Evaluating Score Normalization Methods in Data Fusion

Shengli Wu[1], Fabio Crestani[2], and Yaxin Bi[1]

[1] School of Computing and Mathematics
University of Ulster, Northern Ireland, UK
{s.wu1, y.bi}@ulster.ac.uk
[2] Department of Computer and Information Sciences
University of Strathclyde, Glasgow, UK
f.crestani@cis.strath.ac.uk

Abstract. In data fusion, score normalization is a step to make scores, which are obtained from different component systems for all documents, comparable to each other. It is an indispensable step for effective data fusion algorithms such as CombSum and CombMNZ to combine them. In this paper, we evaluate four linear score normalization methods, namely the fitting method, Zero-one, Sum, and ZMUV, through extensive experiments. The experimental results show that the fitting method and Zero-one appear to be the two leading methods.

1 Introduction

Data fusion in information retrieval has been investigated by many researchers and has been a well-established technique. The essential idea is to more accurately estimate the relevance of all the retrieved documents for a given query via combining these retrieved documents from multiple information retrieval systems into a single list. It can be used as an alternative for implementing an effective information retrieval system, it can also be useful in the WWW environment as a meta-search engine to merge resultant documents from different Web search engines.

For data fusion algorithms, several factors need to be considered: (a) performance of all component results; (b) correlation among all component results; (c) available information associated with retrieved documents. A lot of research work (e.g., in [1,6,7,9,10]) has been done on these topics.

For a component result, it comprises a ranked list of documents. Sometimes a score is associated with each document to indicate the estimated relevance of the document to the information need. Various experiments (for example, in [2,8]) have confirmed that rankings are not as informative as scores, therefore, we should use score information preferentially if it is possible. In this paper we assume that all component systems provide scores for retrieved documents.

Different component systems may use very different policies to apply scores to retrieved documents. This difference can be both on range and distribution. Therefore, It is necessary to normalize scores for effective data fusion algorithms such as CombSum and CombMNZ to combine them.

H.T. Ng et al. (Eds.): AIRS 2006, LNCS 4182, pp. 642–648, 2006.

Fox and Shaw designed a series of data fusion algorithms including CombSum and CombMNZ [3]. CombSum sets the score of each document in the fused result to the sum of the scores obtained by the individual 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 [4] suggested a linear transformation method for score normalization. All normalized scores are in the range of [0,1], with the minimal score mapped to 0 and the maximal score to 1. This method is referred to as Zero-one later in this paper. Actually this can be improved since the top-ranked documents are not always relevant and bottom-ranked documents are not always irrelevant. A smaller range [a,b] ($0 < a < b < 1$) would be a better solution for score normalization. This method is referred to as the fitting method in this paper.

Montague and Aslam [5] 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.

In this paper, our major goal is to evaluate the four linear normalization methods (Zero-one, the fitting method, Sum and ZMUV). However, some of our experimental results are useful for us to better understand score normalization in general.

2 Effect of Different Score Normalization Methods to Data Fusion

In this section we report an experiment which investigates the effect of different score normalization methods on data fusion. For a group of results, we combine them using CombSum and CombMNZ several times, each time with a different score normalization method. Then we compare the performance of the fused results from the same data fusion method but different score normalization methods. For the fitting method, [0.06, 0.6] is arbitrarily chosen in all cases. Some other values may provides better performance. ZMUV cannot be properly used for CombMNZ because about half of the scores in every result are negative. Instead, as in Montague and Aslam's experiment [5], we used ZMUV2, a variation of ZMUV, which added 2 to every score normalized with ZMUV.

We used 4 groups of results, which were subsets of results submitted to TREC 7 (ad hoc track), 8 (ad hoc track), 9 (web track) and 2001 (web track) [7]. All selected results include 1000 documents for every query and have a mean average precision of 0.15 or over. In each group, we randomly select 3-10 systems for a data fusion run with CombSum and CombMNZ, and 200 runs were tested for any given number of results.

Mean average precision was used for the evaluation. Figures 1-2 show the experimental result.

The experimental results of all four score normalization methods are very consistent. Also all score normalization methods have a very similar performance for both data fusion methods CombSum and CombMNZ. We can observe that

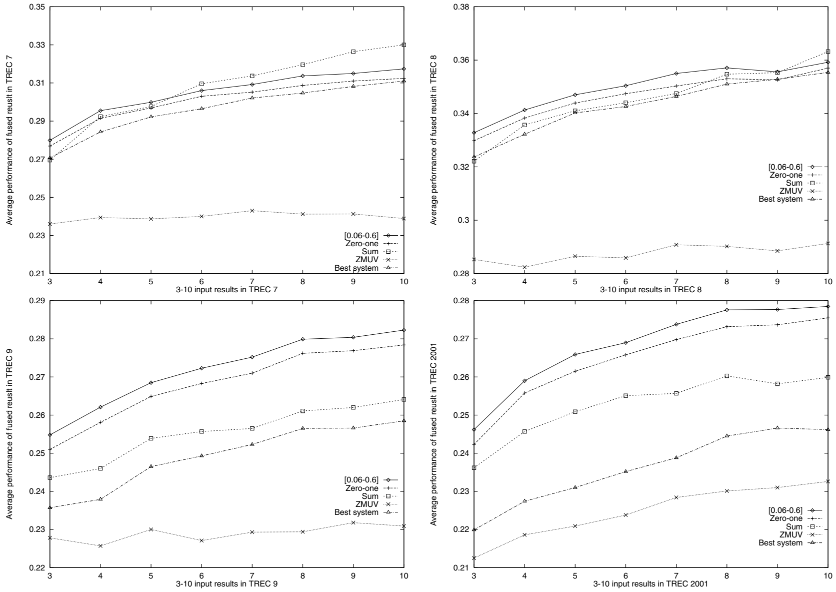

Fig. 1. Mean average precision of CombSum in TREC 7-2001 with different normalization methods

ZMUV2 is always the worst in all cases. The fitting method [0.06-0.6] is better than the Zero-one method using CombSum, but the Zero-one method is better than the fitting method [0.06-0.6] using CombMNZ. The fitting method and the Zero-one method perform better than Sum in most cases. In a few cases, Sum is a little better. Therefore, it suggests that the Zero-one method and the fitting method are very likely the best normalization method and ZMUV is the worst. The claim that both Sum and ZMUV are significantly better than Zero-one [5] cannot be supported in our experiment. According to our analysis in Section 2, Sum can lead to good fusing results when the (different) weights assigned to all input results fit their performances.

Another observation is: using the fitting method, Zero-one and Sum normalization methods, the fused results are better than the best component system on average, it confirms that data fusion is an effective technique to improve retrieval performance.

3 Distribution of Relevant Documents in Different Score Ranges

Actually, all these four score normalization methods (Zero-one, the fitting method, Sum, and ZMUV) are linear transformation methods. That is to say, it is straightforward to define a "general" schema to include all of them. We need to set up two pairs of ending points - the maximal score and the minimal score for raw scores

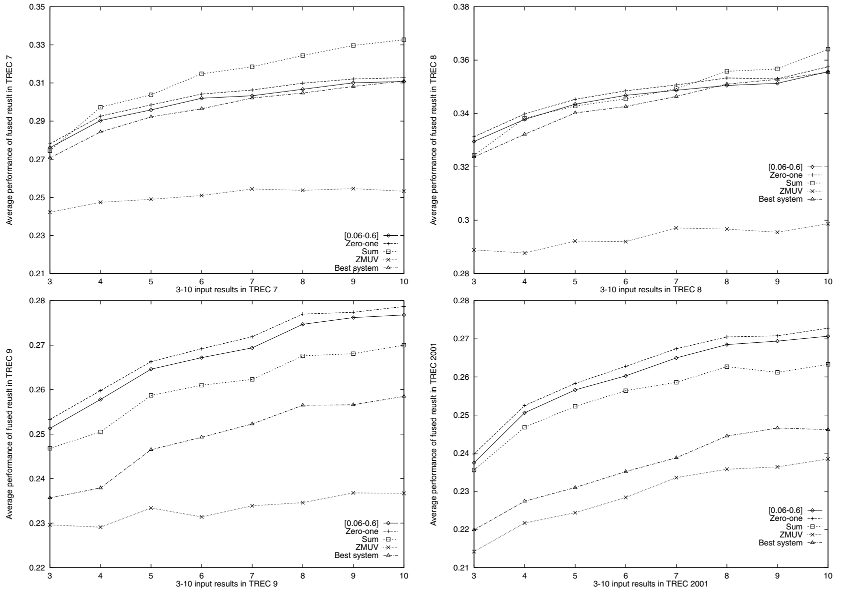

Fig. 2. Mean average precision of CombMNZ in TREC 7-2001 with different normalization methods

and normalized scores. We use *max* and *min* to denote the maximal and minimal normalized score, and *r_max* and *r_min* to denote the maximal and minimal raw score respectively. Please note that *r_max* is not necessarily the maximal raw score, and *r_min* is not necessarily the minimal raw score obtained by documents in the result. But instead we should guarantee that any raw score is between *r_max* and *r_min*. Then for any raw score *r_s*, we use the following formula to normalize it:

$$s = min + \frac{r\_s - r\_min}{r\_max - r\_min}(max - min)$$

If only considering one result, then there is no difference at all using any of the four methods. Any normalized score in any method can be linearly transformed into a corresponding score in another method. For example, suppose s_{zmuv} is a normalized score obtained with ZMUV, we can transform it into a normalized score for Zero-one by $s_{zero-one}$=$(s_{zmuv} - min_{zmuv})/(max_{zmuv} - min_{zmuv})$, where max_{zmuv} and min_{zmuv} denotes the normalized maximal score and the normalized minimal score with ZMUV respectively.

In data fusion, we need multiple results at the same time. The difference among these four normalization methods is that Sum and ZMUV normalize scores in different results into different ranges. The key issue is: can this bring any benefit to data fusion?

The goal of score normalization is to make scores in different results comparable. Ideally, if there is a linear relation between score and probability of relevance, it is a perfect condition for data fusion algorithms such as CombSum.

More specifically, we need to meet the following two conditions for n results r_i $(i = 1, 2, ..., n)$ retrieved from n information retrieval systems for the same information need:

- r is any one of the results, d_1 and d_2 are two randomly selected documents in r. d_1 obtains a normalized score of s_1, and d_2 obtains a normalized score of s_2, then the probability of d_1 being relevant to the information need equals to s_1/s_2 times of the probability of d_2 being relevant;
- for any two results r_1 and r_2 in n results, d_1 is a document in r_1 and d_2 is a document in r_2. If d_1's normalized score equals to d_2's normalized score, then these two documents have an equal probability of being relevant.

Next we report an experiment to examine to which extent the two requirements presented in the above can be met by those normalized scores. The procedure is as follows: first, for a result, we normalize the scores of its documents using a given score normalization method. Next we divide the whole range of scores into a certain number of intervals. Then for every interval, we count the number of relevant documents (*num*) and the total score (*t_score*) obtained by all the documents in that interval. In this way the ratio of *num* and *t_score* can be defined for every interval. We compare these ratios to investigate if the score normalization method is a good method or not.

We are not aimed at any particular component system here. A more effective way to understand the scoring mechanism of a particular information retrieval system is to investigate the scoring algorithm used in that system, not just analyse some query results produced by it. Rather than that, we try to find out some collective behaviour of a relatively large number of information retrieval systems in an open environment such as TREC. In such a situation, to analyse these query results becomes a sensible solution. There are a few reasons for this: Firstly, if the number of systems is large, it is a tedious task to understand the implementation mechanisms of all these systems; secondly, it is not possible if the technique used in some of the systems are not published; thirdly, the collective behaviour of these systems may not be clear even we know each of them well.

As in Section 2, we used the same four groups of results. We normalised all these results using the fitting method [0.06-0.6], Zero-one, Sum and ZMUV1 (a variation of ZMUV, which added 1 to every score normalized with ZMUV) over 50 queries. Then for all normalized results with a particular method, we divided them into 20 groups. Each group is corresponding to a interval and the scores of all documents in that group is located in the same interval.

For the fitting method, we divide [0.06,0.6] into 20 equal intervals [0.06,0.087), [0.087,0.114),..., [0.573,0.6]. For Zero-one, we divide [0,1] into 20 equal intervals [0,0.05), [0.05,0.1),..., [0.95,1]. For Sum, we divide [0,1] into 20 intervals [0,0.00005), [0.00005,0.0001), ..., [0.00095,1], since all normalized scores in Sum are very small and most of them are smaller than 0.001. For ZMUV1, 20 intervals ($-\infty$, 0.1), [0.1,0.2),..., [1.9,$+\infty$) are defined. For each group, we calculate its ratio of *num* and *t_score*. All results are shown in Figure 3. The ideal situation is that the curve is parallel to the horizontal axis, which means a linear relation between score and document's relevance and CombSum is a perfect method for

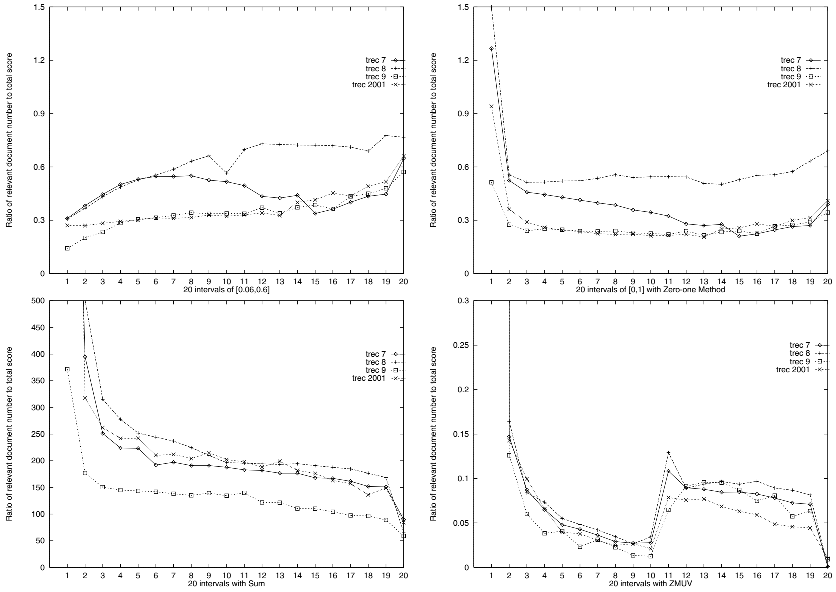

Fig. 3. Score distribution in TREC 7-2001 with different normalization methods

data fusion. It demonstrates that the fitting method is the best since all the curves generated are the most flat. For all other score normalization methods, their corresponding curves are quite flat as well, which suggests that a linear function is a good option to describe the relation between score and document's relevance.

For Zero-one, Sum and ZMUV1, we observe that very often their curves fly into the sky in Intervals 1 and 2. This is understandable since all the normalized scores with Zero-one and Sum are approaching 0 in Intervals 1 and 2. It is not the case for their raw scores and there are a few relevant documents in these two intervals. Therefore the curves are deformed. The situation is even worse for ZMUV1 since negative and positive scores coexist in Interval 1.

4 Conclusion

In this paper, we have evaluated four linear score normalization methods, namely the fitting method, Zero-one, Sum and ZMUV. Comparison analysis and extensive experimentation has been carried out to investigate them. On average, The fitting method and the Zero-one method appear to be the two leading methods. More specifically, the fitting method is more favourable to CombSum, while the Zero-one method is more favourable to CombMNZ. Sum is not as good as the fitting method and Zero-one but outperforms them occasionally. ZMUV always performs very badly and it does not seem to be a proper score normalization method.

References

1. E. Amitay, D. Carmel, R. Lempel, and A. Soffer. Scaling ir-system evaluation using term relevance sets. In *Proceedings of the 27th Annual International ACM SIGIR Conference*, pages 10–17, Sheffield, UK, July 2004.
2. J. A. Aslam and M. Montague. Models for metasearch. In *Proceedings of the 24th Annual International ACM SIGIR Conference*, pages 24–37, Gaithersburg, Maryland, USA, November 2003.
3. E. A. Fox and J. Shaw. Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2)*, pages 243–252, Gaitherburg, MD, USA, August 1994.
4. J. H. Lee. Analysis of multiple evidence combination. In *Proceedings of the 20th Annual International ACM SIGIR Conference*, pages 267–275, Philadelphia, Pennsylvania, USA, July 1997.
5. M. Montague and J. A. Aslam. Relevance score normalization for metasearch. In *Proceedings of ACM CIKM Conference*, pages 427–433, Berkeley, USA, November 2001.
6. I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of 24th Annual International ACM SIGIR Conference*, pages 66–73, New Orleans, Louisiana, USA, September 2001.
7. TREC. http://trec.nist.gov/.
8. S. Wu and F. Crestani. Data fusion with estimated weights. In *Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management*, pages 648–651, McLean, VA, USA, November 2002.
9. S. Wu and F. Crestani. Methods for ranking information retrieval systems without relevance judgments. In *Proceedings of the 2003 ACM Symposium on applied computing (SAC)*, pages 811–816, Melbourne, Florida, USA, March 2003.
10. S. Wu and S. McClean. Performance prediction of data fusion for information retrieval. *Information Processing & Management*, 42(4):899–915, July 2006.