Performance Weights for the Linear Combination Data Fusion Method in Information Retrieval

Shengli Wu[1], Qili Zhou[2], Yaxin Bi[1], and Xiaoqin Zeng[3]

[1] School of Computing and Mathematics
University of Ulster, Northern Ireland, UK, BT37 0QB
{s.wu1,y.bi}@ulster.ac.uk
[2] School of Computing
Hangzhou Dianzi University, Hangzhou, China, 310018
cotzq@abertay.ac.uk
[3] Department of Computer Science
Hohai University, Nanjing, China, 210098
xzeng@hhu.edu.cn

Abstract. In information retrieval, the linear combination method is a very flexible and effective data fusion method, since different weights can be assigned to different component systems. However, it remains an open question which weighting schema is good. Previously, a simple weighting schema was very often used: for a system, its weight is assigned as its average performance over a group of training queries. In this paper, we investigate the weighting issue by extensive experiments. We find 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 data fusion.

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 various text document collections. For example, in the book "Modern Information Retrieval" written by Baeza-Yates and Ribeiro-Neto [2], 11 different retrieval models were discussed. 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.

Quite a few data fusion methods such as CombSum [4,5], CombMNZ [4,5], the linear combination method [3,9,10], the probabilistic fusion method [6], Borda fusion [1], Condorcet fusion [7], and the correlation method [12,13], have been proposed. Among them, the linear combination data fusion method is a very flexible and effective method since different weights can be assigned to different

A. An et al. (Eds.): ISMIS 2008, LNAI 4994, pp. 465–470, 2008.

systems. However, it is unclear which weighting schema is good. In some previous researches, different search methods such as golden section search [9,10] and conjugate gradient [3] were used to search suitable weights for component systems. One major drawback of these methods is their very low efficiency. Because of this, data fusion with only two or three component systems were investigated in [3] and [9,10]. In some situations such as the WWW and digital libraries, documents are updated frequently, then each component system's performance may change considerably from time to time. The weights for the systems should be updated accordingly. In such a situation, it is very difficult or impossible to use those low efficient weighting methods.

In some data fusion experiments, (e.g., in [1,8,11,13]), a simple weighting schema was used: for a system, its weight is set as its average performance over a group of training queries. There is a straightforward relationship between performance and weight. This method can be used in very dynamic situations since weights can be calculated and modified very easily. However, it has not been investigated how good this schema is. We would like to investigate this issue with extensive experiments. We shall demonstrate that, a power function weighting schema, with a power of between 2 and 6, is more effective than the simple weighting schema (which is a special case of power function, power = 1) for data fusion, though both of them can be implemented in the same way.

2 Performance Weights

Suppose we have n information retrieval systems ir_1, ir_2,..., ir_n. For a given query q, each of them provides a result r_i. Each r_i is a ranked list of documents, with an estimated relevance score for every document included. w_i is the performance weight assigned to system ir_i. Then for any document d in one or more results, the linear combination method uses the following equation to calculate its score:

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

Here $s_i(d,q)$ is the normalized score of document d in result r_i, $M(d,q)$ is the calculated score of d. All the documents can be ranked using their calculated scores $M(d,q)$.

For each system ir_i, suppose its average performance over a group of training queries is p_i, then p_i is set as ir_i's weight (w_i) in the simple weighting schema, which has been used in previous research (e.g., in [1,8,11,13]). However, it is not clear how good this simple weighting schema is or is there any other effective schemas available. The purpose of our investigation is to try to find some other schemas which are more effective than the simple weighting schema but can be implemented as efficiently as the simple weighting schema. In order to achieve this, we set pw_i as a power function of p_i. Besides p_i, we used ${p_i}^{0.5}$, ${p_i}^{1.5}$, ${p_i}^{2}$, ${p_i}^{2.5}$ and ${p_i}^{3}$ as ir_i's weights. Note if a larger power is used for the weighting schema, then those systems with better performance have a larger impact on fusion, and those results with poorer performance have a smaller impact on fusion.

3 Experimental Results

4 groups of TREC data (2001 Web, 2003 and 2004 Robust, and 2005 Terabyte) were used for the experiment. These 4 groups of submitted results (called runs in TREC) are different in many ways from track (Web, Robust, and Terabyte), the number of results selected (32(2001), 62(2003), 77(2004), and 41(2005)), the number of queries used (50(2001 and 2005), 100(2003), and 249(2004)), to the number of retrieved documents for each query in each submitted result (1000(2001, 2003, and 2004) and 10000(2005))[1]. They comprise a good combination for us to evaluate data fusion methods.

The Zero-one linear normalization method was used for score normalization. It maps the highest score into 1, the lowest score into 0, and any other scores into a value between 0 and 1. For all the systems involved, we evaluated their average performance measured by MAP (mean average precision) over a group of queries. Then different values (0.5, 1.0, 1.5, 2.0,...) were used as powers 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 for fusion. For each setting of m, we randomly chose m component results 200 times and carried out fusion. Two metrics were used to evaluate the fused retrieval results. They are mean average precision (MAP) and recall-level precision(RP). Besides the linear combination method with different weighting schemas, CombSum and CombMNZ were also involved in the experiment.

Tables 1-2 show the performance of the fused result in MAP and RP, respectively. 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 each setting, and q_num is the number of queries in each year group. The improvement rate over the best component result is shown as well.

From Tables 1 and 2 we can see that the two measures MAP and RP are very consistent, though the RP values are usually smaller than the corresponding MAP values. Comparing CombMNZ with CombSum, CombMNZ is not as good as CombSum in all 4 year groups. With any of the weighting schemas chosen, the linear combination method performs better than the best component result, CombSum, and CombMNZ in all 4 year groups. 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. Two-tailed tests were carried out to compare the differences between all the data fusion methods involved. The tests show that the differences between any pair of the data fusion methods are statistically significant at a level of .000 ($p < 0.001$, or the probability is over 99.9%). From the worst to the best, the data fusion methods are ranked as follows: CombMNZ, CombSum, LC(0.5), LC(1.0), LC(1.5), LC(2.0), LC(2.5), LC(3.0).

[1] Some submitted results include fewer documents. For convenience, those results were not selected.

Table 1. Performance (on MAP) of several data fusion methods (In LC(a), the number a denotes the power value used for weight calculation; for every data fusion method, the improvement rate of its MAP value over the best component result is shown)

Group/ Best	Comb-Sum	Comb-MNZ	LC(0.5)	LC(1.0)	LC(1.5)	LC(2.0)	LC(2.5)	LC(3.0)
2001	0.2614	0.2581	0.2620	0.2637	0.2651	0.2664	0.2673	0.2681
0.1861	+10.44%	+9.04%	+10.69%	+11.41%	+12.00%	+12.55%	+12.93%	+13.27%
2003	0.2796	0.2748	0.2841	0.2865	0.2879	0.2890	0.2900	0.2908
0.2256	-0.71%	-2.41%	+0.89%	+1.74%	+2.24%	+2.63%	+2.98%	+3.27%
2004	0.3465	0.3434	0.3482	0.3499	0.3512	0.3522	0.3530	0.3537
0.2824	+4.40%	+3.46%	+4.91%	+5.42%	+5.82%	+6.12%	+6.36%	+6.57%
2005	0.3789	0.3640	0.3857	0.3897	0.3928	0.3952	0.3970	0.3986
0.2991	-0.89%	-4.79%	+0.89%	+1.94%	+2.75%	+3.37%	+3.85%	+4.26%

Table 2. Performance (on RP) of several data fusion methods (In LC(a), the number a denotes the power value used for weight calculation; for every data fusion method, the improvement rate of its RP value over the best component results is shown)

Group	Comb-Sum	Comb-MNZ	LC(0.5)	LC(1.0)	LC(1.5)	LC(2.0)	LC(2.5)	LC(3.0)
2001	0.2815	0.2783	0.2821	0.2838	0.2854	0.2865	0.2874	0.2882
0.2174	+6.75%	+5.54%	+6.98%	+7.62%	+8.23%	+8.65%	+8.99%	+9.29%
2003	0.2982	0.2943	0.3009	0.3024	0.3034	0.3043	0.3051	0.3058
0.2508	+0.17%	-1.14%	+1.07%	+1.58%	+1.91%	+2.22%	+2.49%	+2.72%
2004	0.3629	0.3599	0.3643	0.3656	0.3667	0.3676	0.3682	0.3687
0.3107	+3.60%	+2.74%	+4.00%	+4.37%	+4.68%	+4.94%	+5.11%	+5.25%
2005	0.4021	0.3879	0.4077	0.4112	0.4137	0.4156	0.4171	0.4183
0.3357	-1.01%	-4.51%	+0.37%	+1.23%	+1.85%	2.31%	+2.68%	+2.98%

Table 3. Percentage of the fused results whose performances (MAP) are better than the best component result

Group	CombSum	CombMNZ	LC(0.5)	LC(1.0)	LC(1.5)	LC(2.0)	LC(2.5)	LC(3.0)
2001	83.18%	79.44%	86.75%	91.25%	94.87%	97.44%	98.69%	99.38%
2003	54.62%	28.16%	65.88%	71.06%	75.25%	78.25%	81.00%	84.19%
2004	87.56%	81.69%	90.50%	92.69%	94.62%	95.88%	96.88%	97.75%
2005	50.81%	29.62%	62.87%	69.44%	75.00%	79.88%	83.00%	85.44%

For MAP, we also calculated the percentage that the fused results is more effective than the best component result, which is shown in Table 3. The figures for RP are very similar, therefore we do not present them here. From Table 3, we can see that the linear combination methods are better than CombSum and CombMNZ in all year groups.

From the above experimental results we can see that the linear combination method increases in performance with the power used for weight calculation.

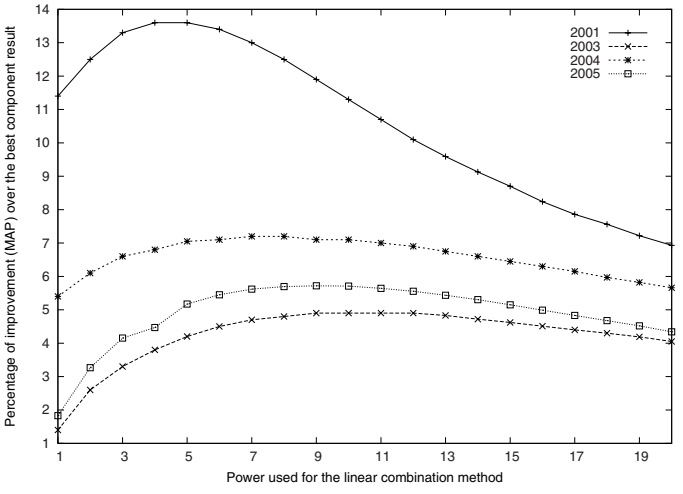

Fig. 1. Percentage of improvement (on MAP) of the linear combination method when using different powers

Since only six different values (0.5, 1, 1.5, 2, 2,5, 3) have been tested, it is interesting to find how far this trend continues. Therefore, we use more values (4, 5,... , 20) as powers for the linear combination method with the same setting as before. The experimental result is shown in Figure 1.

In Figure 1, the curve of TREC 2004 reach its 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 [14]. 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.

4 Conclusions

In this paper we have presented our work about assigning appropriate performance weights for the linear combination data fusion method. From the extensive experiments conducted with the TREC data, we conclude that for performance weighting, a series of power functions (e.g., a power of 2 to 6) are better than the simple weighting schema, in which the performance weight of a system is assigned as its average performance (power equals to 1). The power function

schema can be implemented as efficiently as the simple weighting schema. We expect that the finding in this paper is very useful in practice. As our next stage of work, we plan to carry out some theoretical analysis to see why this is the case.

References

1. Aslam, J.A., Montague, M.: Models for metasearch. In: Proceedings of the 24th Annual International ACM SIGIR Conference, New Orleans, Louisiana, USA, September 2001, pp. 276–284 (2001)
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press, Addison-Wesley (1999)
3. Bartell, B.T., Cottrell, G.W., Belew, R.K.: Automatic combination of multiple ranked retrieval systems. In: Proceedings of ACM SIGIR 1994, Dublin, Ireland, July 1994, pp. 173–184 (1994)
4. Fox, E.A., Koushik, M.P., Shaw, J., Modlin, R., Rao, D.: Combining evidence from multiple searches. In: The First Text REtrieval Conference (TREC-1), Gaitherburg, MD, USA, March 1993, pp. 319–328 (1993)
5. Fox, E.A., Shaw, J.: Combination of multiple searches. In: The Second Text REtrieval Conference (TREC-2), Gaitherburg, MD, USA, August 1994, pp. 243–252 (1994)
6. Lillis, D., Toolan, F., Collier, R., Dunnion, J.: Probfuse: a probabilistic approach to data fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference, Seattle, Washington, USA, August 2006, pp. 139–146 (2006)
7. Montague, M., Aslam, J.A.: Condorcet fusion for improved retrieval. In: Proceedings of ACM CIKM Conference, McLean, VA, USA, November 2002, pp. 538–548 (2002)
8. Thompson, P.: Description of the PRC CEO algorithms for TREC. In: The First Text REtrieval Conference (TREC-1), Gaitherburg, MD, USA, March 1993, pp. 337–342 (1993)
9. Vogt, C.C., Cottrell, G.W.: Predicting the performance of linearly combined IR systems. In: Proceedings of the 21st Annual ACM SIGIR Conference, Melbourne, Australia, August 1998, pp. 190–196 (1998)
10. Vogt, C.C., Cottrell, G.W.: Fusion via a linear combination of scores. Information Retrieval 1(3), 151–173 (1999)
11. Wu, S., Crestani, F.: Data fusion with estimated weights. In: Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, November 2002, pp. 648–651 (2002)
12. Wu, S., McClean, S.: Data fusion with correlation weights. In: Proceedings of the 27th European Conference on Information Retrieval, Santiago de Composite, Spain, March 2005, pp. 275–286 (2005)
13. Wu, S., McClean, S.: 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 (2006)
14. Wu, S., McClean, S.: Performance prediction of data fusion for information retrieval. Information Processing & Management 42(4), 899–915 (2006)