better than every component result; sometimes they are not. On the other hand, most of the proposed data fusion algorithms are competitive in performance and there is no all-time winner. Therefore, to find out under which condition which data fusion methods can make improvement on effectiveness is an important issue.

Lee (1997) addressed this issue by conducting some experiments with CombMNZ and CombSum to support his overlap hypothesis: *Different runs might retrieve similar sets of relevant documents but retrieve different sets of non-relevant documents.* Furthermore, Lee defined two overlap coefficients R_{overlap} and N_{overlap}, which denotes the relevant and non-relevant documents in two retrieval results, respectively. Lee concluded that improvement on data fusion could be observed if there was a greater overlap of relevant documents than of non-relevant documents among component results. Also Lee suggested a linear transformation method for score normalisation. All scores are in the range $[0, 1]$, with the minimal score mapping to 0 and the maximal score to 1. This method is referred to Zero-one later in this paper.

Montague and Aslam (2001) suggested two other linear transformation methods Sum and ZMUV (Zero-Mean and Unit-Variance). In Sum, the minimal score is mapped to 0 and the sum of all scores in the result to 1. In ZMUV, the average of all scores is mapped to 0 and their variance to 1. Some experiments were conducted to compare the effect of different score normalisation methods on data fusion. Their experiments show that Sum and ZMUV can achieve significant improvement over Zero-one in performance for both CombSum and CombMNZ.

Wu and McClean (submitted for publication) reviewed three linear score normalisation methods: Zero-one, Sum, and ZMUV. Through comparison analysis and extensive experimentation, they concluded that Zero-one is very likely the best method among these three methods.

Vogt and Cottrell (1998, 1999) analysed the performance of the linear combination algorithm using linear regression. In their experiments, two systems were always used for fusion, which is the simplest situation. Fourteen variables were used in the analysis: two different performance measures (one of them was average precision and the other was a statistical measure of rank correlation between the system and the relevance judgement) of each system, the number of relevant documents returned by one system but not the other divided by the total number of relevant documents returned by that system, the similarity of two results' rankings and others. The performance analysis and prediction for the fused result was very accurate ($R^2 = 0.94$). However, for the prediction of performance improvement of the fused result over the best component system, their analysis model was not useful ($R^2 = 0.06$).

Ng and Kantor (2000) focused on predicting if the performance of CombSum is better than all component systems or not. They used several different statistical techniques: linear analysis, multiple linear regression, logistic regression and a non-parametric training and testing method which they called the bin-ranking method. They also used two systems for each fusion, as Vogt and Cottrell did in (1998, 1999). Two variables were used: performance ratio of two systems, and a measure of the dissimilarity between two systems. They found that the two variables were informative to predict if the fused result was better than both component systems and the detection rate of their approach was about 70–75%. Also they used multiple linear regression to predict the performance improvement of the fused result over the better one among two results. A R^2 of 0.204 was observed.

More recently, Beitzel et al. (2004) conducted some experiments to compare the performances of CombMNZ using several different groups of systems. They observed no improvement when fusing results from three different retrieval strategies in the same information retrieval system, while the merged result was better than the best system when choosing the top three systems submitted to TREC 6, 7, 8, 9, and 2001. In all these cases, relevant overlap was greater than non-relevant overlap. Therefore, they argued that greater overlap of relevant documents than of non-relevant documents, which was proposed by Lee, was not a very good indicator for fusion improvement.

In this paper we focus on the use of multiple regression to analyse component results to identify variables that may affect data fusion. Based on that, we predict the performance of data fusion algorithms such as CombMNZ and CombSum. Ng and Kantor's work (2000) is the most relevant to this, since multiple linear regression, among other statistical techniques, was used in their work. However, there are several differences. Firstly, they only considered data fusion with two component results, while we consider more and variable numbers (3–10) of component results, which is a more general situation. Secondly, our variables used are different, and non-linear forms of variables were used by us. Thirdly, they focused on answering a yes/no question, which was: if the fused result was better than both component results; while our major goal is to predict the performance of data fusion algorithms. However, improvement detection is also achievable by our method.

The remainder of this paper is organised as follows: in Section 2 we describe the method we use, and in Section 3 we describe the analytical results. Three data fusion methods are used, which are CombSum, CombMNZ, and Round-robin. We mainly focus on prediction of the performance of the fused results, the prediction of the performance of the fused result over average performance of all component systems, and the prediction of the performance of the fused result over the best component system. Also we analyse and compare the predictive ability of several different variables used in our model and Ng and Kantor's work (2000). Section 4 presents some other observations from the experiment. Section 5 concludes the paper.

2. Method

Our overall approach is to run several fusion algorithms with a large number of combinations of results from actual IR systems, and to identify the variables, via multiple regression, that affect the performance of data fusion algorithms. With a few independent variables and one dependent variable, a multiple linear regression attempts to fit a linear model to data. As used by some other researchers, TREC data is very suitable for our purpose. In this study, we chose three groups of information retrieval results (also called runs or submissions), the first is a subset of 42 results submitted to TREC 6 ad hoc track, the second is a subset of 58 results submitted to TREC 2001 web track, and the third is a subset of 77 results submitted to TREC 2004 robust track. All these chosen results are satisfied with two conditions:

(1) Its mean average precision on all queries is above a threshold (0.15 is chosen). We do not include very poor results because we consider that they are not from proper information retrieval systems and we should avoid using them in data fusion for better effectiveness. Among all three collections, TREC

2001 is the poorest on average performance. 58 results out of a total number of 97 have a performance of over 0.15.

(2) Most results include 1000 documents for each query. But a few submitted results include very few documents. Removing those results provides us a homogeneous environment for the investigation.

From a group of results, we randomly chose a certain number (3–10) of results. We randomly chose 10,000 combinations for every number n ($3 \leqslant n \leqslant 10$). Three fusion methods, CombSum, CombMNZ and Round-robin, were used in the experiment. CombSum and CombMNZ work as follows. Suppose we have n results for the same query:

$$R_i = \{(d_1, s_{i1}), (d_2, s_{i2}), \ldots, (d_n, s_{im})\} \quad (1 \leqslant i \leqslant n)$$

Every result includes all m documents $d_1, d_2, \ldots, d_m$ and their corresponding normalised scores $\{s_{i1}, s_{i2}, \ldots, s_{im}\}$. Scores are normalised using Zero-one normalisation method and are in the range of $[0, 1]$. CombSum uses the following formula to calculate the score of every document:

$$\text{Sum\_score}(d_j) = \sum_{i=1}^{n} s_{ij}$$

And CombMNZ uses the following formula:

$$\text{MNZ\_score}(d_j) = \sum_{i=1}^{n} t_{ij} \sum_{i=1}^{n} s_{ij} (t_{ij} = 1 \text{ if } s_{ij} > 0;\ t_{ij} = 0 \text{ if } s_{ij} = 0)$$

Then we rank these merged documents according to their calculated scores. Round-robin chooses one document from each result in turn, deleting any document if it has occurred before. CombSum and CombMNZ are typical data fusion methods, while Round-robin only merges the multiple results but does not vote for documents' ranking for effectiveness improvement. We include this Round-robin method in order to observe the effect of voting on data fusion.

2.1. Variables

We consider several aspects: the average performance of all component systems, the standard deviation of the performance of all component systems, the number of results, and the correlation among component results. For performance evaluation, we use average precision, since it is a single value measure and convenient for us to use.

We calculate the mean average precision of every result over a certain number of queries (50 for TREC 6 and TREC 2001, 249 for TREC 2004[1]); and for each combination, we calculate the standard deviation of their mean average precision. How to decide the strength of correlation among two or more component results is a question that needs to be considered carefully. Note here we are not concerned about the difference/similarity of information retrieval processes which are used to retrieve documents, but only the final document results, though there is strong relation between the result we obtain from an information retrieval process for a given query and the information retrieval process itself (including many aspects such as retrieval strategies, query formations, system settings, and so on). We may have several different ways of calculating the correlation coefficient of two results over the same group of queries (e.g., Spearman correlation coefficient, Kendall's tau measure). However, we need to calculate $n(n-1)/2$ correlation coefficients for n

[1] Two hundred and fifty queries were used in TREC 2004 robust track. However, there is one query (number 672) whose relevant document set is not included in the official relevance judgements file "qrels.robust2004.txt". Therefore, we used 249 queries.

Table 1
Variables and their meanings

num	Number of results for fusion
o_rate	Overlap rate among all component results
m_av	Mean average precision of all component results
dev	Standard deviation of mean average precision of all component results
best	Mean average precision of the best component result

systems. Moreover, it is difficult to calculate the correlation among more than 2 results. Instead of using correlation coefficients, we therefore calculate the overlap rate among a group of results:

$$o\_rate = \frac{D_{\text{all}} - D_{\text{unique}}}{D_{\text{all}}}$$

where D_{all} is the number of documents in all results, and D_{unique} is the number of documents which only occur in any one of the results but not the others. We use this *o_rate* to describe the correlation among a group of results.

We list all variables used and their meanings in Table 1.

3. Regression analytical results

We set several different objectives (performance of the fused result, performance improvement rate of the fused result to the average of all component results, and performance improvement of the fused result to the best component result) as dependent variables in the multiple regression analysis to observe the effect on them of those defined variables. SPSS for Windows is used for the analysis.

3.1. Regression of data fusion performance

Let us see the performance analysis first. Tables 2–4 present the performances of CombSum, CombMNZ, and Round-robin for TREC 6, TREC 2001, and TREC 2004 respectively. The standardised coefficients of the resulting regression equation can be interpreted as indicating how much each variable contributes to the overall estimate of the dependant variable. Thus, a positive coefficient indicates that the corresponding variable should be maximised in order to maximize the performance. Conversely, a negative coefficient indicates that the variable should be minimised in order to maximize the performance. The actual coefficients of the regression equation is standardised based on the distribution of the individual independent variables, so that their magnitude can be compared. R^2 measures how well we can predict

Table 2
Effect of four variables on the performance of data fusion methods (TREC 6)

Variable	Standardised coefficients		
	CombSum	CombMNZ	Round-robin
num	0.445	0.484	0.200
o_rate	−0.283	−0.327	−0.082
m_av	0.576	0.613	0.800
dev	0.309	0.258	0.175
	$R^2 = 0.848$	$R^2 = 0.852$	$R^2 = 0.924$

Significance: 0.000 for all variables in all three methods.

Table 3
Effect of four variables on the performance of data fusion methods (TREC 2001)

Variable	Standardised coefficients		
	CombSum	CombMNZ	Round-robin
num	0.625	0.630	0.221
o_rate	−0.365	−0.402	−0.140
m_av	0.646	0.669	0.668
dev	0.177	0.172	0.283
	$R^2 = 0.816$	$R^2 = 0.807$	$R^2 = 0.837$

Significance: 0.000 for all variables in all three methods.

Table 4
Effect of four variables on the performance of data fusion methods (TREC 2001)

Variable	Standardised coefficients		
	CombSum	CombMNZ	Round-robin
num	0.727	0.714	0.188
o_rate	−0.392	−0.412	−0.155
m_av	0.931	0.929	0.915
dev	0.269	0.217	−0.121
	$R^2 = 0.778$	$R^2 = 0.791$	$R^2 = 0.894$

Significance: 0.000 for all variables in all three methods.

the performance knowing only the four independent variables in the model. For example, if the value of R^2 is 0.65, it means that the four variables explain 65% of the variation in the performance of the data fusion method. From Tables 2–4, we can also observe that significance for all independent variables is listed at the 0.000 level, which means that the p value is less than 0.0005 and all independent variables are statistically highly significant with a probability of over 99.95% ($1 - 0.0005 = 99.95\%$).

Comparing CombSum and CombMNZ, we find that they are very similar in many ways:

- all corresponding variables take similar values in both methods;
- their R^2 values are close;
- the Pearson correlation coefficients for the results of CombMNZ and CombSum are 0.933 (TREC 6), 0.990 (TREC 2001), and 0.991 (TREC 2004) which indicate that these two methods are strongly correlated to each other;
- the mean average precisions of 80,000 combinations on 50 queries are 0.3061 and 0.3051 for CombSum and CombMNZ, respectively in TREC 6; and they are 0.2551 and 0.2555 in TREC 2001. In TREC 2004, the figures on 249 queries are 0.3461 and 0.3431 for CombSum and CombMNZ, respectively.

Though all variables are highly significant, their effects on the performance of fused results are not the same. According to the absolute values of coefficients, we can rank the four variables in descending order according to their significances. For all three methods, the mean average precision of all results (*m_av*) is always the most significant variable. However, in Round-robin, *m_av* is in the dominating position since its coefficient value is much bigger than the others. This situation does not happen in either CombSum or CombMNZ. This is understandable because Round-robin fuses all component results in such a way that the order of a document is totally determined by its original position in one of the component results.

In all three methods, *o_rate* takes a negative value. This indicates that overlapping is harmful to the performance of data fusion in all cases. However, the effect of overlapping on these three methods is not the same. CombMNZ is the most sensitive one; Round-robin is the least sensitive one; while CombSum is in the middle. This is because CombMNZ heavily uses "the multiple evidence principle", which arranges the documents retrieved by multiple results in high priority, while Round-robin does not do this at all. When overlap rate is high, which means these results are not very different, to use methods such as CombMNZ cannot boost much the performance of the fused result.

The above multiple linear regression analysis assumes that all the relations are linear, which may not be appropriate for all variables. Therefore, we tried some variations. One variation is, instead of using only *dev*, we use both *dev* and dev^2, instead of using *num*, we use both *num* and *LN*(*num*), instead of using only *o_rate*, we use both *o_rate* and o_rate^2, also we use both *m_av* and $SQRT(m\_av) = (m\_av)^{1/2}$ to replace *m_av*. These changes lead to considerable improvement for both CombSum ($R^2 = 0.914, 0.872, 0.860$) and combMNZ ($R^2 = 0.916, 0.863, 0.871$), but only very slight improvement ($R^2 = 0.930, 0.848, 910$) for Round-robin. The scatter graphs of CombSum are shown in Figs. 1–3, for TREC 6, TREC 2001, and

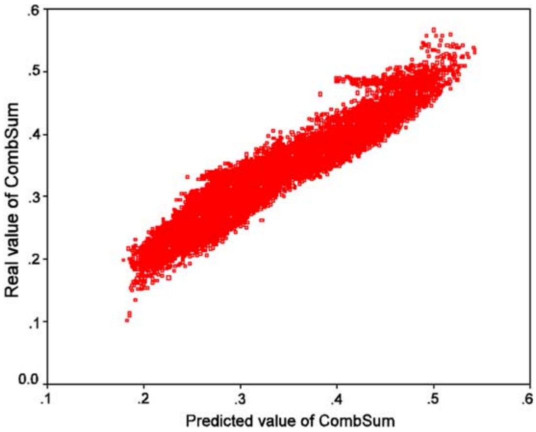

Fig. 1. Scatter graph of CombSum with predicted values vs. real values (TREC 6).

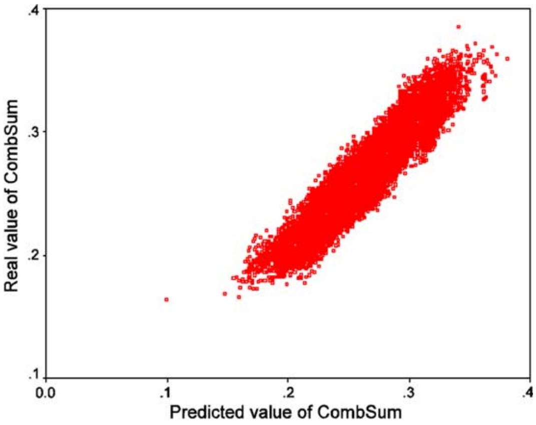

Fig. 2. Scatter graph of CombSum with predicted values vs. real values (TREC 2001).

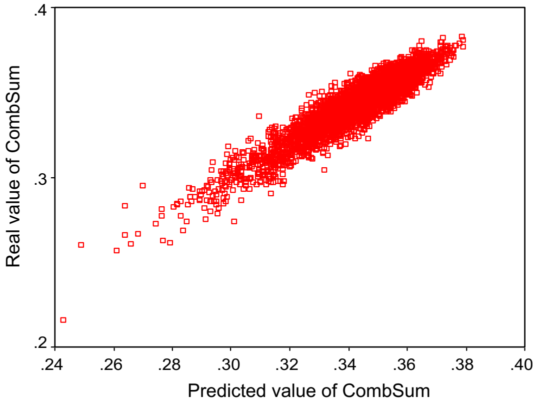

Fig. 3. Scatter graph of CombSum with predicted values vs. real values (TREC 2004).

TREC 2004, respectively. In these three figures, the x axis shows the predicted average precision of the fused result while the y axis shows the real average precision of the fused result with CombSum. If the prediction is 100% accurate, then all the points will take the same value on both x and y axes.

3.2. Regression of performance improvement over average performance

One issue that concerns us is which variables may lead to performance improvement for data fusion methods. We continue to use multiple regression to investigate this. All the variables used in the above analysis are kept the same; however, we change the dependent variable into performance improvement (percentage of performance improvement of data fusion over mean average performance of component results).

Tables 5–7 present the results for all three fusion methods. All four variables are statistically highly significant with only one exception. Comparing Tables 5, 6 and 7 with 2, 3, and 4, the orders of significance of these variables are very different. Instead of "mean average precision", "number of results" becomes the most significant variable, while "mean average precision" becomes the least significant one in all cases. In both CombSum and combMNZ, all these variables are ranked in the same order. Both "overlap rate" and "mean average precision" take negative values, which means "overlap rate" and "mean average precision" should be kept at the minimal level in order to obtain a maximal performance improvement. Conversely, "standard deviation of mean average precision" has a quite big positive coefficient, and we should boost this for improvement.

Let us consider an example. Suppose we have two groups of results, with each group includes 5 results, and the overlap rates of these two groups are the same. Five results in the first group have an average pre-

Table 5
Effect of several variables on the performance improvement of data fusion methods over average performance (TREC 6)

Variable	Standardised coefficients		
	CombSum	CombMNZ	Round-robin
num	0.827	0.916	0.567
o_rate	−0.519	−0.612	−0.237
m_av	−0.112	−0.094	−0.079
dev	0.500	0.419	0.471
	$R^2 = 0.565$	$R^2 = 0.553$	$R^2 = 0.404$

Significance: 0.000 for all variables in all three methods.

Table 6
Effect of several variables on the performance improvement of data fusion methods over average performance (TREC 2001)

Variable	Standardised coefficients		
	CombSum	CombMNZ	Round-robin
num	1.015	1.055	0.434
o_rate	−0.595	−0.636	−0.276
m_av	−0.004 (.282)	−0.032	−0.022
dev	0.260	0.246	0.532
	$R^2 = 0.534$	$R^2 = 0.537$	$R^2 = 0.395$

Significance: 0.000 for all variables (except one) in all three methods.

Table 7
Effect of several variables on the performance improvement of data fusion methods over average performance (TREC 2004)

Variable	Standardised coefficients		
	CombSum	CombMNZ	Round-robin
num	0.824	0.849	0.474
o_rate	−0.469	−0.468	−0.398
m_av	−0.246	−0.265	−0.483
dev	0.316	0.268	0.313
	$R^2 = 0.693$	$R^2 = 0.680$	$R^2 = 0.467$

Significance: 0.000 for all variables in all three methods.

cision of 0.2, 0.2, 0.3, 0.4, 0.4, respectively, and the average precision is 0.3 for every result in the second group. According to regression analysis, the first group is more likely to obtain better fusion result than the second group even though their mean average precision is the same in both groups. We explain this phenomenon like this: if some results are better than some others, then these good results are more likely to share some common opinion, and their common opinion will dominate the whole group; while those poor results share less common opinion, and their effect on fusion is limited. On the other hand, if all the results are close in performance, then no one result or several results can dominate the whole group, and less improvement can be made by data fusion.

Further improvement of the model is possible as in Section 3.1. When we use *dev*, dev^2, *num*, *LN*(*num*), *o_rate*, o_rate^2, *m_av*, and $SQRT(m\_av) = (m\_av)^{1/2}$, improvement can be observed for all methods in all situations. The R^2 values become 0.755 (TREC 6) and 0.688 (TREC 2001) and 0.804 (TREC 2004) for CombSum, 0.745 (TREC 6) and 0.683 (TREC 2001) and 0.797 (TREC 2004) for CombMNZ, and 0.461 (TREC 6) and 0.443 (TREC 2001) and 0.506 (TREC 2004) for Round-robin. Since all values of R^2 are bigger here than those with linear variables, the predictions here are more accurate.

We also observe that fused result is almost always better than the average of component results. The opposite situation rarely happens. Out of 240,000 combinations, 8 times it occurred for CombSum, 6 times for CombMNZ, and 25 times for Round robin. This demonstrates that data fusion methods are effective on improving the performance of the fused result over the average of component results.

3.3. Regression of performance improvement over best performance

We now use the performance improvement of data fusion over the best component result as the dependent variable to run the multiple regression analysis. Tables 8–10 show the results with five linear variables:

Table 8
Effect of several variables on the performance improvement of data fusion methods over best system (TREC 6)

Variable	Standardised coefficients		
	CombSum	CombMNZ	Round-robin
num	0.474	0.498	0.067
o_rate	−0.454	−0.487	−0.157
m_av	0.517	0.519	0.543
dev	−0.143	−0.184	−0.265
best	−1.083	−1.071	−1.044
	$R^2 = 0.654$	$R^2 = 0.694$	$R^2 = 0.889$

Significance: 0.000 for all variables in all three methods.

Table 9
Effect of several variables on the performance improvement of data fusion methods over best system (TREC 2001)

Variable	Standardised coefficients		
	CombSum	CombMNZ	Round-robin
num	0.543	0.571	−0.070
o_rate	−0.395	−0.422	−0.170
m_av	0.470	0.438	0.417
dev	−0.114	−0.116	−0.210
best	−1.169	−1.150	−0.938
	$R^2 = 0.788$	$R^2 = 0.791$	$R^2 = 0.827$

Significance: 0.000 for all variables in all three methods.

Table 10
Effect of several variables on the performance improvement of data fusion methods over best system (TREC 2004)

Variable	Standardised coefficients		
	CombSum	CombMNZ	Round-robin
num	0.853	0.820	0.153
o_rate	−0.488	−0.453	−0.167
m_av	1.088	1.075	0.711
dev	0.317	0.264	−0.289
best	−1.208	−1.201	−0.766
	$R^2 = 0.695$	$R^2 = 0.727$	$R^2 = 0.738$

Significance: 0.000 for all variables in all three methods.

num, *o_rate*, *m_av*, *dev*, and *best*. We observe that increasing the number of component results and increasing average performance of all component results are helpful, while higher overlap rate among results, diversified performances of component results, and especially lofty best results are very harmful for data fusion methods to outperform the best component result.

As in Sections 3.1 and 3.2, we can also increase the values of R^2 in these methods by introducing non-linear variables as in Sections 3.1 and 3.2. The R^2 values become 0.820 (TREC 6) and 0.864 (TREC 2001) and 0.811 (TREC 2004) for CombSum, 0.839 (TREC 6) and 0.863 (TREC 2001) and 0.835 (TREC 2004) for CombMNZ, and 0.840 (TREC 6) and 0.905 (TREC 2001) and 0.922 (TREC 2004) for Round-robin. Since all values of R^2 are bigger here than those in Section 3.2, the predictions here are more accurate than that in Section 3.2. Compared with Ng and Kantor's work (2000) and Vogt and Cottrell's work (1998, 1999)

(R^2 values are 0.204 and 0.06, respectively), our model is much more accurate. Moreover, our model allows variable number of component systems, while both models of Ng and Kantor's and Vogt and Cottrell's only allow two component systems.

Out of 80,000 combinations, 35,206 (44.0%) of the fused results using CombSum outperform the best component result, and 35,143 (43.9%) of the fused results using CombMNZ outperform the best component result in TREC 6. In TREC 2001, these two figures are 57,959 (72.4%) and 58,514 (73.1%). In TREC 2004, they are 68,160 (85.2%) and 64,321 (80.4%). Therefore, about 67% of the chances we observe that CombSum and CombMNZ are better than the best component result. For Round-robin, the figures are 11,514 (14.4%), 21,149 (26.4%), and 15,203 (19.0%), in TREC 6, TREC 2001, and TREC 2004, respectively. We also notice that the figures in TREC 6 are lower than that in TREC 2001 and TREC 2004. This is because a few component systems in TREC 6 are much better than the others, while the performances of all component results in TREC 2001 and TREC 2004 are close.

3.4. Performance prediction

We divided 50 queries in TREC 6 into two parts, the first part includes the first 25 queries and the second part the second 25 queries. The first half was used for training, and the second half was used for prediction. We calculated the mean average precision for every combination and every fused result, and then compared them with real values. The relative errors for CombSum, CombMNZ, and Round-robin are 0.0310, 0.0358, and 0.0278, respectively.

Next we used all 80,000 combinations in TREC 6 for training, and then use the formula obtained to predict the performance of 80,000 combinations of TREC 2001. The relative errors for CombSum, CombMNZ, and Round-robin are 0.0575, 0.0570, and 0.0350, respectively. Considering that the two groups of systems, document collections, and queries are totally different, this suggests that the analytical result is still useful even when we apply it in a very different situation from that used in training.

Discriminant analysis is discussed in (Ng & Kantor, 2000) and aims to predict if the fused result is better than the best component result. Our above analysis is applicable for the same purpose. For every combination, we calculate the mean average precision of the fused result *real_p*, and estimate the mean average precision of that *es_p* according to the multiple regression analysis, then we compare them with the mean average precision of the best result *best* to see how many times the judgement is correct by checking if (($real\_p > best$) and ($es\_p > best$)) or (($real\_p < best$) and ($es\_p < best$)) holds. For CombSum, the detection rates are 90.0% (TREC 6) and 93.2% (TREC 2001); for CombMNZ, the detection rates are 90.6% (TREC 6) and 92.8% (TREC 2001). Our result is better than that in Ng and Kantor's work (2000): 70% for testing runs and 75% for training runs.

If we do not have to make judgements for all the cases, then we can increase the correct detection rate by neglecting those cases which are on the margin of profit/loss for data fusion. Table 11 shows the detection rates of the prediction in various conditions for TREC 6 and TREC 2001. We check (($real\_p > best$) and ($es\_p > (1+k)best$)) or (($real\_p < best$) and ($es\_p < (1+k)best$)) holds for how many combinations with different k ($k = 0, 0, 01, \ldots, 0.10$). Generally speaking, the prediction is more accurate when the condition is more restrictive.

When using the multiple regression model to predict the performance of the data fusion, we need to assign values to those independent variables used. It is straightforward for the number of component systems and the overlap rate among component systems. For the rest two variables, the average performance of all component systems and the standard deviation of these performances, it is worth more consideration. For any given query, to obtain the exact values for these variables need to know the performances of all component systems, which demands document relevance judgements. It will not be realistic to do that each time for all component systems. Besides this, we may have two other options. The first is to evaluate the performances of all component systems using some training queries, and then we use these values from training

Table 11
Detection rate in different situations

Condition (k)	TREC 6		TREC 2001	
	CombSum (%)	CombMNZ (%)	ComSum (%)	CombMNZ (%)
0.00	90.0	90.5	93.2	92.8
0.01	92.0	93.2	95.2	94.8
0.02	94.2	95.2	96.7	96.4
0.03	96.0	96.7	92.8	97.6
0.04	97.5	97.8	98.6	98.5
0.05	98.4	98.6	99.1	99.0
0.06	99.0	99.1	99.4	99.4
0.07	99.4	99.4	99.7	99.7
0.08	99.7	99.7	99.8	99.8
0.09	99.8	99.7	99.9	99.9
0.10	99.9	99.9	99.9	99.9

queries for all test queries. The second is for every query, we estimate the performances of all component systems without document relevance judgements. Several methods (e.g., in Amitay, Carmel, Lempel, & Soffer, 2004; Soboroff, Nicholas, & Cahan, 2001; Wu & Crestani, 2003) on this issue have been proposed.

3.5. The predictive ability of variables

Although the same multiple regression technique has been used in Ng and Kantor's work (2000), but we use different variables. That is why we are able to achieve much more accurate prediction. Therefore, it is interesting to conduct an experiment to compare the predictive ability of those variables used in their model and/or our model. In Ng and Kantor's work, they used two variables to predict the performance of the fused result with CombSum: (a) a list-based measure of result dissimilarity and (b) a pair-wise measure of the similarity of performance of the two systems. The result dissimilarity of two systems is calculated as follows: for the same query, assume we obtain the same number (e.g., 1000) of retrieved documents from both systems. We merge these two results to obtain a larger group of documents (with n documents). For every possible combination of any two documents in this large group, we compare their respective rankings in both results. If the rankings are the same, a score of 0 is given; if the rankings are opposite, a score of 1 is given; if the situation is uncertain, a score of 0.5 is given, then we sum up all scores and divided it by $n(n-1)/2$, which is the maximal possible score for the two results. In this way we calculate a normalised score between 0 and 1 for any pair of results.

We used 42 systems in TREC 6 for the experiment. All possible combinations (861) of them were used for data fusion with CombSum, CombMNZ and Round-robin. We analysed these results using the multiple regression method with the same dependent but different independent variables. In such a way, we can decide the predictive ability of different variables. The experimental results are shown in Table 12.

From Table 12, we can observe a few things. Firstly, compare nos. 1 and 2, 3 and $4,\ldots,23$ and 24, the only difference between them is using *o_rate* in one case and *diss* in the other. In all pairs, using *o_rate* always leads to bigger R^2 values. The last column "Pair comparison" presents the increase rate of R^2 when using *o_rate* to replace *diss*. Therefore, we conclude that *o_rate* has more predictive ability than *diss*. Secondly, we may use *m_av* and *dev* to replace *ratio* to predict if the fused result is better than the best of the two results, or replace *first* and *second* to predict the performance of the fused result. However, in both cases, the substitute is not as good as the original one though the difference is not big. Thirdly, the prediction is very poor when we use *first* and *second* to predict the performance of the fused result and use *ratio* to predict if the fused result is better than the best of the two results. Therefore, related results are not presented. On the other hand, *m_av* and *dev* can be decently used in both situations.

Table 12
Predictive ability of different variables

No.	Method	Dependent variable	Independent variables	R^2	Pair comparison (%)
1	CombSum	*fused*	*o_rate*, *first*, *second*	0.932	2.31
2	CombSum	*fused*	*diss, first, second*	0.911	
3	CombSum	*fused*	*o_rate*, *m_av*, *dev*	0.915	1.67
4	CombSum	*fused*	*diss*, *m_av*, *dev*	0.900	
5	CombSum	*imp*	*o_rate*, *ratio*	0.409	10.84
6	CombSum	*imp*	*diss*, *ratio*	0.369	
7	CombSum	*imp*	*o_rate*, *m_av*, *dev*	0.399	11.76
8	CombSum	*imp*	*diss*, *m_av*, *dev*	0.357	
9	CombMNZ	*fused*	*o_rate*, *first*, *second*	0.927	3.11
10	CombMNZ	*fused*	*diss, first, second*	0.899	
11	CombMNZ	*fused*	*o_rate*, *m_av*, *dev*	0.909	2.60
12	CombMNZ	*fused*	*diss*, *m_av*, *dev*	0.886	
13	CombMNZ	*imp*	*o_rate*, *ratio*	0.403	13.52
14	CombMNZ	*imp*	*diss*, *ratio*	0.355	
15	CombMNZ	*imp*	*o_rate*, *m_av*, *dev*	0.401	14.57
16	CombMNZ	*imp*	*diss*, *m_av*, *dev*	0.350	
17	Round_robin	*fused*	*o_rate*, *first*, *second*	0.984	1.03
18	Round_robin	*fused*	*diss, first, second*	0.974	
19	Round_robin	*fused*	*o_rate*, *m_av*, *dev*	0.979	0.82
20	Round_robin	*fused*	*diss*, *m_av*, *dev*	0.971	
21	Round_robin	*imp*	*o_rate, ratio*	0.432	9.37
22	Round_robin	*imp*	*diss, ratio*	0.395	
23	Round_robin	*imp*	*o_rate*, *m_av*, *dev*	0.460	10.31
24	Round_robin	*imp*	*diss*, *m_av*, *dev*	0.417	

Note: fused denotes the performance (average precision) of the fused result, *imp* is a Boolean variable indicating if the fused result is better than the best of the two results, *o_rate* denotes overlap rate between two results, *first* denotes the average precision of the first result, *second* denotes the average precision of the second result, *diss* denotes the dissimilarity measure between the two results, *m_av* denotes the mean average precision of the two results, *dev* denotes the standard deviation of *first* and *second*, *ratio* denotes the ratio of performance of two results (the better one divided by the worse one, therefore, its value is always no less than 1).

Both *o_rate* and *diss* are used for the same purpose; it is interesting to investigate why *o_rate* has more predictive ability than *diss*. Because it seems that the calculation of *o_rate* is primitive and that of *diss* is more sophisticated. However, we notice that the ranking difference is not fully considered when calculating *diss*. Let us consider an example. Suppose we have two documents d_x and d_y. They occur in both results r_1 and r_2 but in different positions. The first case is: in r_1, d_x is in position 1 and d_y in position 2; while in r_2, d_x is in position 2 and d_y is in position 1. Since the rankings of these two documents are opposite in r_1 and r_2, a score of 1 is given. The second case is: in r_1, d_x is in position 1 and d_y in position 2; while in r_2, d_x is in position 500 and d_y is in position 1000. Since the rankings of these two documents are the same in r_1 and r_2, a score of 0 is given. However, these two scores are questionable. In the first case, the difference is tiny and the two results are very similar; while in the second case, these two results are very different by any means. Thus we hypothesize that is why *diss* is not as good as *o_rate* as indicator of results similarity (dissimilarity). Besides, for three or more systems, it is still very straightforward to calculate *o_rate*, *m_av*, and *dev*. However, how to calculate *diss* and *ratio* is not clear.

4. Some further observations

In Section 3, we have discussed several aspects which affect data fusion. Overlap rate among component results is one of them. Here we ignore the other aspects and focus on the overlap rate and its effect on data fusion. We divide the possible range of overlap rate [0, 1] into 20 ranges [0, 0.05], [0.05, 0.1], ..., [0.95, 1], then we observe the percentage of the improvement on performance that the fused result can obtain compared with the average performance of the component results. Figs. 4 and 5 show the curves of the percentage of improvement for CombSum in TREC 6 and TREC 2001, respectively. In Figs. 4 and 5, each curve is associated with a number, which is the number of results involved in the fusion. These two figures demonstrate that there is a strong relation between the overlap rate and the performance improvement percentage of data fusion. When overlap rate increases, the performance improvement percentage decreases accordingly. Besides, the figures also demonstrate that the number of results has considerable effect on data fusion as well. The curves for CombMNZ, which are not presented, are very similar to that for CombSum.

Another observation is about the distribution of the overlap rate among component results. Figs. 6 and 7 shows the distribution of overlap rate among 3, 4, ..., 10 results. In both figures, all curves should be well

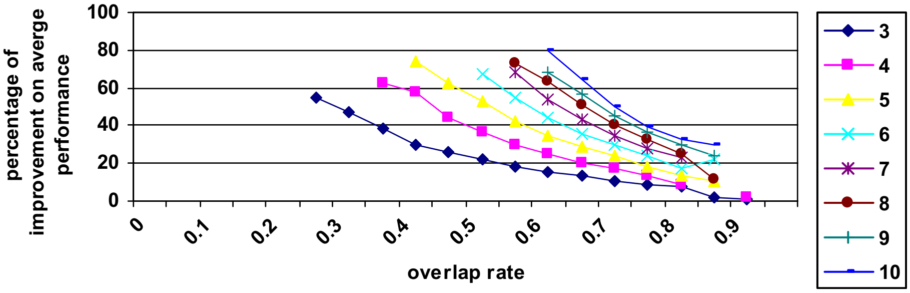

Fig. 4. Effect of overlap rate on the percentage of performance improvement (TREC 6, 3–10 results, CombSum).

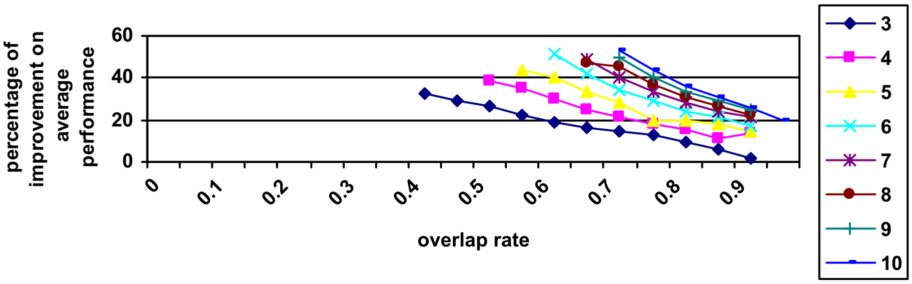

Fig. 5. Effect of overlap rate on the percentage of performance improvement (TREC 2001, 3–10 results, CombSum).

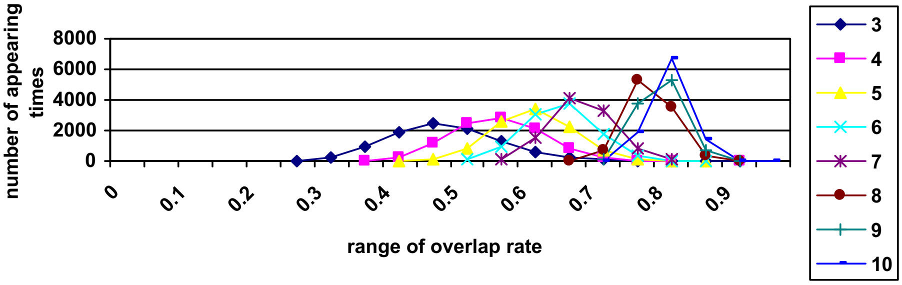

Fig. 6. Overlap rate distribution for 10,000 runs in TREC 6.

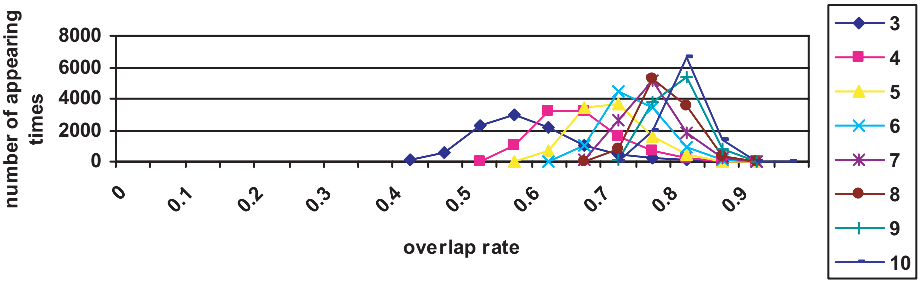

Fig. 7. Overlap rate distribution for 10,000 runs in TREC 2001.

described by normal distribution curves. Also we observe the same pattern of curves in both figures when we have the same number of results. Another seemingly interesting phenomenon we observe is: the more systems we put in data fusion, the bigger values of overlap rate we obtain from the results of these systems. From 3 to 10 systems, the increase of overlap rate is considerable and monotonous. It suggests that there are a few systems which are quite different from each other (the average overlap rate for three systems is the lowest, around 0.5 in TREC 6 and around 0.6 in TREC 2001), but the number of quite different systems cannot be large.

Besides the retrieval systems involved in the data fusion, several other factors, such as the document collections and the queries used and the number of documents which includes one or more query words, have influence on these curves. In TREC 6, the document collection included a little over half a million documents. A pool of 66,300 documents (which were the top 100 documents retrieved by each submitted system) were judged and 4611 (7.0%) were judged relevant for 50 queries. In TREC 2001, there were 1.69 million documents. For a total of 50 queries, 70,400 documents were judged and 3363 (4.8%) were judged either relevant (2573) or highly relevant (790). However, we do not have the figures for the number of documents which includes one or more query words. Comparing Fig. 6 with Fig. 7, we observe that curves in Fig. 7 are more compact with each other than in Fig. 6. We hypothesize that the all pairs of systems are more "similar" to each other in TREC 2001 than in TREC 6, which can explain the difference of overlap rate at the low end (3 systems). On the other hand, since fewer systems are involved and very likely more potential documents can be retrieved by these systems in TREC 2001 (considering the collection is three times as big as the collection used in TREC 6), Therefore, we can explain why the curves for 10 systems in both figures are almost in the same position.

The relation between the number of component results and fusion performance is an interesting issue. As we know, the more component results are used, the more improvement we can expect for the fused results, if all other conditions are keep the same. In Section 3, when we use both *LN*(*num*) and *num* to replace *num*, the prediction is more accurate in all cases. This suggests that it is better to use a logarithmic function than a linear function to describe the relation between the number of component results and fusion performance. This is confirmed by Fig. 8, in which the performance of CombSum is averaged for every certain number of component results. Each data value is the average of 10,000 combinations over 50 (for TREC 6 and TREC 2001) or 249 (for TREC 2004) queries. For the three curves observed in TREC 6, TREC 2001, and TREC 2004, we use linear model and logarithmic model to estimate them by SPSS. For all three curves, the logarithmic functions for the estimation are at a significance level of 0.0000 ($F = 326.9$, 299.5, and 342.6 for TREC 6, TREC 2001, and TREC 2004, respectively); while the linear model for the estimation is not as good as the logarithmic model: significance = 0.0002 and $F =$ 61.1 for TREC 6; significance = 0.0003 and $F = 58.3$ for TREC 2001; and significance = 0.0002 and $F = 63.2$ for TREC 2004.

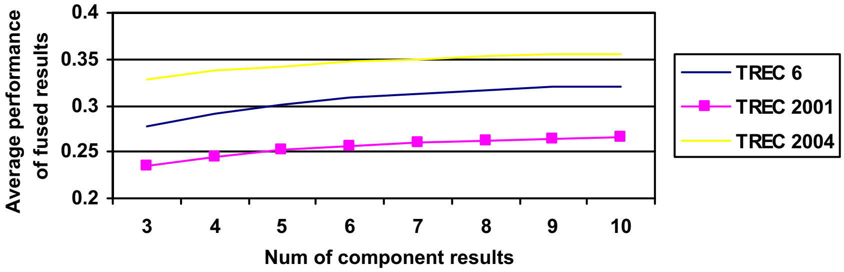

Fig. 8. Average performance of CombSum with a particular number of component results.

5. Conclusion

In this paper we have reported the result of a multiple regression analysis of three data fusion methods, CombSum, CombMNZ and Round-robin, with three groups of component results submitted to TREC 6, TREC 2001, and TREC 2005. Several different aspects, which are the number of component results, the overlap rate among the results, the mean average precision of the results, and the standard deviation of the mean average precision of the results are identified as highly significant variables which affect the performance of data fusion.

Our analysis provides quite accurate prediction of the performance of the fused result with CombSum and CombMNZ. When using linear variables, all methods obtain a R^2 value of between 0.778 and 0.852. The accuracy of the prediction can be improved by introducing nonlinear variables, and R^2 then ranges from 0.860 to 0.916. Especially using *LN*(num) to replace *num* (the number of results) can make considerable difference. When predicting the percentage of performance improvement of the fused result over the best component result, the prediction is quite accurate (R^2 values for all methods are between 0.811 and 0.863). Compared with Ng and Kantor's work (2000) and Vogt and Cottrell's work (1998, 1999) (they focus on fusing two component systems and R^2 values are 0.204 and 0.06, respectively), our model is more useful for real applications.

Though our major goal is to predict the performance of the fused result, the analytical result can also be used to predict if the fused result will be better than the best component result. Compared with Ng and Kantor's work (2000), our analysis is also more accurate in this situation. In all cases, a detection rate of 90% or over is observed; while in their work, about 70% of the detection rate is obtained for the testing runs and about 75% of the detection rate for the training runs. Besides, their analysis only considered the situation of two component results, while our analysis is working in a more general situation: more than 2 and variable numbers (3–10) of component results.

In our experiment with 240,000 combinations in all, we observe that almost all the fused results (using either CombSum or CombMNZ) are better than the average performance of component results, and some of the fused results (about 67% in our case, using either CombSum or CombMNZ) are better than the best component result. Another interesting observation is the normal distribution of overlap rate among component results. This should be useful for us to improve the data fusion algorithms. Our experiment also demonstrates that overlap-rate, one variable used in our model, has more predictive ability than the dissimilarity measure used in Ng and Kantor's work (2000).

In our study, two variables, which are mean average performance of all component systems and the standard deviation of the performances of these component systems, need document relevance judgements. Several methods (e.g., in Amitay et al., 2004; Soboroff et al., 2001; Wu & Crestani, 2003) have been proposed to estimate the performance of component systems without document relevance judgements. To use these methods and the multiple regression analysis to estimate the performances of component systems and also the fused result remains to be our future research.

References

Amitay, E., Carmel, D., Lempel, R., & Soffer, A. (2004). Scaling IR-system evaluation using term relevance sets. In *Proceedings of the 27th annual ACM SIGIR conference*, Sheffield UK, pp. 10–17.

Aslam, J., & Montague, M., 2001. Models for metasearch. In *Proceedings of the 24th annual ACM SIGIR conference*, New Orleans, USA, pp. 275–284.

Bartell, B. T., Cottrell, G. W., & Belew, R. K. (1994). Automatic combination of multiple ranked retrieval systems. In *Proceedings of the 17th annual ACM SIGIR Conference*, Dublin, Ireland, pp. 173–181.

Beitzel, S., Jensen, E., Chowdhury, A., Grossman, D., Frieder, O., & Goharian, N. (2004). On fusion of effective retrieval strategies in the same information retrieval system. *Journal of the American Society for Information Science and Technology, 55*(10), 859–868.

Belkin, N. J., Cool, C., Croft, W. B., & Callan, J. P. (1993). The effect of multiple query representations on information retrieval performance. In *Proceedings of 16th annual ACM SIGIR conference*, Pittsburgh PA, USA, pp. 339–346.

Belkin, N. J., Kantor, P., Fox, E. A., & Shaw, J. A. (1995). Combining evidence of multiple query representations for information retrieval. *Information Processing & Management, 31*(3), 431–448.

Foltz, P. W., & Dumais, S. T. (1992). Personalized information delivery: An analysis of information-filtering methods. *Communications of the ACM, 35*(12), 51–60.

Fox, E. A., & Shaw, J. A. (1994). Combining of multiple searches. In *Proceedings of the 2nd annual text retrieval conference (TREC-2)*, NIST, Gaithersburg, USA.

Lee, J. H. (1997). Analyses of multiple evidence combination. In *Proceedings of the 20th annual ACM SIGIR conference*, Philadelphia, USA, p. 276.

Montague, M., & Aslam, J. (2001). Relevance score normalization for metasearch. In *Proceedings of the 10th ACM CIKM conference*, Berkeley, USA, pp. 427–433.

Montague, M., & Aslam, J. (2002). Condorcet fusion for improved retrieval. In *Proceedings of the 11th ACM CIKM conference*, McLean USA, pp. 538–548.

Ng, K. B., & Kantor, P. B. (2000). Predicting the effectiveness of naive data fusion on the basis of system characteristics. *Journal of the American Society for Information Science, 50*(13), 1177–1189.

Renda, M. E., & Straccia, U. (2003). Web metasearch: Rank vs. score based rank aggregation methods. In *Proceedings of ACM 2003 symposium of applied computing*, Melbourne, USA, pp. 847–452.

Saracevic, T., & Kantor, P. (1998). A study of information seeking and retrieving. III: Searchers, searches, overlap. *Journal of the American Society of Information Science, 39*(3), 197–216.

Soboroff, I., Nicholas, C., & Cahan, P. (2001). Ranking retrieval systems without relevance judgements. In *Proceedings of the 24th annual ACM SIGIR conference*, New Orleans, LA, USA, pp. 66–73.

Thompson, P. (1990a). A combination of expert opinion approach to probabilistic information retrieval. Part 1: The conceptual model. *Information Processing and Management, 26*(3), 371–382.

Thompson, P. (1990b). A combination of expert opinion approach to probabilistic information retrieval. Part 2: Mathematical treatment of CEO model. *Information Processing and Management, 26*(3), 383–394.

Turtle, H., & Croft, W. B. (1991). Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems, 9*(3), 187–222.

Vogt, C. C., & Cottrell, G. W. (1998). Predicting the performance of linearly combined IR systems. In *Proceedings of the 21st annual ACM SIGIR conference*, Melbourne, Australia, pp. 190–196.

Vogt, C. C., & Cottrell, G. W. (1999). Fusion via a linear combination of scores. *Information Retrieval, 1*(3), 151–173.

Wu, S., & Crestani, F. (2002). Data fusion with estimated weights. In *Proceedings of the 11th ACM CIKM conference*, McLean, USA, pp. 648–651.

Wu, S., & Crestani, F. (2003). Methods for ranking information retrieval systems without relevance judgments. In *Proceedings of the 2003 ACM symposium on applied computing*, Melbourne Florida, USA, pp. 811–816.

Wu, S., & McClean, S. (submitted for publication). Reexamining score normalization methods in data fusion.