Relevance Score Normalization for Metasearch[*]

Mark Montague
Department of Computer Science
Dartmouth College
6211 Sudikoff Laboratory
Hanover, NH 03755
montague@cs.dartmouth.edu

Javed A. Aslam
Department of Computer Science
Dartmouth College
6211 Sudikoff Laboratory
Hanover, NH 03755
jaa@cs.dartmouth.edu

ABSTRACT

Given the ranked lists of documents returned by multiple search engines in response to a given query, the problem of *metasearch* is to combine these lists in a way which optimizes the performance of the combination. This problem can be naturally decomposed into three subproblems: (1) *normalizing* the relevance scores given by the input systems, (2) *estimating* relevance scores for unretrieved documents, and (3) *combining* the newly-acquired scores for each document into one, improved score.

Research on the problem of metasearch has historically concentrated on algorithms for *combining* (normalized) scores. In this paper, we show that the techniques used for *normalizing* relevance scores and *estimating* the relevance scores of unretrieved documents can have a significant effect on the overall performance of metasearch. We propose two new normalization/estimation techniques and demonstrate empirically that the performance of well known metasearch algorithms can be significantly improved through their use.

1. INTRODUCTION

Metasearch is a well-established technique for improving the performance of document retrieval systems [4, 14, 7, 22, 12, 9]. The fundamental idea is to more accurately assess the relevance of each document (with respect to a given query) by first giving the query to a number of different search engines and then combining their output ranked lists into a single, improved ranked list.

Applications: Metasearch algorithms are used in basically two ways: *externally* or *internally*. Web metasearch engines like MetaCrawler, ProFusion, SavvySearch, MetaFerret, InFind, etc. are external: they combine the output of "complete" search engines, as a kind of post-processing, value-adding stage. On the other hand, a search engine may be composed of a number of very different and perhaps very simple "sub-engines," not intended for use on their own, but very effective when combined with a metasearch algorithm. In this internal use, the metasearch component is central to the operation of the system. Metasearch offers a systematic way of incorporating all of the various types of evidence available to a given search engine. For example, in the context of web page retrieval, many sources of information exist: each page has text, in-links, out-links, images, tags, keywords, and structural information. For each of these elements, numerous indexing and retrieval algorithms may exist. A metasearch algorithm can be used to automatically combine the information provided by these disparate retrieval components.

Scores vs. Ranks: The final desired output of a search system (or for that matter a metasearch system) is usually a ranked list of documents, in more-relevant to less-relevant order. But usually *relevance scores* are computed for each document first, from which the rankings are then derived. If these "intermediate" relevance scores from each input system are available to the metasearch system, it may be advantageous, as they generally contain more information than the mere rankings: the ranked ordering can be computed from the relevance scores, but not vice-versa. In this paper, we assume that the metasearch algorithms are always given access to the underlying input systems' relevance scores. The popular Comb{SUM,MNZ,. . . } [9] metasearch algorithms make this assumption, though other metasearch algorithms have been developed which do not require relevance scores [4, 12].

Training Data: Sometimes training data is available to the metasearch system; for example statistics about the average performance of each input system or a sample set of queries with associated documents judged relevant or irrelevant by human judges. The algorithms that we investigate in this paper do not require training data.

Database Overlap: The metasearch problem can be studied in the context of *data fusion*, where each input search system is over the same data set, *collection fusion*, where their databases are disjoint, or even with arbitrarily overlapping databases. In this study we restrict ourselves to the data fusion setting—a setting more like internal metasearch than external.

[*] This work generously supported by NSF grants EIA-9802068, BCS-9978116, and Career Award CCR 00-93131.

Benefits: Metasearch can provide improved recall, precision, and consistency [4, 14].

We propose the following natural decomposition of the metasearch problem into three sub-problems: relevance score *normalization*, *estimation*, and *combination*.

Normalization addresses the problem that relevance scores given to the same document by different input systems may be totally incomparable. Although they are usually real numbers, they may be on different scales, in different ranges, and distributed differently. The goal of relevance score normalization is to make relevance scores comparable across input systems. The normalization stage is the topic of this paper.

Estimation, or *unretrieved document relevance score estimation*, addresses the problem that the input systems return different documents. Since we assume that the input systems' databases contain the same set of documents, if system S does not return a document d, we know S would have given d a lower score than any of the observed scores from S. But how much lower? If we can estimate the score S would have given d, we may simplify the score combination stage: each document will have a score from each system. We will only raise the issue of estimation in this paper, leaving its resolution to future work.

Combination is the final relevance score calculation for each document as a function of the newly-comparable (through normalization), and newly-complete (through unretrieved document score estimation) set of input scores. This topic has been the traditional subject of metasearch research; for the purposes of this paper, we will focus on the popular Comb{SUM,MNZ,...} [9] metasearch algorithms.

In most work on metasearch, the combination technique has been the focus of the study, with the normalization technique assumed to be less important. In this paper, we take the opposite approach, fixing two simple, standard combination algorithms (CombMNZ and CombSUM), and experimenting with different normalization algorithms.

We find that indeed the normalization stage is important: even very simple modifications to the standard shift-and-scale normalization scheme yield significant improvements for standard combination algorithms. The key factor seems to be removing the normalization's sensitivity to outliers.

In the remainder of this work, we first review the relevant literature. We then present the normalization, estimation, and combination algorithms that we use, and detail the results of a number of experiments on TREC data. Finally, we conclude with directions for future research.

2. RELATED WORK

The use of data fusion to combine document retrieval results has received considerable attention in the past few years: it has been the subject of a number of doctoral dissertations [5, 15, 22, 19], journal articles [21, 10, 24], and conference papers [8, 6, 11, 12, 16, 23], being especially used in the TREC competitions [9, 20, 17]. In this section we review the results of these publications as they relate to our work.

Fox and Shaw [9] designed the Comb{SUM,MNZ,...} metasearch algorithms which we will discuss later in this work.

Lee [12] performed experiments with CombMNZ, suggesting that scores be normalized before combination by shifting and scaling them into the range $[0, 1]$ so that they are comparable across systems.

In the context of the filtering problem, Hull et al. [11] try averaging algorithms. In their work, each of the four input systems that they fuse output estimates for the probability of relevance of each document, so they need not normalize: scores are already directly comparable. They try both directly averaging these probabilities as well as averaging the log-odds ratios, $\log \frac{p}{1-p}$

Croft [7] reviews the literature surrounding fusion in IR. He concludes that the systems being combined should: (1) have compatible outputs, (2) each produce accurate probability estimates, and (3) be independent of each other. His first requirement, that the outputs be compatible, is the goal of normalization.

Aslam and Montague's Bayes-fuse algorithm [4, 14] can be viewed as a normalization and estimation technique that relies on training data. For each input system, two probability distributions are estimated based on the training data: the probability of a document being given rank r given that it is relevant or irrelevant. The final normalized relevance score is computed as the log odds of a document receiving rank r. Although this work is rank-based, the same techniques could be applied to relevance scores.

Bartell [5], Vogt [25, 24, 22, 23], and others experiment with linearly combining the normalized relevance scores given to each document. This work focuses on the training required to learn the weights to give each system, but does not question the wisdom of the standard, $[0, 1]$ normalization.

Manmatha et al. [13] model the distribution of scores for each query, using a mixture model of a negative exponential distribution (for irrelevant documents) and a Gaussian distribution (for relevant docs). Using EM procedure, they tune the model for each query. This allows them to compute the final normalized relevance score for a document as the probability that it is relevant based on its original score and the model. They find that this normalization scheme yields some improvement.

3. SCORE NORMALIZATION

We assume that the relevance score given to a document by an input search system is a real number in the interval $(-\infty, \infty)$ (See Thompson's CEO model [21] for an example where this is not the case). For each query, each input system S_i returns their top n documents, along with their associated scores. Let us denote the score given by S to a document a as $scr_S(a)$. By convention, if $scr_S(a) > scr_S(b)$, then we know that S is asserting that a is more relevant, or more likely relevant than b. But beyond this, we do not assume that we know how to interpret the relevance scores; for example, we do not know if they represent probabilities of relevance, odds of relevance, log odds of relevance, or some other measure. We treat each input system as a black box expert that need not and perhaps cannot reveal how it generated these scores or how to interpret them. Hence the problem: how can we combine the scores if we do not know what they mean? Even if we cannot interpret them individually, perhaps we can make the scores comparable between systems. This is the goal of normalization: to make scores directly comparable between input systems.

Traditionally normalization has been accomplished by a linear transformation: shifting and scaling relevance scores into the range $[0, 1]$ (see for instance [12]). But is this optimal?

We define three desirable qualities of a normalization scheme:

Shift invariant: Let R be a set of relevance scores and R_c be R shifted by an additive constant c. That is, for $scr(a) \in R$, $scr(a_c) = scr(a) + c \in R_c$. Let $scr'(a)$ denote the normalized score of document a. Then we say that a normalization scheme is shift invariant if $scr'(a) = scr'(a_c)$; both the shifted and unshifted set of scores normalize to the same set. In other words, we would like our normalization scheme to be insensitive to mere shifts of the input.

Scale invariant: Similarly, we would like our normalization scheme to be insensitive to its input being scaled by a multiplicative constant.

Outlier insensitive: It is also desirable that normalization not be overly-sensitive to the score of a single document. That is, adding a "reasonable" outlier does not significantly change the normalized score for the rest of the docs.

In this paper we test three simple normalization schemes, summarized in table 1: the standard $[0, 1]$ scheme, and two others that we designed in order to avoid sensitivity to outliers.

Name	Method
Standard	Shift min to 0, scale max to 1
Sum	Shift min to 0, scale sum to 1
ZMUV	Shift mean to 0, scale variance to 1

Table 1: Normalization algorithms. We propose the sum and ZMUV norms as simple and effective replacements for the standard norm.

Standard Norm: The standard norm is shift and scale invariant, but is sensitive to the max and min scores given for each query, and hence highly sensitive to outliers.

Sum Norm: The sum norm is shift and scale invariant, and it is sensitive only to the min score given for each query. The sum of the scores, as an aggregate statistic, is more robust. In practice the min score is not an outlier, due to the fact that ranked lists are truncated to only return a certain number of documents. So in practice, the sum norm is fairly outlier insensitive. Note also that equivalent results would be obtained using the average score instead of the sum of scores.

ZMUV Norm (Zero-Mean, Unit-Variance): ZMUV is shift and scale invariant. It is also outlier insensitive: the ZMUV transformation does not depend directly on either the min or max scores given (except insomuch as they individually affect the mean and variance of the collection). The mean and variance of the relevance scores are both aggregate, and thus more robust, statistics.

Note that none of these normalization schemes require training data.

4. SCORE ESTIMATION

In this paper, we use only the simplest possible score estimators for unretrieved docs. Typically, a normalized relevance score of zero is assigned to unretrieved documents [9], and we also use a normalized relevance score of zero for the standard and sum normalization schemes.

For the ZMUV normalization scheme, wherein a normalized relevance score of zero would imply average relevance, we instead assign a relevance score of -2 to unretrieved documents. In other words, unretrieved documents are assigned a relevance score two standard deviations below the mean.

Note that we have not attempted to optimize these constants; in future work we will investigate unretrieved document score estimation and its effect on metasearch.

5. SCORE COMBINATION

Fox and Shaw [9] designed some of the most simple, popular, and effective metasearch algorithms to date. They are summarized in table 5. CombMNZ works best, with CombSUM a close second. Usually they are used with the standard norm, and unretrieved documents are assigned a relevance score of zero. We will use both CombSUM and CombMNZ to compare normalization algorithms.

Name	New relevance score is:
CombMIN	minimum of individual rels
CombMED	median of individual rels
CombMAX	maximum of individual rels
CombSUM	sum of individual rels
CombANZ	CombSUM ÷ num nonzero rels
CombMNZ	CombSUM × num nonzero rels

Table 2: Fox and Shaw's Comb{SUM,MNZ,...} fusion algorithms. "Rels" stands for relevance scores.

6. EXPERIMENTS

6.1 Data Sets

We use systems submitted to the annual Text REtrieval Conference (TREC) as input to our metasearch algorithms. TREC offers large, standard data sets with many ranked lists for each query, ready to be fused. Also, each system submits 50 queries, so we can more accurately assess performance by averaging over the queries. Table 3 shows information about three of the data sets. Note that in TREC, each system is allowed to return up to 1000 documents for each query. For the TREC 3 and TREC 5 data, we used the data

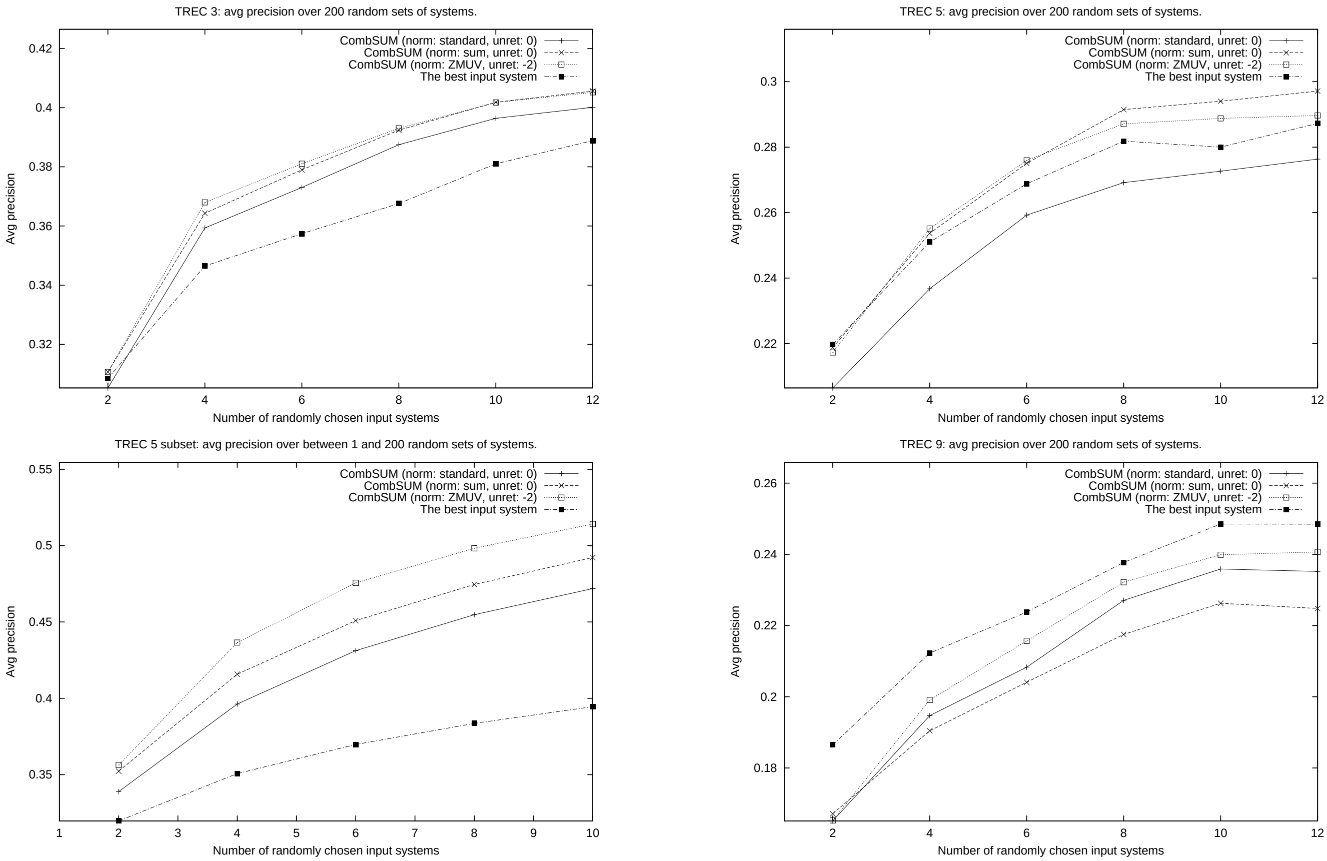

Figure 1: The performance of CombSUM over four data sets using different normalization schemes.

Data set	Topics	No. Sys
TREC 3	151–200	40
TREC 5	251–300	61
TREC 9	451–500	105

Table 3: Three of the four data sets used in our experiments.

submitted to the TREC "adhoc" task. For TREC 9, the adhoc task had been replaced by the "web" track. Therefore, over that data set we are fusing the results of World Wide Web search engines. In TREC terminology, a "topic" is a query; they are numbered consecutively. In the table, the column labelled "No. Sys" contains the number of search systems that submitted results to TREC that year—this is the number of systems available for us to fuse. Also note that the TREC 9 dataset was large and challenging. Some of the input systems from TREC 9 purposefully masked their scores by replacing them with reversed ranks. One system had an average precision of zero.

Our fourth data set consists of a subset of the TREC 5 data set as defined by Vogt [23]. In particular, it contains only 10 of the 61 TREC 5 systems, and only 10 of the 50 TREC 5 queries. This subset was chosen by Vogt to highlight the strengths of the metasearch technique: it contains retrieval systems chosen to maximize diversity, as measured by nine similarity criteria. The systems are: CLTHES, DCU961, anu5aut1, anu5man6, brkly17, colm1, fsclt4, gm96ma1, mds-002, and uwgcx0. The queries were chosen for their large number of relevant documents: queries 257, 259, 261, 272, 274, 280, 282, 285, 288, and 289. We include this data set because of its diverse inputs: we expect that it more closely models the environment of "internal" metasearch than the other data sets.

6.2 Experimental Setup

We examine the performance of metasearch strategies when combining random groups of retrieval systems. Each data point represents the average value obtained over 200 trials (or as many as are combinatorially possible) performed as follows. Randomly select a set of n (for $n \in \{2, 4, ...12\}$) input systems, apply the metasearch algorithm to these systems, and record the average precision of the metasearch algorithm's output. (Additionally, we record the average precision of the best underlying system in order to meaningfully assess the improvement gained, if any.) This experiment is designed to test how well a fusion algorithm performs on average, and to see how the algorithm improves when more input systems are available. A successful system will consistently improve on the *best* of its inputs, no matter how many input systems are available.

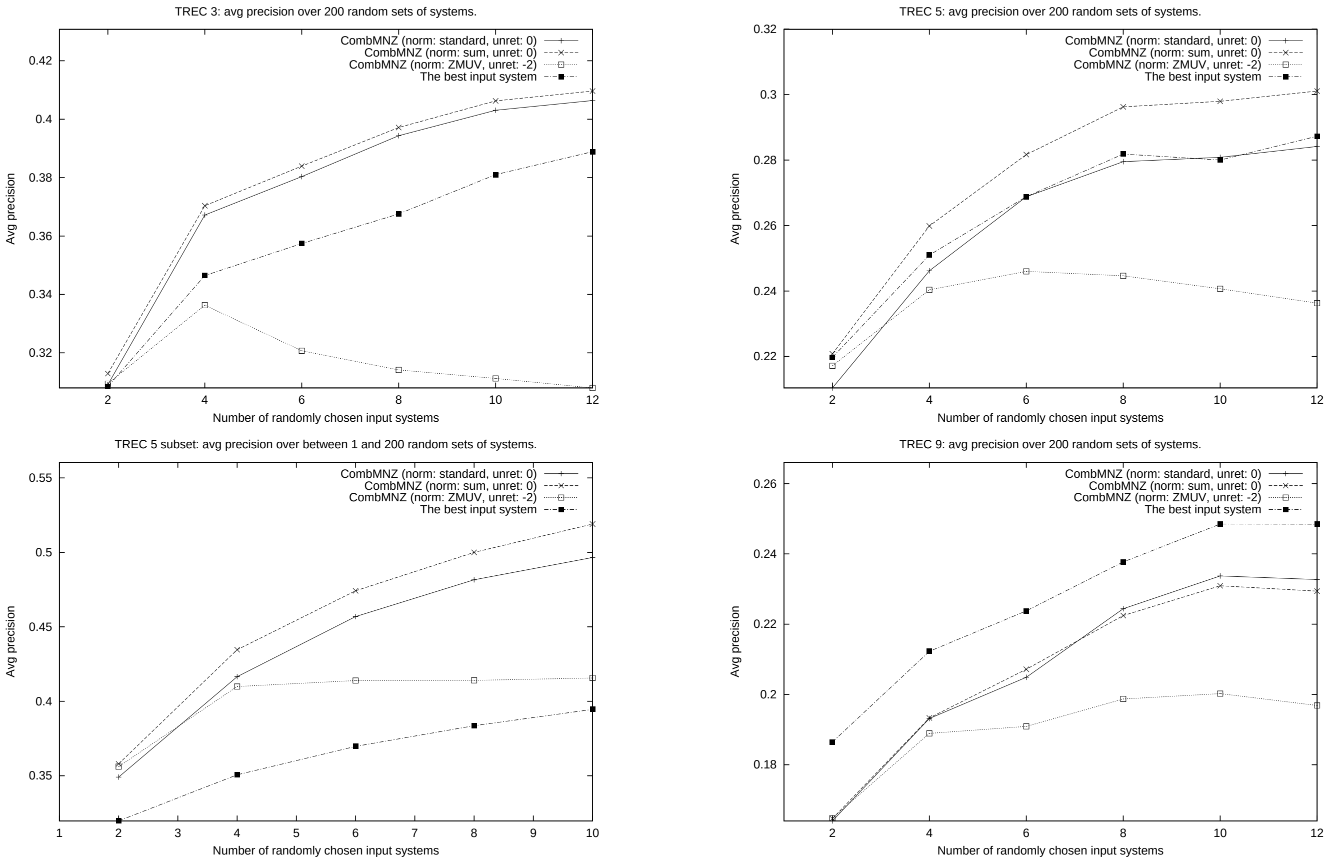

Figure 2: The performance of CombMNZ over four data sets using different normalization schemes.

6.3 Experimental Results

Figure 1 shows the results of using different normalization schemes with CombSUM. The ZMUV norm almost always performed significantly better than the standard norm. The ZMUV norm also usually outperforms the sum norm. Indeed, on TREC 5, both ZMUV and the sum norm cause CombSUM to outperform the best input, whereas it had not previously.

Figure 2 shows the results of using different norms with CombMNZ. ZMUV performs very poorly with CombMNZ, but this is easily understood: CombMNZ assumes all relevance scores are positive. But roughly half of the scores produced by ZMUV are negative since it shifts the mean to zero. Thus the final step in CombMNZ, multiplying by the number of systems that returned a document, actually penalizes those documents with negative scores whenever they are returned by many systems.

The sum norm, however, almost always improves CombMNZ, sometimes significantly. In TREC 5 for instance the standard norm has trouble reaching the performance of the best input system, but the sum norm easily exceeds it.

To explore the problem of CombMNZ interacting poorly with ZMUV, we tried a “2MUV” normalization scheme: shift the mean to two instead of zero (still using unit variance). This will force most scores to be positive, so that CombMNZ can handle them properly. The results are shown in Figure 3. Here we can see that the ZMUV norm can be made to work with CombMNZ, though the sum norm usually works as well. We expect that with proper normalization and unretrieved document score estimation, CombSum will be a better system than CombMNZ.

Figure 4 compares the CombSUM–ZMUV combination with the CombMNZ–sum combination (along with the standard CombMNZ and best input for reference). There is no clear winner between these two—each is best for one data set and they tie on two data sets.

7. CONCLUSIONS

Relevance score normalization is an important step in the metasearch problem. By simply using more robust statistics than max and min in the normalization scheme we can achieve significant improvements in the performance of both CombMNZ and CombSUM. The inferior performance of the standard shift-and-scale normalization scheme is likely due to its dependence on the value of *outliers* (the min and max relevance scores); normalization schemes which are not as sensitive to outliers (such as those proposed) yield better performance.

The proper estimation of normalized relevance scores for un-

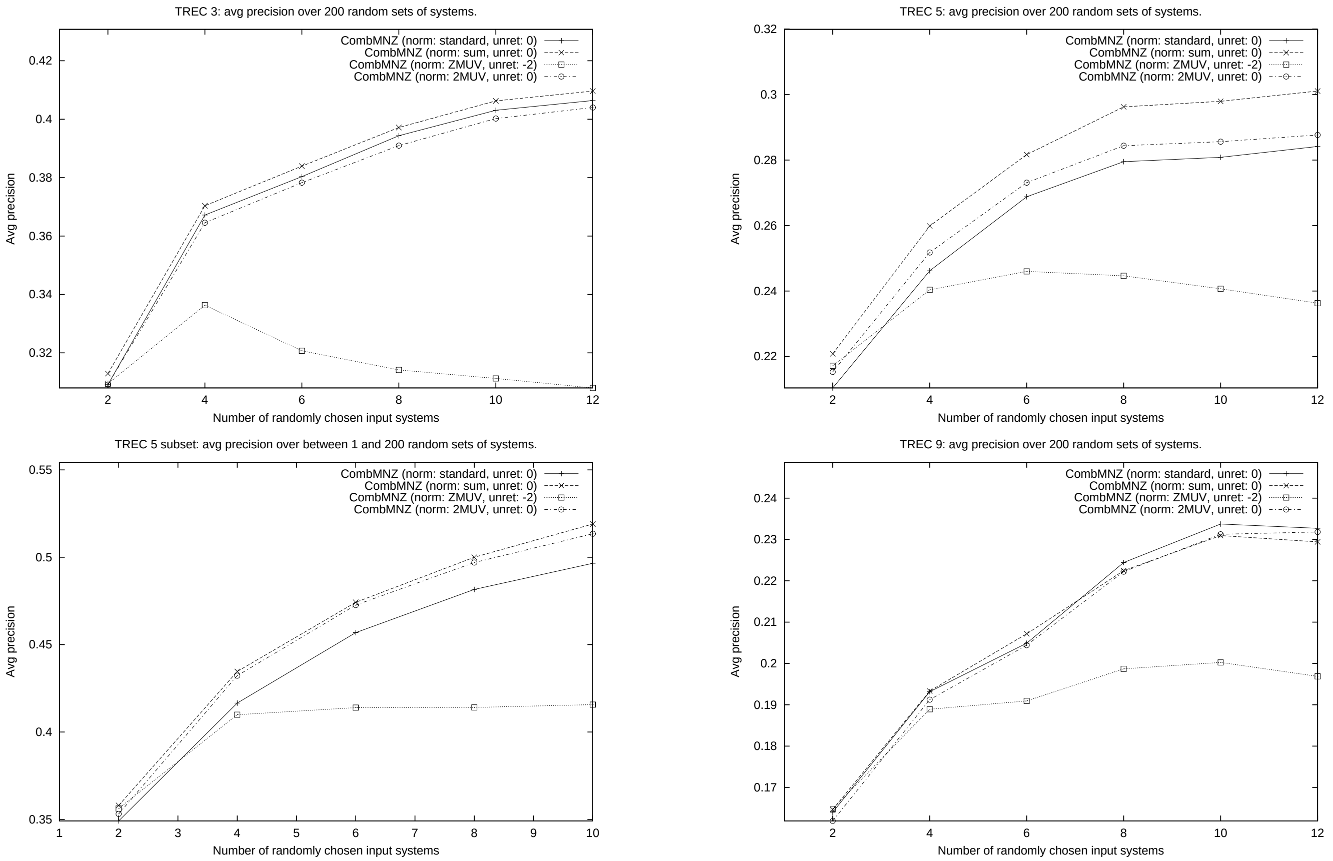

Figure 3: The performance of the 2MUV variant of the ZMUV normalization with CombMNZ.

retrieved documents is an open problem and the subject of future work. Simply assigning a normalized relevance score of zero (equivalent to the relevance score assigned to the lowest ranked retrieved document) seems overly optimistic, though this is classically done for the standard normalization scheme and we have adopted it for the sum normalization scheme as well. Assigning a normalized relevance score two standard deviations below the mean for unretrieved documents in the ZMUV scheme, while reasonable, is perhaps non-optimal. We expect that further research on properly estimating the normalized relevance scores which should be assigned to unretrieved documents will lead to further improvements in the performance of metasearch.

8. REFERENCES

[1] *TREC 2*, Gaithersburg, MD, USA, Mar. 1994. U.S. Government Printing Office, Washington D.C.

[2] *TREC 5*, Gaithersburg, MD, USA, 1997. U.S. Government Printing Office, Washington D.C.

[3] *ACM SIGIR 2001*, New Orleans, Louisiana, USA, 2001. ACM Press, New York.

[4] J. Aslam and M. Montague. Models for metasearch. In *ACM SIGIR 2001* [3].

[5] B. T. Bartell. *Optimizing Ranking Functions: A Connectionist Approach to Adaptive Information Retrieval*. PhD thesis, University of California, San Diego, 1994.

[6] N. Belkin, P. Kantor, C. Cool, and R. Quatrain. Combining evidence for information retrieval. In *TREC 2* [1], pages 35–43.

[7] W. B. Croft. Combining approaches to information retrieval. In W. B. Croft, editor, *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*, chapter 1. Kluwer, 2000.

[8] E. A. Fox, M. P. Koushik, J. Shaw, , R. Modlin, and D. Rao. Combining evidence from multiple searches. In *TREC 1*, pages 319–328, Gaithersburg, MD, USA, Mar. 1993. U.S. Government Printing Office, Washington D.C.

[9] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *TREC 2* [1], pages 243–249.

[10] K. L. Fox, O. Frieder, M. Knepper, and E. Snowberg. SENTINEL: A multiple engine information retrieval and visualization system. *Journal of the ASIS*, 50(7), May 1999.

[11] D. A. Hull, J. O. Pedersen, and H. Schütze. Method combination for document filtering. In *ACM SIGIR*

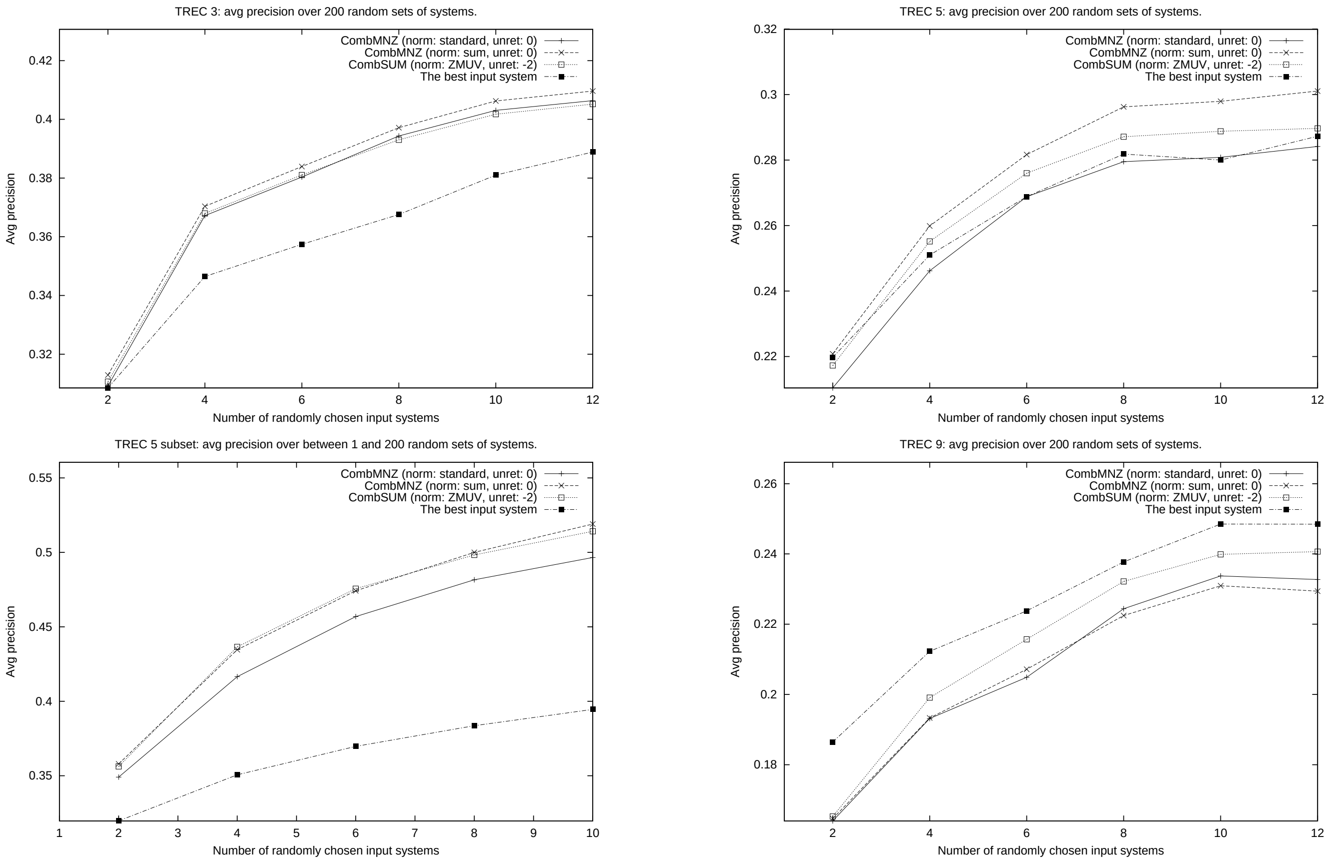

Figure 4: The performance of the best algorithms over four data sets using different normalization schemes.

'96, pages 279–287, Zurich, Switzerland, 1996. ACM Press, New York.

[12] J. H. Lee. Analyses of multiple evidence combination. In *ACM SIGIR '97*, pages 267–275, Philadelphia, Pennsylvania, USA, July 1997. ACM Press, New York.

[13] R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In *ACM SIGIR 2001* [3].

[14] M. Montague and J. Aslam. Metasearch consistency. In *ACM SIGIR 2001* [3].

[15] K. B. Ng. *An Investigation of the Conditions for Effective Data Fusion in Information Retrieval*. PhD thesis, School of Communication, Information, and Library Studies, Rutgers University, 1998.

[16] K. B. Ng and P. B. Kantor. An investigation of the preconditions for effective data fusion in ir: A pilot study. In *Proceedings of the 61th Annual Meeting of the American Society for Information Science*, 1998.

[17] K. B. Ng, D. Loewenstern, C. Basu, H. Hirsh, and P. B. Kantor. Data fusion of machine-learning methods for the TREC5 routing task (and other work). In *TREC 5* [2], pages 477–487.

[18] *Content-Based Multimedia Information Access (RIAO)*, Paris, France, Apr. 2000.

[19] E. W. Selberg. *Towards Comprehensive Web Search*. PhD thesis, University of Washington, 1999.

[20] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *TREC 3*, pages 105–108, Gaithersburg, MD, USA, Apr. 1995. U.S. Government Printing Office, Washington D.C.

[21] P. Thompson. A combination of expert opinion approach to probabilistic information retrieval, part 1: the conceptual model. *Information Processing and Management*, 26(3):371–382, 1990.

[22] C. C. Vogt. *Adaptive Combination of Evidence for Information Retrieval*. PhD thesis, University of California, San Diego, 1999.

[23] C. C. Vogt. How much more is better? Characterizing the effects of adding more IR systems to a combination. In RIAO [18], pages 457–475.

[24] C. C. Vogt and G. W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, Oct. 1999.

[25] C. C. Vogt, G. W. Cottrell, R. K.Belew, and B. T. Bartell. Using relevance to train a linear mixture of experts. In *TREC 5* [2], pages 503–515.