Information Processing and Management 45 (2009) 413–426

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Assigning appropriate weights for the linear combination data fusion method in information retrieval

Shengli Wu [a,*], Yaxin Bi [a], Xiaoqin Zeng [b], Lixin Han [b]

[a] School of Computing and Mathematics, University of Ulster, Newtownabbey, UK
[b] Department of Computer Science, Hohai University, Nanjing, China

ARTICLE INFO

Article history:
Received 16 June 2008
Received in revised form 13 February 2009
Accepted 20 February 2009
Available online 21 March 2009

Keywords:
Data fusion
Information retrieval
The linear combination method
Weight assignment

ABSTRACT

In data fusion, the linear combination method is a very flexible method since different weights can be assigned to different systems. However, it remains an open question which weighting schema should be used. In some previous investigations and experiments, a simple weighting schema was used: for a system, its weight is assigned as its average performance over a group of training queries. However, it is not clear if this weighting schema is good or not. In some other investigations, different numerical optimisation methods were used to search for appropriate weights for the component systems. One major problem with those numerical optimisation methods is their low efficiency. It might not be feasible to use them in some situations, for example in some dynamic environments, system weights need to be updated from time to time for reasonably good performance. In this paper, we investigate the weighting issue by extensive experiments. The key point is to try to find the relation between performances of component systems and their corresponding weights which can lead to good fusion performance. We demonstrate that a series of power functions of average performance, which can be implemented as efficiently as the simple weighting schema, is more effective than the simple weighting schema for the linear data fusion method. Some other features of the power function weighting schema and the linear combination method are also investigated. The observations obtained from this study can be used directly in fusion applications of component retrieval results. The observations are also very useful for optimisation methods to choose better starting points and therefore to obtain more effective weights more quickly.

1. Introduction

Information retrieval as a core technology has been widely used for the WWW search services and digital libraries. In recent years, an increasing number of researchers have been working in this area and many different techniques have been investigated to improve the effectiveness of retrieval. Quite a large number of retrieval models have been proposed and experimented with test document collections. For example, in the book "Modern Information Retrieval" written by Baeza-Yates and Ribeiro-Neto (1999), 11 different retrieval models were discussed. In addition, some other aspects such as user relevance feedback, document representation, query representation, query expansion, phrase recognition, thesaurus, context analysis, structure analysis, link analysis, and so on, may have a considerable impact on retrieval results. It is very likely that no single information retrieval system is able to deal with all these aspects at the same time, let along deal with them in the same way as some other systems. Therefore, many information retrieval systems developed are close in effectiveness but

* Corresponding author. Tel.: +44 2890366585.
E-mail address: s.wu1@ulster.ac.uk (S. Wu).

0306-4573/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2009.02.003

different from each other in implementation. In such a situation, data fusion, which uses a group of information retrieval systems to search the same document collection, and then merges the results from these different systems, is an attractive option to improve retrieval effectiveness. Recently, meta-search becomes an application of the data fusion technique. If the document collections across different web services are more or less the same, then the data fusion methods can be applied directly; if the document collections are quite different, then some variations of the data fusion methods can be used for obtaining more effective results (Wu & McClean, 2007).

Quite a few data fusion methods such as CombSum (Fox, Koushik, Shaw, Modlin, & Rao, 1993; Fox & Shaw, 1994), CombMNZ (Fox et al., 1993; Fox & Shaw, 1994), the linear combination method (Bartell, Cottrell, & Belew, 1994; Vogt & Cottrell, 1998, 1999), Borda fusion (Aslam & Montague, 2001), the probabilistic fusion method (Lillis, Toolan, Collier, & Dunnion, 2006), the correlation method (Wu & McClean, 2005, 2006a), Markov chain-based methods (Dwork, Kumar, Naor, & Sivakumar, 2001; Renda & Straccia, 2003), Condorcet fusion (Montague & Aslam, 2002), and the multiple criteria approach (Farah & Vanderpooten, 2007) have been proposed, and extensive experiments using TREC data have been reported to evaluate these methods. Experimental results show that, in general, data fusion is an effective technique for improvement of effectiveness, and very often the fused results are better than the best component results involved.

The linear combination data fusion method is a very flexible method since different weights can be assigned to different systems. In some related experiments, (e.g., Aslam & Montague, 2001; Thompson, 1993; Wu & Crestani, 2002; Wu & McClean, 2006a), a simple weighting schema was used: for a system, its weight is set as its average performance over a group of training queries. This weighting schema is straightforward and can be calculated or updated easily, therefore it is especially suitable in a very dynamic environment. However, it has not been investigated how good this weighting schema is.

In some previous researches, different numerical optimisation methods such as golden section search (Vogt & Cottrell, 1998, 1999) and conjugate gradient (Bartell et al., 1994) were used to search suitable weights for component systems. One major drawback of using these optimisation methods is their low efficiency. In some situations, such as the WWW, and digital libraries, where documents are updated frequently, each component system's performance may vary considerably from time to time. The weights for the systems should be updated accordingly. In such a situation, it may not be feasible to use those very time-consuming weighting methods to update weights frequently.

In this paper, we investigate the weighting issue through extensive experiments. The key point is to try to find the relation between performances of component systems and their corresponding weights which can lead to good fusion performance. As we shall see later in this paper, a power function weighting schema with a power of between 2 and 8, is more effective than the simple weighting schema for data fusion. On the other hand, those power function weights can be obtained as efficiently as simple weights. In fact, the simple weighting schema can be regarded as a special case of a power function weighting schema with a power of 1.

The rest of this paper is organized as follows: in Section 2 we review some related work on data fusion, especially on the linear combination method. Section 3 describes the linear combination method and the weighting issue. Section 4 discusses an experiment to evaluate different performance weighting schemas. Further observations from the experiment are discussed in Section 5. Section 6 provides a few conclusive remarks.

2. Previous work on data fusion

Usually relevance-related scores are provided by information retrieval systems for all retrieved documents. Some algorithms, such as CombSum (Fox et al., 1993; Fox & Shaw, 1994), CombMNZ (Fox et al., 1993; Fox & Shaw, 1994), the linear combination method (Bartell et al., 1994; Thompson, 1993; Vogt & Cottrell, 1999; Wu & Crestani, 2002), the probabilistic fusion method (Lillis et al., 2006), and the correlation method (Wu & McClean, 2005, 2006a), make use of relevance-related scores assigned to documents in component retrieval results. Others, such as Borda fusion (Aslam & Montague, 2001), Markov chain-based methods (Dwork et al., 2001; Renda & Straccia, 2003), Condorcet fusion (Montague & Aslam, 2002), and the multiple criteria approach (Farah & Vanderpooten, 2007) make use of the rank that each document occupies in each component result, as the scores are not always available.

Relevance-related scores obtained from different information retrieval systems may be diverse. Usually it is impossible to compare them directly and some kind of score normalization is required. Linear score normalization methods have been discussed in Lee (1997), Montague and Aslam (2001) and Wu et al. (2006), and non-linear score normalization methods have been discussed in Manmatha, Rath, and Feng (2001) and Nottelmann and Fuhr (2003). Since different retrieval systems use different ways to score documents, different score normalization methods may be required for different retrieval systems to obtain better effectiveness. However, the linear score normalization method has been widely used before in data fusion experiments.

CombSum, CombMNZ and some other methods were investigated by Fox et al. (1993) and Fox and Shaw (1994). They found that CombSum and CombMNZ outperformed the others. CombSum sets the score of each document in the fused result to the sum of the scores obtained by the component result, while in CombMNZ the score of each document is obtained by multiplying this sum by the number of results which have non-zero scores.

Lee (1997) conducted an experiment with six submitted results to TREC 3. He found that CombMNZ was slightly better than CombSum in his experiment. However, later experiments, for example, in Montague and Aslam (2001), Lillis et al. (2006), Wu, Crestani, and Bi (2006) and Wu and McClean (2006b) and others, found that Lee's conclusion is not always true and the probability that CombSum and CombMNZ are better than each other is roughly the same.

In the following we review a few papers which are very relevant to the linear combination data fusion method. Thompson (1993) used the linear combination method to fuse results submitted to TREC 1. He found that the combined results, weighted by performance level, performed no better than a combination using uniform weights (CombSum). This performance level weighting schema has been referred to as the simple weighting schema in this paper.

Bartell et al. (1994) used a numerical optimisation method, conjugate gradient, to search good weights of different systems. Because the method is time-consuming, only 2–3 component systems and top 15 documents returned from each system for a given query were considered in their investigation.

Vogt and Cottrell (1998, 1999) analysed the performance of the linear combination method by linear regression. In their experiments, they used all possible pairs of 61 systems submitted to the TREC 5 ad-hoc track. An optimisation method, golden section search, was used to search good system weights. Due to the nature of the golden section search, only two component systems can be considered for fusion.

Human relevance judgments are needed to evaluate the performances of component systems in order to decide the appropriate weights for them. Without any human relevance judgment involved, several methods (e.g., Nuray & Can, 2006; Soboroff, Nicholas, & Cahan, 2001; Wu & Crestani, 2003) could automatically estimate each component result's performance, then the linear combination method can be used for data fusion. However, performance estimation from these methods are usually not very accurate.

Unlike other studies (e.g., Aslam & Montague, 2001; Thompson, 1993; Wu & Crestani, 2002) which only concern performance, both system performance and dissimilarities between results were considered in Wu and McClean (2005, 2006a). In this weighting schema, any system's weight is a compound weight which is the product of its performance weight and its dissimilarity weight. Here the performance weight is also defined as the average performance of the system over a group of training queries, while the dissimilarity weight is defined as the average dissimilarity between the system in question and all other systems over a group of training queries. More recently, the combination of performance weights and dissimilarity weights was justified using statistical principles (Wu, 2009).

In general, the linear combination method is more effective than CombSum and CombMNZ in all above experiments conducted before except (Bartell et al., 1994). However, how to decide system weights is still an open question. Numerical optimisation methods can find effective solutions but too costly for many applications. This is especially the case when we do not have a good understanding of the problem and have to search in a very large area to find possible good solutions. On the other hand, the simple weighting schema can be implemented efficiently, but its effectiveness has not been investigated. In this paper, we would like to investigate this issue through extensive experiments. Besides the simple weighting schema, we explore more options. In this study, we only consider performance, but not dissimilarity among component results for two reasons. Firstly, compared with performance, the effect of dissimilarity on data fusion is smaller (Wu & McClean, 2006b). Secondly, it has been investigated in Wu and McClean (2005, 2006a). Anyway, it should be true that considering results dissimilarity can bring further (though small) improvement.

3. The linear combination method and weights assignment

Suppose we have n information retrieval systems $ir_1, ir_2, \ldots, ir_n$, and for a given query q, each of them provides a result r_i. Each r_i includes a ranked list of documents. The documents are ranked according to their scores. If we have no knowledge of the systems, then an equal weight for every component system is a reasonable policy. If we have some knowledge of those retrieval systems (e.g., evaluating performance by using some training queries), then we are able to take a more profitable policy: good systems are assigned heavy weights and poor systems are assigned light weights. Thus the performance of the fused results can be improved by the linear combination method, which uses the following equation to calculate score:

$$M(d,q) = \sum_{i=1}^{n} w_i * s_i(d,q) \tag{1}$$

Here $s_i(d,q)$[1] is the normalized score of document d in result r_i, w_i is the weight assigned to system ir_i, $M(d,q)$ is the calculated score of d for q. All the documents can be ranked according to their calculated scores.

For each system ir_i, suppose its average performance (e.g., measured by mean average precision (MAP) or recall-level precision (RP)) over a group of training queries is p_i, then p_i is set as ir_i's weight ($w_i = p_i$) in the simple weighting schema, which was used in previous research in data fusion (e.g., Aslam & Montague, 2001; Thompson, 1993; Wu & Crestani, 2002; Wu & McClean, 2006a). In most cases (except in Thompson, 1993) the linear combination method with the simple weighting schema outperformed both CombSum and CombMNZ. However, except the simple weighting schema, no other weighting schemas have been investigated.

Here we assume that the weight that each result possesses is only determined by its performance, or $w_i = Function(p_i)$. Note that the performance of a retrieval system is always in the range of [0,1] when using MAP or RP or other commonly used measures. For the above function, the multiplication function is useless, since both weights are enlarged or dwindled by the same times. We find that using power functions of performance p_i or $w_i = p_i^{power}$ is a very good choice. When different power values are used, we can associate weight with performance in many different ways.

[1] See Eq. (2) for how to normalize raw scores.

Table 1
Information summary of four groups of results submitted to TREC.

TREC group	2001	2003	2004	2005
Track	Web	Robust	Robust	Terabyte
Number of submitted results	97	78	110	58
Number of selected results	32	62	77	41
Number of queries	50	100	249	50
Number of retrieved documents	1000	1000	1000	10,000
Average performance (MAP)	0.1929	0.2307	0.2855	0.3011
Standard deviation of MAP	0.0354	0.0575	0.0420	0.0754

Let us take an example to illustrate this. Suppose we have two systems ir_1 and ir_2, whose performances are 0.6 and 0.8, respectively. If a power of 1 is used, then the weights w_1 and w_2 of ir_1 and ir_2 are $p_1 = 0.6$ and $p_2 = 0.8$, respectively. The following table list the normalized weights of them when a power of 0–5 is used. All the scores have been normalized using the equation $w_i = \frac{w_i}{w_1+w_2}$ to make them more comparable.

power	0	1	2	3	4	5
w_1	0.50	0.43	0.36	0.30	0.24	0.19
w_2	0.50	0.57	0.64	0.70	0.76	0.81

When a negative power is used, then the system in good performance is assigned a light weight, while the system in poor performance is assigned a heavy weight. In such a case, we cannot obtain very effective results by fusing them using the linear combination method. Therefore, negative powers are not appropriate for our purpose and we only consider non-negative powers later in this paper.

When a power of greater than 0 is used, then the system in good performance is assigned a heavy weight, while the system in poor performance is assigned a light weight. Two special cases are 0 and 1. When 0 is used, then the same weight is assigned to all systems. Thus the linear combination method becomes CombSum. When 1 is used, then we obtain the simple weighting schema. If we use a very large power (much greater than 1) for the weighting schema, then the good system has a larger impact, and the poor system has a smaller impact on the fused result. If a large enough power is used, then the good system will be assigned a weight very close to 1, and the poor system will be assigned a weight very close to 0. Thus the fusion process will be dominated by the good component system and the fused result will be very much like the good component result. However, we shall see later in this paper, using a large power like this is not the best weighting policy. In other words, we can obtain some good weights to make the fused results better than the best component result.

It is straightforward to expand the situation in which more than two component systems are included.

4. Empirical investigation of appropriate weights

The purpose of our empirical investigation is to evaluate the simple weighting schema and to try to find more effective schemas if possible. Four groups of TREC data were used for the experiment. Their information is summarized in Table 1 (see Appendix for all the results involved and their performances measured by MAP). From Table 1 we can see that these four groups of submitted results (called runs in TREC) are different in many ways from track (Web, Robust, and Terabyte), the number of results submitted (97, 78, 110, and 58) and selected (32, 62, 77, and 41) in this experiment,[2] the number of queries used (50, 100, and 249), to the number of retrieved documents for each query in each submitted result (1000 and 10,000). They comprise a good combination for us to evaluate data fusion methods.

The zero-one linear normalization method was used for score normalization. It uses the following equation:

$$s(d,q) = \frac{initial(d,q) - min(q)}{max(q) - min(q)} \tag{2}$$

to normalize scores. Here all variables are related to a given query q and a given result r. $max(q)$ and $min(q)$ are the maximum and minimum scores in the result r, respectively; $initial(d,q)$ is the score that document d obtains initially; and $s(d,q)$ is the normalized score that d should obtain. No matter what the initial range is, the normalized scores of any result are always in the range of $[0,1]$.

For all the systems involved, we evaluated their average performance measured by mean average precision (MAP) over a group of queries. Then different values $(0.5, 1.0, 1.5, 2.0, \ldots)$ were used as the power in the power function to calculate weights for the linear combination method. In a year group, we chose m (m = 3, 4, 5, 6, 7, 8, 9, or 10) component results from all available results for fusion. For each setting of m, we randomly chose m component results from all available results for fusion. This process was repeated 200 times. Four metrics were used to evaluate the fused retrieval results. They are mean

[2] Some submitted results include fewer documents than required (1000 or 10,000). For convenience, those results were not used in the experiment.

average precision (MAP), recall-level precision (RP), percentage of the fused results whose performance on MAP is better than the best component result (PMAP), and percentage of the fused results whose performance on RP is better than the best component result (PRP). Besides the linear combination method with different weighting schemas, CombSum and CombMNZ were also involved in the experiment.

Tables 2–5 show the experimental result, in which four different powers (0.5, 1.0, 1.5, and 2.0) are used for the linear combination method. Each data point in the tables is the average of 8*200*q_num measured values. Here 8 is the different number (3,4,...,9,10) of component results used, 200 is the number of runs for a certain number of component results, and q_num is the number of queries in each year group (see Table 1).

All the data fusion methods involved have similar behaviours, though these methods are quite different in performance. For all of them, TREC 2001 is the most successful group, TREC 2004 is the second most successful group, TRECs 2003 and 2005 are the least successful groups.

Among all the data fusion methods, CombMNZ has the worst performance. CombMNZ does well in two year groups TRECs 2001 and 2004, but poorly in two other year groups TRECs 2003 and 2005. In fact, CombMNZ beats the best component result in TREC 2001 (+9.04% in MAP, +5.54% in RP, 79.44% in PMAP, and 73.37% in PRP) and TREC 2004 (+3.46% in MAP, +2.74% in RP, 81.69% in PMAP, and 80.81% in PRP), but worse than the best result in TREC 2003 (−2.41% in MAP, −1.14% in RP, 28.16% in PMAP, and 49.5% in PRP) and TREC 2005 (−4.79% in MAP, −4.51% in RP, 29.62% in PMAP, and 30.56% in PRP).

It is worthwhile to compare CombMNZ with CombSum. CombSum is better than CombMNZ in all four year groups. However, just like CombMNZ, CombSum does not perform as well as the best component result in TRECs 2003 and 2005; while it performs better than the best component result in TRECs 2001 and 2004.

With any of the four powers chosen, the linear combination method performs better than the best component result, CombSum, and CombMNZ in all four year groups. However, the improvement rates of the linear combination method are different from 1 year group to another. Comparing with all different weighting schemas used, we can find that the larger the power is used for weighting calculation, the better the linear combination method performs. The differences are especially bigger for PMAP and PRP. Considering the average of all year groups, the percentage is over 76 for both PMAP and

Table 2
Performance (on MAP) of several data fusion methods (in LC(a), the number a denotes the power used for weight calculation; for every data fusion method, the improvement percentage over the best component result is shown in parentheses).

	2001	2003	2004	2005
Best	0.2367	0.2816	0.3319	0.3823
CombSum	0.2614(+10.44)	0.2796(−0.71)	0.3465(+4.40)	0.3789(−0.89)
CombMNZ	0.2581(+9.04)	0.2748(−2.41)	0.3434(+3.46)	0.3640(−4.79)
LC(0.5)	0.2620(+10.69)	0.2841(+0.89)	0.3482(+4.91)	0.3857(+0.89)
LC(1.0)	0.2637(+11.41)	0.2865(+1.74)	0.3499(+5.42)	0.3897(+1.94)
LC(1.5)	0.2651(+12.00)	0.2879(+2.24)	0.3512(+5.82)	0.3928(+2.75)
LC(2.0)	0.2664(+12.55)	0.2890(+2.63)	0.3522(+6.12)	0.3952(+3.37)

Table 3
Percentage (%) of the fused results whose performance on MAP is better than the best component result (PMAP) (in LC(a), the number a denotes the power used for weight calculation).

	2001	2003	2004	2005	Average
CombSum	83.18	54.62	87.56	50.81	69.05
CombMNZ	79.44	28.16	81.69	29.62	54.73
LC(0.5)	86.75	65.88	90.50	62.87	76.50
LC(1.0)	91.25	71.06	92.69	69.44	81.11
LC(1.5)	94.87	75.25	94.62	75.00	84.94
LC(2.0)	97.44	78.25	95.88	79.88	87.86

Table 4
Performance (on RP) of several data fusion methods (in LC(a), the number a denotes the power used for weight calculation; for every data fusion method, the improvement percentage over the best component result is shown).

	2001	2003	2004	2005
Best	0.2637	0.2977	0.3503	0.4062
CombSum	0.2815(+6.75)	0.2982(+0.17)	0.3629(+3.60)	0.4021(−1.01)
CombMNZ	0.2783(+5.54)	0.2943(−1.14)	0.3599(+2.74)	0.3879(−4.51)
LC(0.5)	0.2821(+6.98)	0.3009(+1.07)	0.3643(+4.00)	0.4077(+0.37)
LC(1.0)	0.2838(+7.62)	0.3024(+1.58)	0.3656(+4.37)	0.4112(+1.23)
LC(1.5)	0.2854(+8.23)	0.3034(+1.91)	0.3667(+4.68)	0.4137(+1.85)
LC(2.0)	0.2865(+8.65)	0.3043(+2.22)	0.3676(+4.94)	0.4156(+2.31)

Table 5
Percentage (%) of the fused results whose performance (on RP) is better than the best component result (PRP) (for LC(a), the number a denotes the power used for weight calculation).

	2001	2003	2004	2005	Average
CombSum	77.06	59.69	86.44	51.88	68.77
CombMNZ	73.37	49.50	80.81	30.56	58.56
LC(0.5)	80.50	68.94	89.31	63.38	76.06
LC(1.0)	86.44	73.06	91.75	69.94	80.30
LC(1.5)	89.81	76.13	93.69	75.25	83.72
LC(2.0)	92.44	79.25	95.37	79.88	86.74

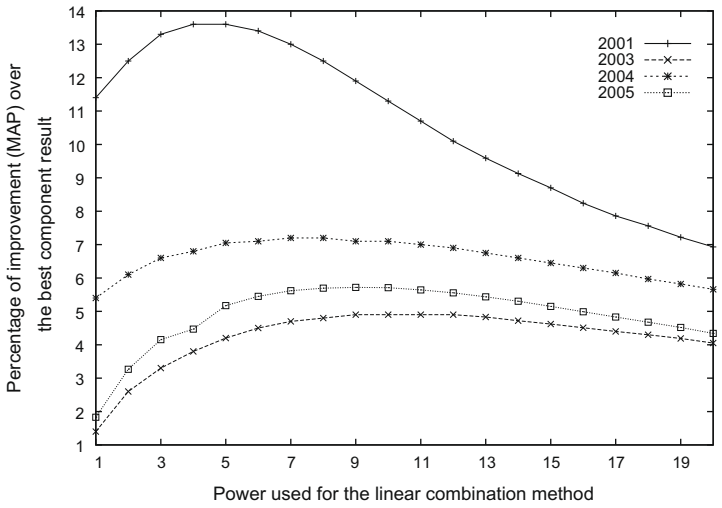

Fig. 1. Percentage of improvement (on MAP) of the linear combination method over the best component result when using different powers.

PRP when a power of 0.5 is used. When the power reaches 2.0, the percentage is above 86 for both PMAP and PRP. This demonstrates that the linear combination method is more reliable than CombSum (69.05% for PMAP, 68.77% for PRP) and CombMNZ (54.73% for PMAP, 58.56% for PRP), when a power of 2 is used for weight calculation.

From the above experimental results we can see that the linear combination method increases in performance with the power used for weight calculation. Since only four different values (0.5, 1, 1.5, 2) have been tested, it is interesting to find how far this trend continues. Therefore, we use more values (3, 4, 5, ..., 20) as power for the linear combination method with the same setting as before. The experimental result is shown in Figs. 1–4.

In Figs. 1 and 2, the curves of TREC 2004 reach their maximum when a power of 4 or 5 is used. While for the three other groups, the curves are quite flat and they reach their maximum when a power of between 7 and 10 is used. It seems that, for obtaining the optimum fusion results, different powers may be needed for different sets of component results. This may seem a little strange, but one explanation for this is: data fusion is affected by many factors such as the number of component results involved, performances and performance differences of component results, dissimilarity among component results, and so on (Wu & McClean, 2006b). Therefore, it is likely that the optimum weight is decided by all these factors, not just by any single factor, though performances of component results is probably the most important one among all the factors. Anyway, if we only consider performance, then a power of 1, as the simple weighting schema does, is far from the optimum.

Two-tailed T tests were carried out to compare the performance of the data fusion methods involved.[3] The test shows that the differences between CombSum and the linear combination method (with any power 1–20) are always statistically significant at a level of 0.000 ($p < 0.0005$, or the probability is over 99.95%). In TREC 2001, the linear combination method is not as good as CombSum when a power of 12 or more is used for MAP (or a power of 11 or more for RP). In all other cases, the linear combination method is better than CombSum. We also compared the linear combination method pairs with adjacent integer powers such as 1 and 2, 2 and 3, etc. In every year group, most of the pairs are different at a level of 0.000 with a few exceptions. In TREC 2001, when comparing the pair with powers 4 and 5, the significance value is 0.745 (MAP). In TREC 2003, the significance value is 0.024 (MAP) for the pair with powers 10 and 11. In TREC 2004, the significance value is 0.037 (RP) for

[3] The linear combination method is regarded to as being multiple methods when different powers are used.

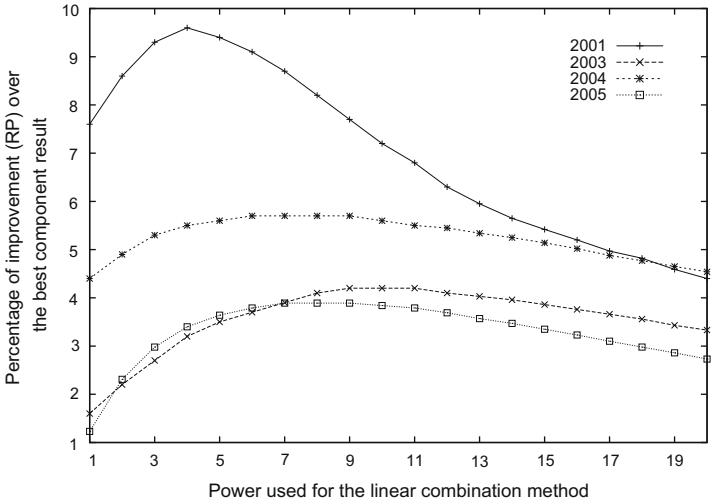

Fig. 2. Percentage of improvement (on RP) of the linear combination method over the best component result when using different powers.

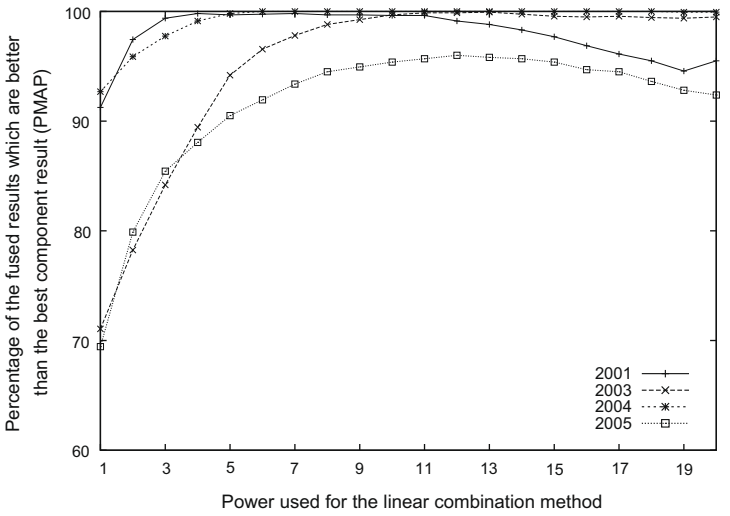

Fig. 3. Percentage of the fused results whose performance in MAP is better than the best component result for the linear combination method using different powers.

the pair with powers 6 and 7; the significance values are 0.690 (MAP) and 0.037 (RP) for the pair with powers 7 and 8. In TREC 2005, the significance value is 0.072 (PP) for the pair with powers 7 and 8; the significance value is 0.035 (RP) for the pair with powers 8 and 9; the significance value is 0.002 (MAP) for the pair with powers 9 and 10. All such exceptions happen when the linear combination method is at the peak of performance.

In Figs. 3 and 4, PMAP and PRP increase very rapidly with the power at the beginning. Both of them reach their maximum almost at the same point as corresponding MAP and RP curves. After that, all the curves decrease gently with power. In two year groups TRECs 2001 and 2004, both PMAP and PRP are around 90 when a power of 1 is used; while in two other year groups TRECs 2003 and 2005, both PMAP and PRP are about 70% when a power of 1 is used.

If we consider all the metrics and all year groups, then the optimum points are not the same. However, we can observe that the performance always increases when the power increases from 1 to 4 for all years groups and all metrics. This suggests that using 2 or 3 or 4 as the power is very likely a better option than using 1, as the simple weighting schema does. Compared with the simple weighting schema, an improvement rate of 1–2% is achieved on MAP and RP, and an improvement rate of 10–15% is achieved on PMAP and PRP by using a power of 4. TREC 2001 reaches its peak when a power of 4 is used. However, the three other groups continue to increase for some time. For example, if we use a power of 7 or 8, and compare it with the simple weighting schema, then an improvement rate of 2.5% on MAP and RP, and an improvement rate of 15–20% on PMAP and PRP are achievable.

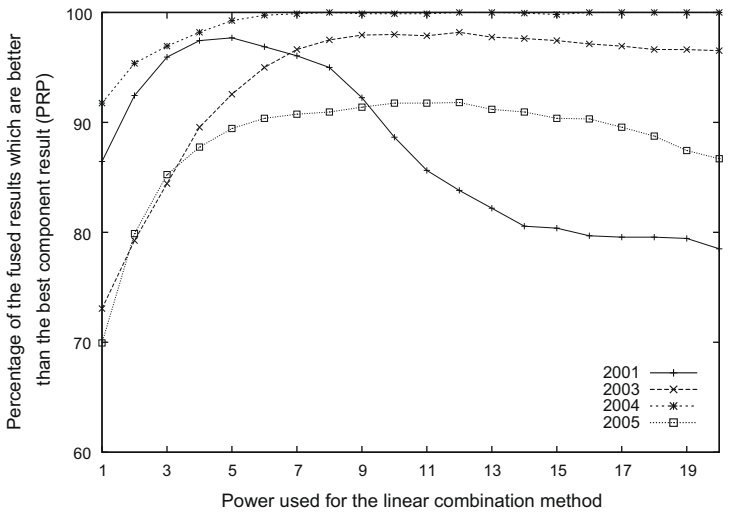

Fig. 4. Percentage of the fused results whose performance in RP is better than the best component result for the linear combination method using different powers.

5. Further observations

In this section we further discuss some related issues about the linear combination data fusion method. The same data as in Section 3 will be used. First let us see the effect of the number of component results on the fused results. For each group of component results chosen randomly, we fused them using the linear combination method 20 times, each with a different weight (a power of 1,2,...,20 was used). Then we chose the best one from all 20 fusion results generated for consideration. The experimental result is presented in Fig. 5. As in Section 3, each data point is the average of 8*200*q_num measured values, where 8 is the number of different groups (including 3,4,...,10 component results), 200 is the number of runs with a specific number of component results, and q_num is the number of queries in each year group. In this section, we only present results measured by MAP. Similar results were observed using RP.

We can see that all four curves almost increase linearly with the number of component results, demonstrating that the number of component results has a positive effect on the performance of the fused result using the linear combination method.

Next let us take a look at the relationship between the average performance of component results and the improvement rate of performance that the fused result can obtain over the average performance. For a group of component results, we chose the best fusion result as above. Then we divided the average performance on MAP into equal intervals

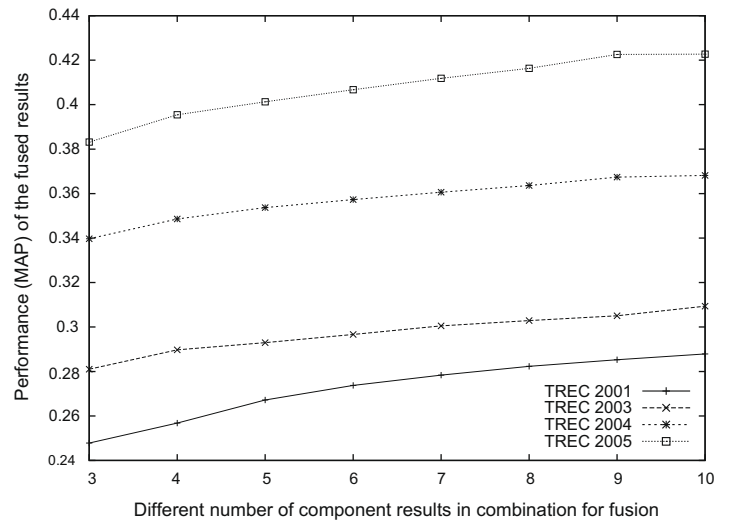

Fig. 5. The relationship between the number of component results and the performance of the fused result on MAP.

0–0.0999, 0.1000–0.1999, . . ., put all the runs into appropriate intervals, and calculated the improvement rate of performance of the fused results for each interval.

Fig. 6 shows the experimental result. Each data point is the average of all the runs in an interval and "0.1" on the horizontal axis denotes the interval of 0.1000–0.1999 and so on. In all year groups, the improvement rate of performance of the fused results decrease rapidly with the average performance of component results, with a few exceptions. In fact, in every year group, all the runs are roughly normally distributed (see Fig. 7). All the exceptions occur in those data points that include very few runs. This demonstrates that the better the component results are in performance, the less benefit we can obtain from data fusion.

When using the linear combination method to fuse results, the contribution of a component result to the fused result is determined by the weight assigned to it. It is interesting to observe the contribution that each component result to the fused result when different weighting schemas are used. In order to achieve this, we carry out the following procedure: first, in a year group, a given number of component results are chosen and fused using different power weighting schemas; second, the best and the worst component results are identified; third, for each weighting schema, we calculate the Euclidean distance between the worst and the best component results, between the worst component results and the fused results, and between the best component results and the fused results after normalising them using the zero-one normalization method.

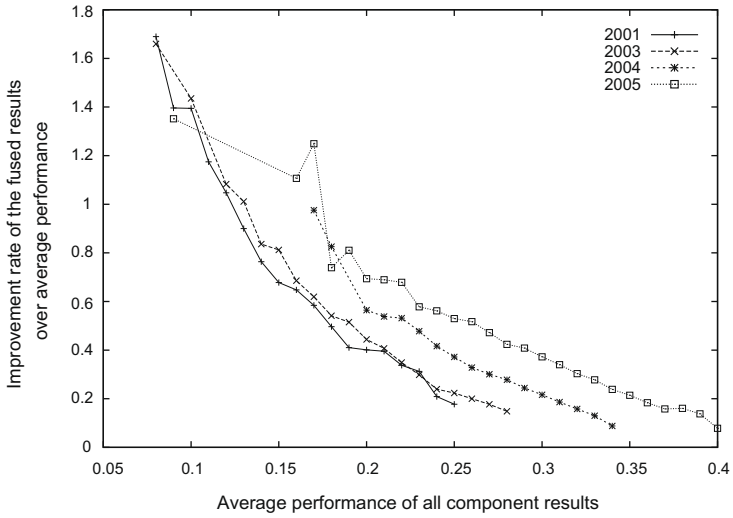

Fig. 6. The relationship between the average performance (on MAP) of component results and the improvement rate of the fused results over average performance of component results.

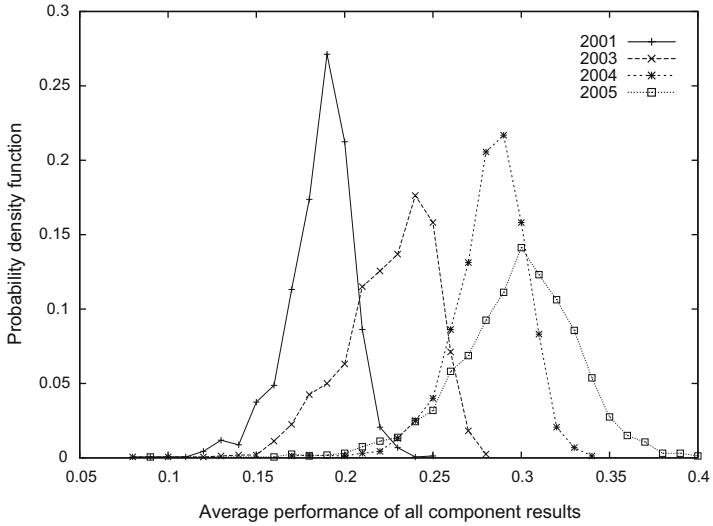

Fig. 7. The distribution of average performance (on MAP) of component results used for data fusion.

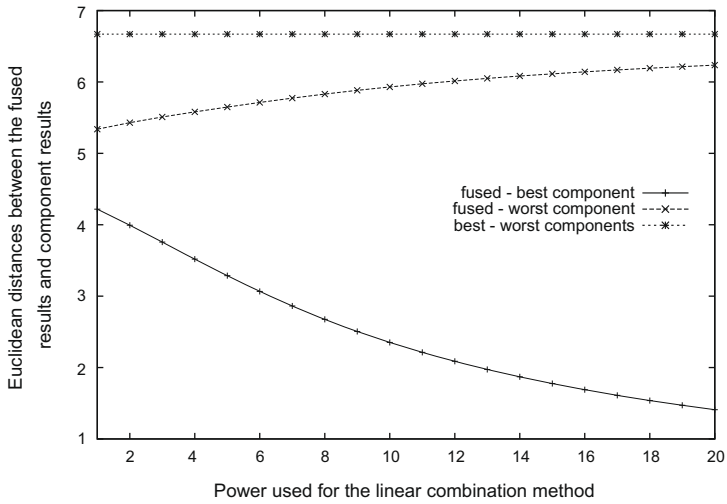

Fig. 8. The Euclidean distances between the best component results, the worst component results and the fused results in TREC 2001.

Using the same data as in Section 3, we carried out the experiment and hereby present the result of TREC 2001 in Fig. 8. The results for the three other year groups are analogous and the figures are omitted here. In Fig. 8, each data point is the average of 8*200*50 fusion runs. Since the Euclidean distance between the worst and best component results is not affected by data fusion at all, it is a flat line in Fig. 8. As expected, when the power increases, the distance between the worst component results and the fused results increases, while the distance between the best component results and the fused results decreases. However, quite surprisingly, even when the power is as high as 20, the distance between the best component results and the fused results is about 1.5 (compared with about 6, the distance between the worst component results and the fused results), which is not close to 0 by any means. This demonstrates that even when a power as high as 20 is used, the fused results are not just affected by the best component results. Other component results (apart from the best ones) still have significant impact on the fused results.

Next we investigate how the power weighting schema can be used with optimisation methods to search suitable weights more effectively and efficiently. For a set of component systems over a group of queries, we would like to find better weights for all of them. It is the case that for every component system, the weights it should take to obtain optimum fusion result are different across queries. However, we would like to find a single weight for each system considering all the training queries as a whole. Such a weight obtained from training queries can be used directly in practice.

Since the optimisation methods are time-consuming, we experimented with 200 fusion runs, each of them including three component results in TREC 2001. As in (Bartell et al., 1994), we used the conjugate gradient method. The conjugate gradient method is an algorithm for finding the nearest local maximum of a function of n variables which presupposes that the gradient of the function can be computed. For each component result involved, we evaluated its average performance (MAP) over all 50 queries. Suppose that the three component results are r_1, r_2, and r_3, their average performances are p_1, p_2, and p_3, respectively. Then we used p_i ($i = 1, 2$, and 3) as r_i's initial weight to begin the search process. In addition, several other options p_i^2, p_i^3, p_i^4, p_i^5, and 0.5 (equal weight for all component results) were also used as initial weights. It seems that there are multiple local maxima in the search space, since different starting weights usually lead to different maxima (local maxima). The conjugate gradient method, like many other optimisation methods, is always able to find a local maximum, but not always a global maximum. Table 6 shows the experimental result.

We should note that the starting point can affect both effectiveness and efficiency of optimisation methods. Firstly, if the starting point chosen is very close to a local or global maximum, then it takes only a few steps for the optimisation method to approach it. Otherwise, it may take much longer time. Thus the efficiency of the optimisation method can be affected. Sec-

Table 6
Performance (on MAP) of the fused result using the linear combination method, in which suitable weights are obtained by the conjugate gradient method (TREC 2001, three component results, 200 runs).

Starting weights	MAP (starting weights)	MAP (searched weights)	Improvement of (3) over (2) (%)	Time (s)
0.5	0.23931	0.23945	+0.06	42.8
p	0.23976	0.23984	+0.03	30.5
p^2	0.24176	0.24193*	+0.07	47.1
p^3	0.24252	0.24402*	+0.62	88.3
p^4	0.24280	0.24316*	+0.15	58.4
p^5	0.24200	0.24200	+0.0	16.5

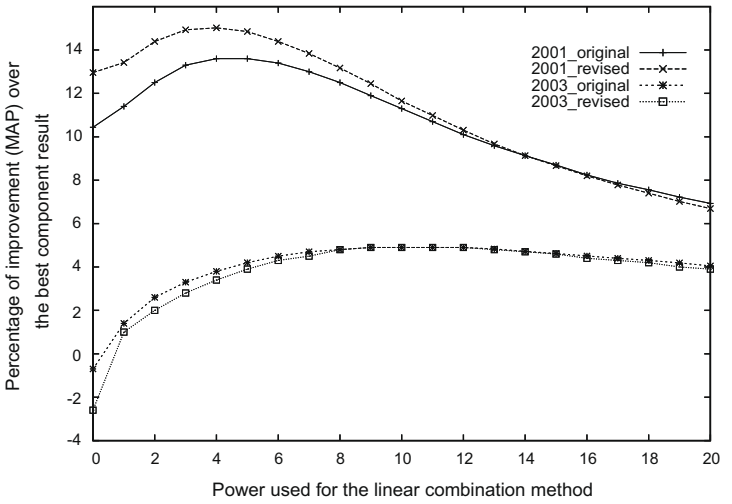

Fig. 9. Comparison of fusion performance of two groups, the non-same-participant restriction is applied to one group and no such restriction to the other group for comparison, the measure used is the percentage of improvement (on MAP) of the linear combination method over the best component result.

ondly, different starting points lead to different (local or global) maxima. It is very likely that for a starting point, the closest local or global maximum will be found. Therefore, the effectiveness of the optimisation method can be affected by the starting point. In Table 6, p^3 leads to the best results among all the options. Therefore, a proper power weighting schema can be used as a good starting point for the optimisation method to search more favourable weights. Compared the figures in column 3 with the figures in column 2, some of them, with a "$*$" mark, are statistically significant (p-value $\leqslant 0.05$).

The time required for each run of the conjugate gradient method is given in Table 6 as well. We used a reasonably good performance PC (Intel Dual_core CPUs and 1 GM of RAM) for this. For each group of three component results over 50 queries, the time required varies from 16 s to 1.5 min. However, in the case of 16 s for p^5, no improvement has been made. Since the optimisation method cannot find any better points in the neighbourhood and stops the search process after only one or two steps. The more time the search process takes, the more improvement on fusion we can obtain (for the searched point to be compared with the start point). More time is needed if more component results or more queries are involved. Especially when more component results are involved, the time complexity of the search process increases very rapidly. On the other hand, it takes less than 2 s to calculate the weights of three component results using the power function schema. Most of the time required is to calculate the performances of all component results. When more results or more queries are used, the time required increases linearly.

One phenomenon in TREC may be worth investigating in this study. In each year group, any participant may submit more than one run to the same track. Those runs were more similar than usual since many of them were just obtained by using the same information retrieval system but different parameter settings/different query formats. In our previous experiments, We do not distinguish runs from the same participant or not. In each combination, a few component results may come from the same participants, then the fused results may be biased to them. In order to avoiding such things from happening, we divide submitted runs into groups and all the runs submitted by the same participant are put in the same group. We randomly select runs from all the groups as before. But for any single combination, we choose at most one run from any particular group. We refer this to be the non-same-participant restriction later in this paper. We used TRECs 2001 and 2003 in this experiment. As before, in a year group, we chose m (m = 3, 4, 5, 6, 7, 8, 9, or 10) component results from all available results for fusion with the non-same-participant restriction. For each setting of m, we formed and tested 200 combinations. Fig. 9 shows the experimental result.

In Fig. 9, "2001 revised" and "2003 revised" denote the groups that observe the non-same-participant restriction, while "2001 original" and "2003 original" denote the groups which do not observe the non-same-participant restriction.[4] The curves of "2001 original" and "2001 revised" are not always very close, this is because different component results were involved in two corresponding groups, though it is the same case for "2003 original" and "2003 revised". We can see that the two corresponding curves for comparison are the same in shape and they reach the maximum points with more or less the same power values. Therefore, it demonstrates that the even dissimilarity among component results does not affect our conclusion before.

In all above experiments, the same group of queries was used for obtaining performance weights and evaluation of data fusion methods. Now let us investigate a more practical scenario: different groups of queries were used for obtaining performance weights and evaluation of data fusion methods. In order to do this, we divide all the queries in every year group

[4] They were the curves labelled as "2001" and "2003" in Fig. 1.

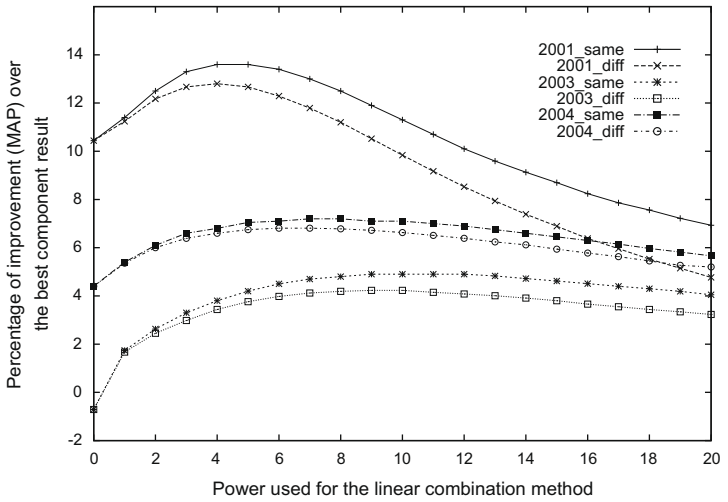

Fig. 10. Comparison of fusion performance in two different settings, one setting uses the same group of queries for obtaining performance weights and evaluation of data fusion methods and the other setting uses different group of queries for obtaining performance weights and evaluation of data fusion methods, the measure used is the percentage of improvement (on MAP) of the linear combination method over the best component result.

Table 7
Correlations of performances between two sub-groups of queries in each year group.

Groups	Pearson	Kendall's tau_b	Spearman
TREC 2001	0.856	0.532	0.685
TREC 2003	0.961	0.720	0.880
TREC 2004	0.957	0.798	0.938

into two sub-groups: odd-numbered queries and even-numbered queries. We obtain two different performance weights from these two sub-groups. Then for those odd-numbered queries, we evaluate all data fusion methods using the weights obtained from even-numbered queries; and for those even-numbered queries, we evaluate all data fusion methods using the weights obtained from odd-numbered queries. Three year groups (TREC 2001, 2003, and 2004) were used in this experiment. All the combinations involved are the same as that in the experiment in Section 4. Thus we can carry out a fair comparison of these two different settings.

Fig. 10 shows the result. "2001_diff", "2003_diff", and "2004_diff" are from the setting that uses different group of queries for performance weights training and fusion methods evaluation; while "2001_same", "2003_same", and "2004_same" are from the setting that uses the same group of queries for performance weights training and fusion methods evaluation.[5]

From Fig. 10, we can see that using the same group of queries can always lead to better fusion performance than using different groups of queries for performance weights training and data fusion methods evaluation. The difference between the two corresponding curves becomes larger when a larger power value is used for weight assignment. However, the two corresponding curves in each year group bear the same shape. Therefore, our observations and conclusions obtained before by using the same group of queries for performance weights training and data fusion methods evaluation are also held for the situation that different queries are used for performance weights training and data fusion methods evaluation.

In Fig. 10, we can also find that, in each year group, the difference between the two curves are different. Among them, TREC 2001 has the biggest difference, TREC 2004 has the smallest difference, while TREC 2003 is in the middle. This is due to the different accuracies of performance estimation for these three groups of component results. Table 7 shows the correlation of performances between the two sub-groups of queries in each year group. Among these three groups, the correlation between the two sub-groups in TREC 2004 is the strongest, which is followed by the two sub-groups in TREC 2003, and the correlation between the two sub-groups in TREC 2001 is the weakest. Further, we believe this is mainly because of the different numbers of queries used in each year group. In TREC 2001, there are only 50 queries, and each sub-group includes 25 queries; in TREC 2003, there are 100 queries, and each sub-group includes 50 queries; and in TREC 2004, there are 249 queries, and each sub-group includes 125 or 124 queries. This experiment also suggests that a relatively large number of queries (at least 50) should be used for an accurate estimation of performance.

[5] They were the curves labelled as "2001", "2003", and "2004" in Fig. 1.

6. Conclusive remarks

In this paper we have presented our work about how to assign appropriate weights for the linear combination data fusion method. From the extensive experiments conducted with the TREC data, we conclude that a series of power functions (powers between 2 and 8) are better than the simple weighting schema, in which a system is assigned a weight equal to its average performance. The power function schema can be implemented as efficiently as the simple weighting schema.

We have also investigated some related issues about the linear combination method. First, we have found that the number of component results has a positive effect on the performance of the fused result, while the average performance of component results has a negative effect on the performance of the fused result. Second, we have investigated the similarity between the fused results and component results when different powers are used in the linear combination method. A very informative observation is: even when a power as big as 20 is used, the difference between the fused results and the best component results is still very significant. In other words, other results than the best component results still have very significant impact on the fused results. Finally, we have demonstrated that the observation from this study is also very useful for the optimisation methods to achieve more effective weights more efficiently. Since the whole space is too large, any optimisation method would only be possible to search in a very small area to find a local maximum. The observations from this study can tell the optimisation methods where is the right area for a search. In summary, we believe that our empirical investigations and observations presented in this paper are very useful for us and other researchers to have a better understanding of the linear combination data fusion method and the data fusion technique in general.

Acknowledgements

The authors thank the anonymous reviewers for their helpful suggestions and comments which have been taken to improve the quality and presentation of this paper. Xiaoqin Zeng and Lixin Han's work is partially supported by the National Natural Science Foundation of China under grant numbers 60673186 and 60571048.

Appendix

In this appendix, all the systems used in the experiment are listed with their average performances (on MAP).

TREC 2001 (32 in total)

apl10wc(0.1567), apl10wd(0.2035), flabxt(0.1719), flabxtd(0.2332), flabxtdn(0.1843), flabxtl(0.1705), fub01be2(0.2225), fub01idf(0.1900), fub01ne(0.1790), fub01ne2(0.1962), hum01tdlx(0.2201), iit01tde(0.1791), jscbtawtl1(0.1890), jscbtawtl2(0.1954), jscbtawtl3(0.2003), jscbtawtl4(0.2060), kuadhoc2001(0.2088), Merxtd(0.1729), msrcn2(0.1863), msrcn3(0.1779), msrcn4(0.1878), ok10wtnd0(0.2512), ok10wtnd1(0.2831), pir1Wa(0.1715), posnir01ptd(0.1877), ricAP(0.2077), ricMM(0.2096), ricMS(0.2068), ricST(0.1933), uncfslm(0.0780), uncvsmm(0.1269), UniNEn7d(0.2242).

TREC 2003 (62 in total)

aplrob03a(0.2998), aplrob03b(0.2522), aplrob03c(0.2521), aplrob03d(0.2726), aplrob03e(0.2535), fub03IeOLKe3(0.2503), fub03InB2e3(0.2435), fub03IneOBu3(0.2329), fub03IneOLe3(0.2479), fub03InOLe3(0.2519), humR03d(0.2367), humR03de(0.2627), InexpC2(0.2249), InexpC2QE(0.2384), MU03rob01(0.1926), MU03rob02(0.2187), MU03rob04(0.2147), MU03rob05(0.2029), oce03noXbm(0.2292), oce03noXbmD(0.1986), oce03noXpr(0.1853), oce03Xbm(0.2446), oce03Xpr(0.1836), pircRBa1(0.3100),pircRBa2(0.3111), pircRBd1(0.2774), pircRBd2(0.2900), pircRBd3(0.2816), rutcor030(0.0482), rutcor03100(0.0582), rutcor0325(0.0590), rutcor0350(0.0683), rutcor0375(0.0767), SABIR03BASE(0.2021), SABIR03BF(0.2263), SABIR03MERGE(0.2254), Sel50(0.2190), Sel50QE(0.2387), Sel78QE(0.2432), THUIRr0301(0.2597), THUIRr0302(0.2666), THUIRr0303(0.2571), THUIRr0305(0.2434), UAmsT03R(0.2324), UAmsT03RDesc(0.2065), UAmsT03RFb(0.2452), UAmsT03RSt(0.2450), UAmsT03RStFb(0.2373), UIUC03Rd1(0.2424), UIUC03Rd2(0.2408), UIUC03Rd3(0.2502), UIUC03Rt1(0.2052), UIUC03Rtd1(0.2660), uwmtCR0(0.2763), uwmtCR1(0.2344), uwmtCR2(0.2692), uwmtCR4(0.2737), VTcdhgp1(0.2649), VTcdhgp3(0.2637), VTDokrcgp5(0.2563), VTgpdhgp2(0.2731), VTgpdhgp4(0.2696).

TREC 2004 (77 in total)

apl04rsTDNfw(0.3172), apl04rsTDNw5(0.2828), fub04De(0.3062), fub04Dg(0.3088), fub04Dge(0.3237), fub04T2ge(0.2954), fub04TDNe(0.3391), fub04TDNg(0.3262), fub04TDNge(0.3405), fub04Te(0.2968), fub04Tg(0.2987), fub04Tge(0.3089), humR04d4e5(0.2756), humR04t5e1(0.2768), icl04pos2d(0.1746), icl04pos2f(0.2160), icl04pos2td(0.1888), icl04pos48f(0.1825), icl04pos7f(0.2059), icl04pos7td(0.1783), JuruDes(0.2678), JuruDesAggr(0.2628), JuruDesLaMd(0.2686), JuruDesQE(0.2719), JuruDesSwQE(0.2714), JuruDesTrSl(0.2658), JuruTitDes(0.2803), NLPR04clus10(0.3059), NLPR04clus9(0.2915), NLPR04COMB(0.2819), NLPR04LcA(0.2829), NLPR04LMts(0.2438), NLPR04NcA(0.2829), NLPR04okall(0.2778), NLPR04OKapi(0.2617), NLPR04okdiv(0.2729), NLPR04oktwo(0.2808), NLPR04SemLM(0.2761), pircRB04d2(0.3134), pircRB04d3(0.3319), pircRB04d4(0.3338), pircRB04d5(0.3319), pircRB04t2(0.2984), pircRB04t3(0.3331), pircRB04t4(0.3304), pircRB04td2(0.3586), pircRB04td3(0.3575), polyudp2(0.1948), polyudp4(0.1945), polyudp5(0.2455), polyudp6(0.2383), SABIR04BA(0.2944), SABIR04BD(0.2627), SABIR04BT(0.2533),

SABIR04FA(0.2840), SABIR04FD(0.2609), SABIR04FT(0.2508), uogRobDBase(0.2959), uogRobDWR10(0.3033), uogRobDWR5(0.3021), uogRobLBase(0.3056), uogRobLT(0.3128), uogRobLWR10(0.3201), uogRobLWR5(0.3161), uogRobSBase(0.2955), uogRobSWR10(0.3011), uogRobSWR5(0.2982), vtumdesc(0.2945), vtumlong252(0.3245), vtumlong254(0.3252), vtumlong344(0.3275), vtumlong348(0.3275), vtumlong432(0.3275), vtumlong436(0.3280), vtumtitle(0.2822), wdo25qla1(0.2458), wdoqla1(0.2914).

TREC 2005 (41 in total)

DCU05ABM25(0.2887), DCU05ACOMBO(0.2916), DCU05ADID(0.2886), DCU05AWTF(0.3021), humT05l(0.3154), humT05x5l(0.3322), humT05xl(0.3360), humT05xle(0.3655), indri05Adm(0.3505), indri05AdmfL(0.4041), indri05AdmfS(0.3886), indri05Aql(0.3252), JuruDF0(0.2843), JuruDF1(0.2855), juruFeRa(0.2752), juruFeSe(0.2692), MU05TBa1(0.3199), MU05TBa2(0.3218), MU05TBa3(0.3063), MU05TBa4(0.3092), NTUAH1(0.3023), NTUAH2(0.3233), NTUAH3(0.2425), NTUAH4(0.2364), QUT05DBEn(0.1645), QUT05TBEn(0.1894), QUT05TBMRel(0.1837), QUT05TSynEn(0.0881), sab05tball(0.2087), sab05tbas(0.2088), UAmsT05aTeLM(0.1685), UAmsT05aTeVS(0.1996), uogTB05LQEV(0.3650), uogTB05SQE(0.3755), uogTB05SQEH(0.3548), uogTB05SQES(0.3687), uwmtEwtaD00t(0.3173), uwmtEwtaD02t(0.2173), uwmtEwtaPt(0.3451), uwmtEwtaPtdn(0.3480), york05tAa1(0.1565).

References

Aslam, J. A., & Montague, M. (2001). Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference* (pp. 276–284).

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. ACM Press and Addison-Wesley.

Bartell, B. T., Cottrell, G. W., & Belew, R. K. (1994). Automatic combination of multiple ranked retrieval systems. In *Proceedings of ACM SIGIR'94* (pp. 173–184).

Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001). Rank aggregation methods for the web. In *Proceedings of the 10th international world wide web conference* (pp. 613–622).

Farah, M., & Vanderpooten, D. (2007). An outranking approach for rank aggregation in information retrieval. In *Proceedings of the 30th ACM SIGIR conference* (pp. 591–598).

Fox, E. A., Koushik, M. P., Shaw, J., Modlin, R., & Rao, D. (1993). Combining evidence from multiple searches. In *Proceedings of the first Text REtrieval conference (TREC-1)* (pp. 319–328).

Fox, E. A., & Shaw, J. (1994). Combination of multiple searches. In *Proceedings of the second Text REtrieval conference (TREC-2)* (pp. 243–252).

Lee, J. H. (1997). Analysis of multiple evidence combination. In *Proceedings of the 20th annual international ACM SIGIR conference* (pp. 267–275).

Lillis, D., Toolan, F., Collier, R., & Dunnion, J. (2006). Probfuse: a probabilistic approach to data fusion. In *Proceedings of the 29th annual international ACM SIGIR conference* (pp. 139–146).

Manmatha, R., Rath, T., & Feng, F. (2001). Modelling score distributions for combining the outputs of search engines. In *Proceedings of the 24th annual international ACM SIGIR conference* (pp. 267–275).

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

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

Nottelmann, H., & Fuhr, N. (2003). From retrieval status values to probabilities of relevance for advanced IR applications. *Information Retrieval, 6*(3–4), 363–388.

Nuray, R., & Can, F. (2006). Automatic ranking of information retrieval systems using fusion data. *Information Processing and Management, 42*(3), 595–614.

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

Soboroff, I., Nicholas, C., & Cahan, P. (2001). Ranking retrieval systems without relevance judgments. In *Proceedings of 24th annual international ACM SIGIR conference* (pp. 66–73).

Thompson, P. (1993). Description of the PRC CEO algorithms for TREC. In *Proceedings of the first Text REtrieval conference (TREC-1)* (pp. 337–342).

Vogt, C. C., & Cottrell, G. W. (1998). Predicting the performance of linearly combined IR systems. In *Proceedings of the 21st annual ACM SIGIR conference* (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 2002 ACM CIKM international conference on information and knowledge management* (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 (SAC)* (pp. 811–816).

Wu, S., Crestani, F., & Bi, Y. (2006). Evaluating score normalization methods in data fusion. In *Proceedings of the third Asia information retrieval symposium (LNCS 4182)* (pp. 642–648).

Wu, S., & McClean, S. (2005). Data fusion with correlation weights. In *Proceedings of the 27th European conference on information retrieval* (pp. 275–286).

Wu, S., & McClean, S. (2006a). Improving high accuracy retrieval by eliminating the uneven correlation effect in data fusion. *Journal of American Society for Information Science and Technology, 57*(14), 1962–1973.

Wu, S., & McClean, S. (2006b). Performance prediction of data fusion for information retrieval. *Information Processing and Management, 42*(4), 899–915.

Wu, S., & McClean, S. (2007). Result merging methods in distributed information retrieval with overlapping databases. *Information Retrieval, 10*(3), 297–319.

Wu, S. (2009). Applying statistical principles to data fusion in information retrieval. *Expert Systems with Applications, 36*(2), 2997–3006.