A Sampling Technique for Efficiently Estimating Measures of Query Retrieval Performance Using Incomplete Judgments

Javed A. Aslam JAA@CCS.NEU.EDU
College of Computer and Information Science, Northeastern University, Boston, MA 02115 USA

Virgiliu Pavlu VIP@CCS.NEU.EDU
College of Computer and Information Science, Northeastern University, Boston, MA 02115 USA

Emine Yilmaz EMINE@CCS.NEU.EDU
College of Computer and Information Science, Northeastern University, Boston, MA 02115 USA

Abstract

We consider the problem of evaluating the performance of query retrieval systems, and we propose a sampling technique for efficiently estimating standard measures of retrieval performance using incomplete judgments. Unlike existing techniques which (1) rely on effectively complete, and thus prohibitively expensive, relevance judgment sets, (2) produce biased estimates of standard performance measures, or (3) produce estimates of non-standard measures thought to be correlated with these standard measures, our proposed sampling technique produces unbiased estimates of the standard measures themselves.

Our technique is based on random sampling, and as such, the greater the number of random samples (i.e., relevance judgments), the higher the accuracy of our estimators. We further derive a number of enhancements to the general technique which allow one to determine accurate estimates for the standard performance measures associated with large collections of systems from a single, small judgment pool. Our experiments with the benchmark TREC data collection indicate that highly accurate estimates of these standard measures can be obtained using a number of relevance judgments as small as 2% of the typical judgment pool.

1. Introduction

We begin by describing relevant background information from the field of Information Retrieval. In Section 1.1 we describe the standard measures used to evaluate the quality of a query retrieval system's performance. In Section 1.2 we describe the manner in which retrieval systems are most often evaluated with respect to these measures, and we describe how this standard evaluation methodology is, in practice, either expensive or only weakly approximate of "ground truth." Finally, in Section 1.3 we introduce our work: a sampling technique for efficiently estimating retrieval performance using incomplete judgments.

1.1. Performance measures

The Information Retrieval community has developed a number of standard measures for assessing the quality of a ranked list of documents returned in response to a user query. (Consider, for example, the problem of assessing the quality of a web search engine's results.) Virtually all standard measures of query retrieval performance are based on the *binary relevance* model; i.e., each document is judged to be either *relevant* (label = 1) or *non-relevant* (label = 0) with respect to the given query.

Perhaps the simplest standard measure of performance is *precision at standard cutoffs*. For example, precision-at-cutoff 10, $PC(10)$, is simply the fraction of documents among the first 10 in a list which are relevant. This may, for example, correspond to the accuracy of the first page of a web search engine's results. $PC(c)$ can be calculated, in principle, for any c; however, for consistency the IR community most often reports $PC(c)$ at the standard cutoffs

Appearing in *Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data*, Bonn, Germany, August 2005. Copyright 2005 by the author(s).

$c = 5, 10, 15, 20, 30, 100, 200, 500$, and 1000.

The performance of a given retrieval system is most often calculated with respect to many returned lists, each corresponding to one of a collection of "representative" queries, and the *average* of these performances is reported, e.g., *mean precision-at-cutoff 10*, $MPC(10)$. However, precision-at-standard-cutoff values do not necessarily "average" well. For example, achieving a high $PC(100)$ value may be quite easy given a query which has thousands of relevant documents while it would be impossible given a query which has only a few dozen relevant documents. R-precision, precision-at-cutoff R where R is the total number of documents relevant to a query, largely avoids these averaging issues. R-precision (RP) is always a value in the range $[0, 1]$, and the value 1 is achieved if and only if the retrieved results are "perfect," i.e., all relevant documents are retrieved before any non-relevant documents. R-precision is known to be a good overall measure of performance, and mean R-precision values are widely reported in the IR literature.

Perhaps the most widely reported overall measure of retrieval performance is (mean) *average precision.* The average precision of a list is the average of the precisions at each relevant document in that list. For example, given a query with three relevant documents retrieved at ranks 2, 5, and 8 in a list, the average precision would be

$$\begin{aligned} AP &= (PC(2) + PC(5) + PC(8))/3 \\ &= (1/2 + 2/5 + 3/8)/3 \\ &= 0.425 \end{aligned}$$

Precisions at unretrieved relevant documents are assumed to be zero, and thus average precision is effectively the sum of the precisions at retrieved relevant documents divided by R. Note that like R-precision, average precision[1] (AP) is always a value in the range $[0, 1]$, and the value 1 is achieved if and only if the retrieved results are "perfect."

1.2. Pooling

The problem of building test collections for evaluating the performance of retrieval systems has been widely studied in the information retrieval community, perhaps most prominently in the annual text retrieval conference TREC (Harman, 1995). In TREC, collections of retrieval systems are evaluated by (1) constructing a test collection of documents, (2) constructing a test collection of queries, (3) judging the relevance of each document to each query, and (4) assessing the quality of the ranked lists of documents returned by each retrieval system for each topic using standard measures of performance such as mean average precision. For meaningfully large collections of documents and/or queries, Step (3) is for all practical purposes impossible: in a typical TREC, for example, one might be faced with the prospect of assessing the relevance of 1,000,000 documents to each of 50 queries. To overcome this difficulty while obtaining substantially identical performance assessments, a relatively small subset of the documents is chosen with respect to each query, and the relevance of these documents to the query is assessed. Documents outside this "pool" are assumed to be non-relevant. The pool of documents to be judged is typically constructed by taking the union of the top c documents returned by each system in response to a given query. In TREC, $c = 100$ has been shown to be an effective cutoff in evaluating the relative performance of retrieval systems (Harman, 1995; Zobel, 1998). Shallower pools (Zobel, 1998) and greedily chosen dynamic pools (Cormack et al., 1998; Aslam et al., 2003) have also been studied in an attempt to alleviate the burden of requiring large numbers of relevance judgments.

While pooling greatly reduces the number of relevance judgments required for effective system evaluation, it can still be quite expensive. In the TREC conference, for example, upwards of 100 systems return lists of 1,000 ranked documents in response to each of 50 topics. While many of the top documents are retrieved by multiple systems, thus reducing the overall size of the depth 100 pool, the total number of relevance judgments is still substantial. For example, in TREC8 (Voorhees & Harman, 2000) 86,830 relevance judgments were used to assess the quality of the retrieved lists submitted by 129 systems in response to 50 topics. (See Table 1.)

1.3. Our work

Our goal in this work is to accurately evaluate the performance of retrieval systems using few relevance judgments. We focus on efficient estimations of the aforementioned standard measures of query retrieval performance and in particular *average precision* since it is perhaps the most widely used and cited overall measure of performance in the IR community. Unlike previously proposed methodologies based on shallow or greedily chosen pools which tend to produce biased estimates using few judgments (Aslam et al., 2003; Cormack et al., 1998; Zobel, 1998) or methodologies based on estimating measures other than the

[1] Average precision is approximately the area under the precision-recall curve (Baeza-Yates & Ribeiro-Neto, 1999); as such, it is analogous to the area under the ROC curve often cited in the machine learning literature.

	TREC		
Pool	7	8	10
Depth	$n = 103$	$n = 129$	$n = 96$
1	32	40	33
3	76	95	82
5	114	144	128
10	207	260	234
20	389	494	425
100	1860	2176	1705

Table 1. The size of the pool (on average per query) for various pool depths if the pooling is performed TREC-style. Here n is the number of input systems in the given data set. The full TREC pool corresponds to depth 100 in principle though in practice the actual TREC pool may be somewhat smaller or larger.

most widely reported standard measures (Buckley & Voorhees, 2004), our methodology, by statistical design, produces *unbiased* estimates of the *standard measures* of retrieval performance themselves.

The core of our methodology is the derivation, for each measure, of a distribution over documents such that the *value* of the measure corresponds to the *expectation* of observing a relevant document drawn according to that distribution. (In the case of average precision, the distribution is over pairs of documents, and the observation is the product of the relevances for the pair drawn.) Given such distributions, one can estimate the expectations (and hence measurement values) using random sampling. We further show how a sample drawn according to one such distribution can be used to properly estimate the expectations with respect to other distributions; thus, we effectively show how a single sample can be used to estimate *multiple* performance measures for a given list. Finally, we show how to generalize our sampling technique so that a single sample can be used to estimate multiple performance measures over *multiple lists* simultaneously, thus providing an efficient alternative to TREC-style evaluations.

We tested our extended methodology using TREC data, and the results we obtained were uniformly excellent across all TREC data sets examined; representative results are reported for the TREC 7, 8, and 10 data sets. While detailed results are described later in the text, Figure 1 provides a preview of these results. On the left is a scatter plot showing actual TREC mean average precision values for the systems in TREC8 vs. the inferred MAP values by using depth 20 pooling (the equivalent of 494 judgments per query on average, approximately 22% of the full TREC pool); on the right is a scatter plot showing actual TREC MAP values for the systems in TREC8 vs. the inferred MAP values by using our sampling technique. Note the *bias* inherent in TREC-style depth-pooling in contrast to the unbiased results obtained from random sampling with an equivalent total number of judgments. Further note that the *variance* in the errors is also reduced.

In the sections that follow, we begin by describing our core methodology as well as the extensions necessary for the efficient evaluation of multiple measures across multiple lists from a single judged pool. We conclude by describing the results of experiments run with the benchmark TREC data collection.

2. Methodology

In this section, we describe our methodology in detail. While many of the details are somewhat complex and/or omitted for space considerations, the basic idea can be summarized in the following sequence of steps. (1) For each measure, we derive a *random variable* and associated *probability distribution* such that the value of the measure in question is the *expectation* of the random variable with respect to the probability distribution. For example, to estimate precision-at-cutoff 500, one could simply uniformly sample documents from the top 500 in a given list and output the fraction of relevant documents seen. Thus, the underlying random variable for precision-at-cutoff c is dictated by the binary relevance assessments, and the associated distribution is uniform over the top c documents. (Since R-precision is effectively precision-at-cutoff R, an identical strategy holds.) For average precision, the situation is somewhat more complex. We show that the required sampling distribution is over *pairs* of documents and the underlying random variable is the product of the binary relevance judgments for that pair. (2) Given that the value of a measure can be viewed as the expectation of a random variable, one can apply standard sampling techniques to estimate this expectation and hence the value of the measure. However, a naive implementation of such a methodology would be relatively inefficient: separate i.i.d. samples would need to be drawn (and evaluated) for each measure and each list. For efficiency purposes, our goal is to draw a *single* sample according to a carefully chosen distribution over the documents, judge those documents, and then evaluate the various measures for the various lists given this single judged pool. As such, one is confronted with the task of estimating the expectation of a random variable with respect to a known distribution by using a sample drawn according to a different (but known) distribution. We introduce *scaling fac-*

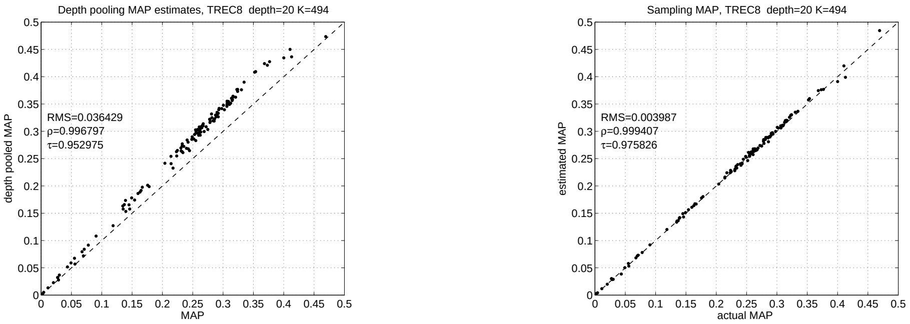

Figure 1. Comparing the performance of the TREC8 depth 20 pool vs. a sampling pool of an equivalent size.

tors to accomplish this task where these scaling factors are essentially the ratio between the desired and sampling distributions. (3) Finally, while our estimators are unbiased by design, it is also important that they have low variance so that their empirical means will converge to their true expectations quickly, i.e., with relatively few samples. We discuss conditions under which our estimators will have low variance, and we describe a heuristic for reducing the variance of our estimators. In the sections that follow, we describe each of these steps in more detail.

2.1. Average precision as an expected value

While our goal is to simultaneously estimate multiple measures of performance over multiple lists, we begin by considering the problem of estimating average precision from a random sample. Unlike R-precision or precision at standard cutoffs, deriving a sampling distribution for average precision is non-trivial, and it yields a distribution which empirically is quite useful for estimating the other measures of interest.

One can compute average precision as follows, where Z is the length of the retrieved list, *rel*(i) is the binary relevance of the document at rank i, and R is the number of relevant documents for the query.

$$\begin{aligned} AP &= \frac{1}{R} \cdot \sum_{i\,:\,rel(i)=1} PC(i) \\ &= \frac{1}{R} \cdot \sum_{i=1}^{Z} rel(i) \cdot PC(i) \\ &= \frac{1}{R} \cdot \sum_{i=1}^{Z} rel(i) \sum_{j=1}^{i} rel(j)/i \\ &= \frac{1}{R} \cdot \sum_{1 \le j \le i \le Z} \frac{1}{i} \cdot rel(i) \cdot rel(j) \end{aligned}$$

	1	2	3	$\ldots$	Z
1	1				
2	1/2	1/2			
3	1/3	1/3	1/3		
$\vdots$					
Z	1/Z	1/Z	1/Z	$\ldots$	1/Z

	1	2	3	$\ldots$	Z
1	2	1/2	1/3	$\ldots$	1/Z
2	1/2	1	1/3	$\ldots$	1/Z
3	1/3	1/3	2/3	$\ldots$	1/Z
$\vdots$					
Z	1/Z	1/Z	1/Z	$\ldots$	2/Z

Table 2. (Left) Weights associated with pairs of ranks; normalizing by Z yields an asymmetric joint distribution. (Right) Symmetric weights; normalizing by $2Z$ yields the symmetric joint distribution *JD*.

Thus, in order to evaluate $R \cdot AP$, one must compute the weighted product of relevances of documents at pairs of ranks, where for any pair $j \le i$, the associated weight is $1/i$. (See Table 2, left.) In order to view this sum as an expectation, we define an *event space* corresponding to pairs of ranks (i, j), a *random variable* X corresponding to the product of the binary relevances *rel*$(i) \cdot$ *rel*(j), and an appropriate *probability distribution* over the event space. One such distribution corresponds to the (appropriately normalized) weights given in Table 2 (left); for convenience, we shall instead define a symmetrized version of these weights (see Table 2 (right)) and the corresponding joint distribution *JD* (appropriately normalized by $2Z$). It is not difficult to see that

$$R \cdot AP = Z \cdot \mathbf{E}[X]$$

where the expectation is computed with respect to either distribution. Thus, if U is a multiset of pairs drawn according to *JD*, we obtain the following estimate for AP

$$\widehat{AP} = \frac{Z}{R} \cdot \frac{1}{|U|} \sum_{(i,j) \in U} rel(i) \cdot rel(j).$$

2.2. Efficiency considerations

The technique described above can be used to estimate the average precision of a single retrieval system with respect to any given query. However, it is relatively inefficient: (1) On a per system basis, i.i.d. pairs of documents are drawn and judged, but the induced pairs of judged documents across i.i.d. samples are not used. Furthermore, (2) one is often faced with the task of evaluating the average precisions of *many* retrieval systems with respect to a given query (as in TREC), and in a naive implementation of the technique described, the documents judged for one system will not necessarily be reused in judging another system. In contrast, TREC creates a single pool of documents from the collection of systems to be evaluated, judges that pool, and evaluates all of the systems with respect to this single judged pool. In order to combat the inefficiency inherent in (1), we shall instead draw a sample from a distribution over *documents* and consider all *induced pairs* of judgments. In order to combat the inefficiency inherent in (2), we shall construct a *single* distribution over documents where this distribution is derived from the joint distributions JD_s associated with every system s.

In both cases, we shall effectively be sampling from a distribution different from the one necessary to estimate the expectations desired. To combat this, we introduce *scaling factors* as follows. Let $D(i,j)$ be the joint distribution over documents from which we effectively sample. Note that i and j now denote documents, not ranks. Similarly abusing notation, let $JD_s(i,j)$ denote the joint distribution over documents (not ranks) required to estimate the proper expectation for system s. We define *scaling factors* $SF_s(i,j)$ which correspond to the ratio between the desired and sampling distributions

$$SF_s(i,j) = \frac{JD_s(i,j)}{D_s(i,j)}$$

where D_s is the distribution induced by D over documents retrieved by s. We then have

$$\widehat{AP} = \frac{Z_s}{R} \cdot \frac{1}{|U_s|} \sum_{(i,j)\in U_s} rel(i) \cdot rel(j) \cdot SF_s(i,j)$$

where Z_s is the length of the list returned by system s and $U_s \subseteq U$ is the subset of samples corresponding to documents retrieved by s. Note that the above formulation holds for any sampling distribution D. In what follows, we describe a heuristic for determining a *good* sampling distribution—one which corresponds to a distribution over *documents* (for efficiency) and which explicitly attempts to minimize the *variance* in the estimates produced (for accuracy).

2.3. Finding the best sampling distribution

In determining a sampling distribution D, we consider two factors. First, we impose the condition that D be a symmetric product distribution, i.e., $D(i,j) = M(i)\cdot M(j)$ for some (marginal) distribution over documents M. The purpose for this is efficiency: we will sample documents according to M and consider all induced pairs of documents, which will be distributed (approximately) according to D. Second, we seek a D which explicitly attempts to minimize the variance in our estimator, for accuracy. We begin by considering the latter factor.

Variance minimization. For a sampling distribution D and a given system s, let D_s be the distribution induced by D over pairs of documents contained in the list returned by system s. Furthermore, let Y be the random variable $rel(i) \cdot rel(j) \cdot SF_s(i,j)$ such that $AP = (Z_s/R)\cdot \mathbf{E}_{D_s}[Y]$. Since Z_s and R are fixed, in order to minimize the variance of AP, we must minimize the variance of Y.

$$\begin{aligned}
&\mathbf{Var}[Y] \\
&\quad = \mathbf{E}[Y^2] - \mathbf{E}^2[Y] \\
&\quad = \sum_{i,j} D_s(i,j)\cdot rel(i)^2 \cdot rel(j)^2 \cdot SF_s(i,j)^2 \\
&\qquad -(AP\cdot R/Z_s)^2 \\
&\quad = \sum_{i,j:rel(i)=rel(j)=1} D_s(i,j)\cdot \frac{JD_s(i,j)^2}{D_s(i,j)^2} \\
&\qquad -(AP\cdot R/Z_s)^2 \\
&\quad = \sum_{i,j:rel(i)=rel(j)=1} \frac{JD_s(i,j)^2}{D_s(i,j)} - (AP\cdot R/Z_s)^2
\end{aligned}$$

To minimize this variance, it is enough to minimize the first term since $AP \cdot R/Z_s$ is fixed. Employing ideas similar to *importance sampling* for minimizing the variance of Monte Carlo estimators (Anderson, 1999), we find that the best sampling distribution D is the distribution induced by JD_s over relevant documents. (Details omitted for space considerations.) Of course, we do not have the complete relevance judgments necessary to calculate the ideal sampling distribution. However, the marginal distribution $MD_s(i) = \sum_j JD_s(i,j)$ associated with the average precision sampling distribution $JD_s(i,j)$ has been shown to be a reasonable prior for relevant documents (Aslam et al., 2005a), and using such a prior one can argue that a sampling distribution $D_s(i,j)$ proportional[2] to $(MD_s(i)\cdot MD_s(j))^{3/2}$ is likely to result

[2] The expression must be normalized to form a distribution.

in low variance. (Again, details omitted for space considerations.) $D_s(i,j)$ is a product distribution having identical marginals with respect to i and j; let $MD'_s(i)$ be the marginal associated with $D_s(i,j)$.

If our task were to estimate the performance of only one retrieval system, we could sample documents according to $MD'_s(i)$, consider all induced pairs of documents, and estimate AP using appropriate scaling factors. However, in general our task is to simultaneously estimate AP for N systems from a single sample. We obtain a final sampling marginal $M(i)$ by averaging the marginals associated with each system s.

$$M(i) = \frac{1}{N}\sum_s MD'_s(i)$$

We finally note that in a typical TREC setting, one averages AP over 50 queries to obtain a final estimate of the performance of a system, and this averaging results in a further significant variance reduction.

Exact computation of scaling factors. $M(i)$ is the distribution we use for sampling documents, and given a sample of K such documents, we consider all K^2 induced pairs and estimate the required expectations from these induced pairs and appropriate scaling factors. For sufficiently large K, the distribution over induced pairs will approximate the associated product distribution $D(i,j) = M(i) \cdot M(j)$; however, the actual distribution is multinomial. One can show that the actual induced pairs distribution $I(i,j)$ is given by

$$I(i,j) = \frac{K-1}{K} \cdot M(i) \cdot M(j)$$

when $i \neq j$ and

$$I(i,i) = \frac{1}{K} M(i)\left(1 + (K-1)M(i) + (1 - M(i))^{K-1}\right)$$

otherwise. As a consequence, if K_s is the size of the subset of K sampled documents which are retrieved by system s, one obtains the following final scaling factors:

$$SF_s(i,j) = \frac{JD_s(i,j)}{K^2 I(i,j)/{K_s}^2}.$$

(Details omitted for space considerations.)

2.4. Estimating R

To obtain an estimate for AP, we must know or obtain estimates for R, Z_s, and the expectation described above. We have described in detail how to estimate the expectation, and Z_s is a known quantity (the length of the system's returned list). However, R, the total number of documents relevant to the given query, is not typically known and must also be estimated. Sophisticated approaches for estimating R exist (Kantor et al., 1999); however, in this preliminary study we employ techniques similar to those described above. In order to estimate R (as calculated by TREC), one could simply uniformly sample documents from the depth 100 pool. Given that our sample is drawn according to $M(i)$ instead, one can employ appropriate scaling factors to obtain the correct estimate.

2.5. Estimating $PC(c)$

To estimate precision-at-cutoff c, one could simply uniformly sample documents from the top c in any given list. Given that we sample according to $M(i)$, we again employ appropriate scaling factors to obtain correct estimates for $PC(c)$.

2.6. Estimating RP

R-precision is simply the precision-at-cutoff R. We do not know R; however, we can obtain an estimate $\widehat{R}$ for R as described above. Given this estimate, we simply estimate $PC(\widehat{R})$ as described above.

3. Results

We tested the proposed sampling method as a mechanism for estimating the performance of retrieval systems using data from TRECs 7, 8 and 10. We used mean average precision (MAP), mean R-precision (MRP), and mean precision at cutoffs 5, 10, 15, 20, 30, 100, 200, 500, and 1000 (MPC(c)) as evaluation measures. We compared the estimates obtained by the sampling method with the "actual" evaluations, i.e. evaluations obtained by depth 100 TREC-style pooling. The estimates are found to be consistently good even when the total number of documents judged is far less than the number of judgments used to calculate the actual evaluations.

To evaluate the quality of our estimates, we calculated three different statistics, root mean squared (RMS) error, linear correlation coefficient ρ (Wackerly et al., 2002), and Kendall's τ (Stuart, 1983).

RMS error measures the deviation of estimated values from actual values. Hence, it is related to the standard deviation of the estimation error. Let $(a_1, a_2, ..., a_N)$ be the actual values and $(e_1, e_2, ..., e_N)$ be the estimated values. The RMS error of the estimation can be calculated as

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(a_i - e_i)^2}$$

The linear correlation coefficient evaluates how well the actual and estimated values are correlated. This correlation is measured based on how well the estimated and actual values fit to a straight line.

We can convert the values of measures to a ranking of systems by sorting the systems in descending order according to the measures. Kendall's τ evaluates how well the estimated measures rank the systems compared to the actual rankings. It is a function of the minimum number of pairwise adjacent interchanges needed to convert one ranking into the other. Both ρ and Kendall's τ values range from -1 (perfectly negatively correlated values) to $+1$ (perfectly correlated values).

Note that in contrast to the RMS error, Kendall's τ and ρ does not measure how much the estimated values differ from the actual values. Therefore, even if they indicate perfectly correlated estimated and actual values, the estimates may still not be accurate. Hence, it is much harder to achieve small RMS errors than to achieve high τ or ρ values. Because of this, we mainly focus on the RMS error values when evaluating the performance of the sampling method.

In order to show how the sampling method compares with the TREC-style pooling method, we run two different sets of experiments on TREC8 data for each measure (MAP, MRP and MPC(100)[3]). In the first set, we compare the estimates of the measures obtained using TREC-style depth pooling with the actual values for depths 1, 3, and 10. TREC-style depth pooling for depths 1, 3, and 10 correspond to 40, 95, and 260 relevance judgments on average per query, respectively. In the second set of experiments, we compare the estimated values of the measures obtained using the sampling method with the actual values of the measures using the same total numbers of judgments obtained from the first set of experiments (i.e., 40, 95, and 260 judgments). Since the performance of the sampling method varies depending on the actual sample, we sampled 10 times and picked a representative sample that exhibited typical performance based on the three evaluation statistics used.

We report the results of the experiments for MAP, MRP, and MPC(100) in Figure 2, Figure 3, and Figure 4, respectively. The results of the first and second set of experiments are illustrated in the upper three and lower three plots of each corresponding figure. As can be seen, for all three depths there is a significant improvement in all three statistics when sampling is used versus the TREC-style pooling for all the measures. As illustrated in the figures, the sampling estimates have reduced variance and little or no bias compared to depth pooling estimates. This can be seen from the great reduction in the RMS error when the estimates are obtained via sampling. Furthermore, the bottom-right plots of all three figures show that with as few as 260 relevance judgments, the sampling method can very accurately estimate the actual measure values which were obtained using $86,830$ total relevance judgments (1,737 relevance judgments on average per query).

Figure 5 illustrates how MAP, MRP, and MPC(100) estimates using TREC-style depth pooling compare with those obtained using sampling as the depth of the pool changes. Since the RMS error is the most important of the three statistics used, the comparison in this figure is based on RMS error. In this set of experiments, for depths 1 to 10, we first calculated the number of documents required to be judged using TREC-style depth pooling. Then, for each depth, we formed 10 different samples of the same size as the required judgment set for each corresponding depth and calculated the average RMS error values over the 10 samples for each measure. The leftmost, middle, and rightmost columns in Figure 5 show the average RMS error for sampling versus the RMS error using TREC-style depth pooling for the measures MAP, MRP, and MPC(100), respectively. Plots in rows 1 to 3 corresponds to the results obtained from the TREC 7, 8 and 10 datasets, respectively. As can be seen in the figure, for all TRECs the sampling method significantly outperforms the TREC-style depth pooling method for all three measures at all depths.

4. Conclusions and Future Work

We proposed a sampling technique for efficiently estimating standard measures of retrieval performance using incomplete judgments. The proposed method can also be used as part of a strategy for estimating the labels of unjudged documents with respect to a query. Given values for average precision and R, one can accurately infer probabilities of relevance for documents via a maximum entropy technique (Aslam et al., 2005b). Given that one can obtain highly accurate estimates of average precision and R from a small number of judged documents using the proposed sampling method, and given that one can then use these estimates to infer accurate estimates for the relevance of each document via the maximum entropy method, one can in principle obtain accurate (probabilistic) relevance assessments for large document collections from

[3] Since sampling has similar performance at all cutoff levels, we only report the results for cutoff 100 due to space limitations.

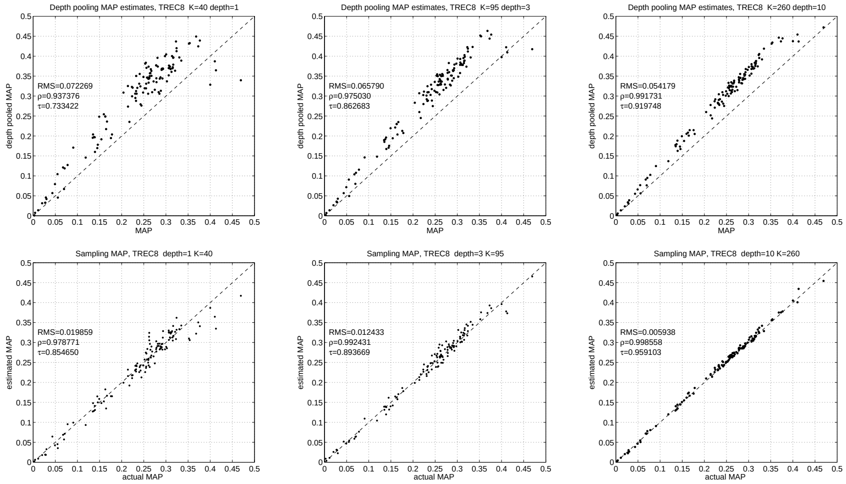

Figure 2. Comparison of correlations: sampling vs. depth pooling mean average precision in TREC8.

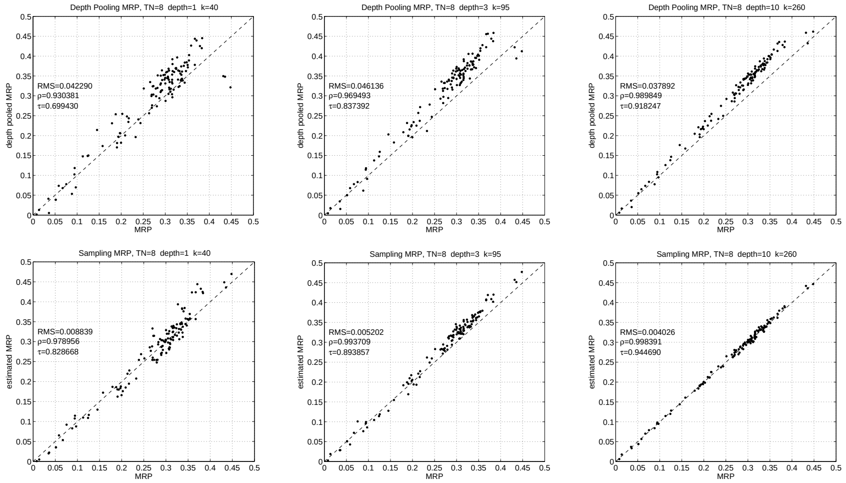

Figure 3. Comparison of correlations: sampling vs. depth pooling for mean R-precision in TREC8.

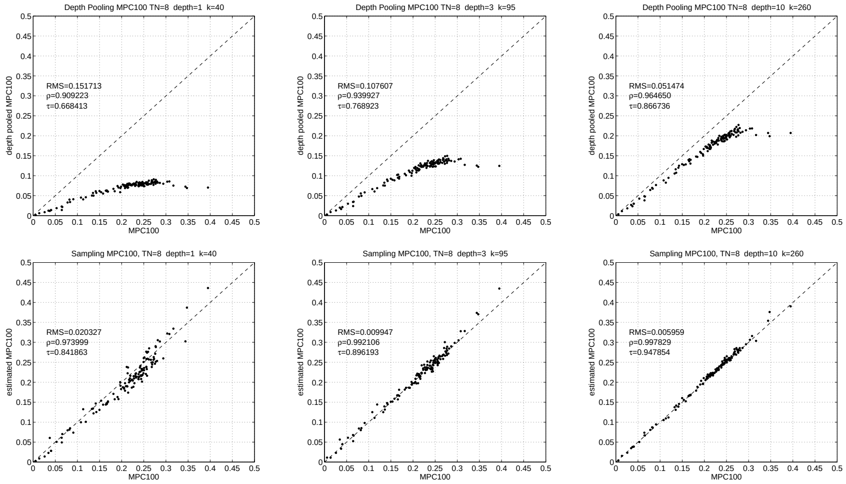

Figure 4. Comparison of correlations: sampling vs. depth pooling for mean precision at cutoff 100 measures in TREC8.

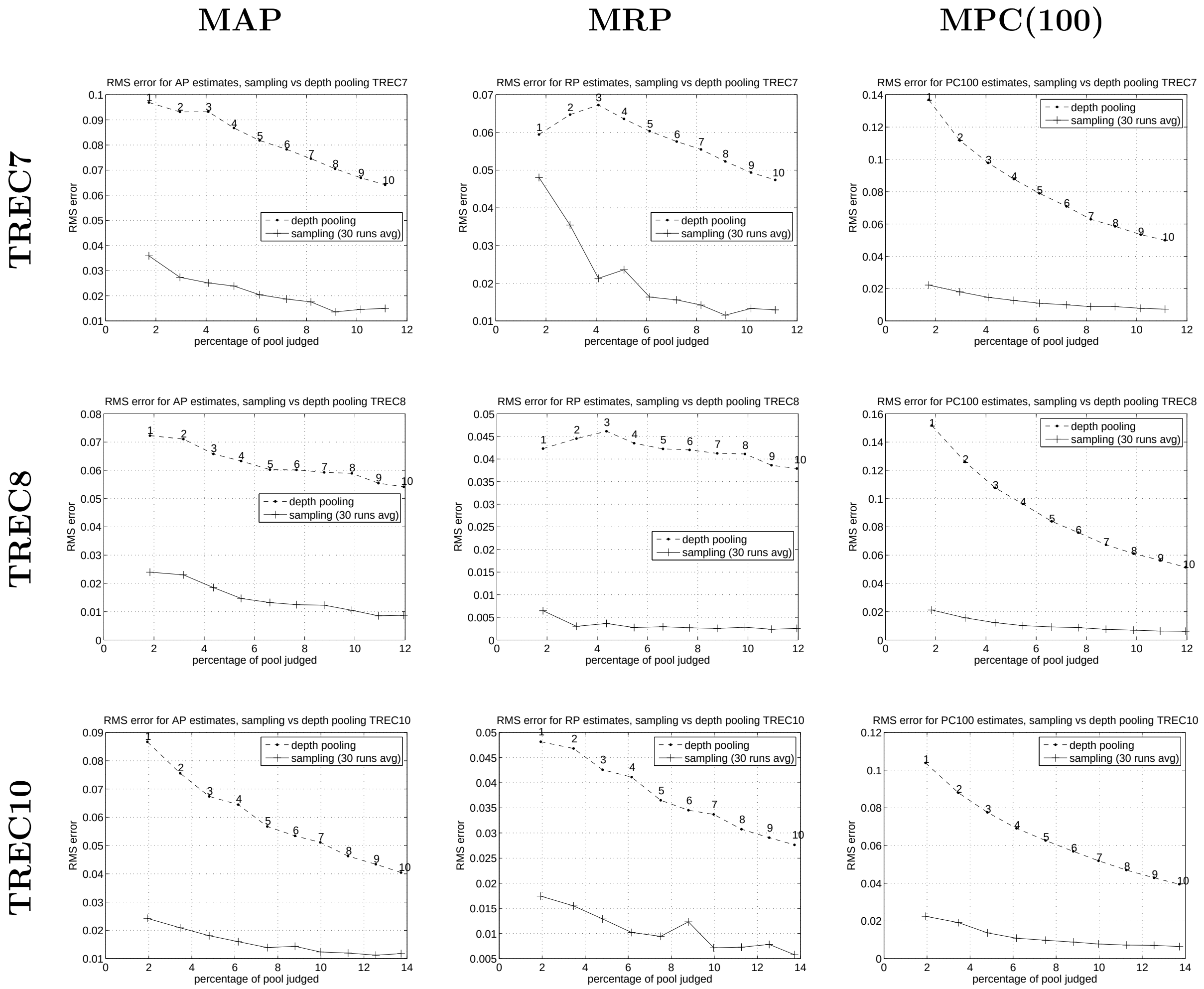

Figure 5. RMS error comparisons for mean average precision, mean R-precision, mean precision at cutoff 100 measures in TRECs 7, 8 and 10.

a small number of judgments.

As mentioned in Section 2.4, the estimation of R can be accomplished in a more sophisticated manner. We have examined our simple estimates, and we note that they are not as accurate as the estimates for the retrieval measures themselves (AP, RP, PC). As such, it is not at all clear that a better estimate for R would yield improved estimates for retrieval measures. In fact, the simultaneous estimate of the expectations described and R from a single sample seems to be "self-correcting:" when one is high (or low), the other is also high (or low), and thus the ratio is "preserved." Understanding this phenomenon will be important in order to derive provable bounds on performance (sample complexity, accuracy, etc.) for the method described.

Acknowledgments

We thank the anonymous reviewers for their insightful comments, and we gratefully acknowledge the support provided by the NSF (CCF-0418390).

References

Anderson, E. C. (1999). Monte carlo methods and importance sampling. Lecture Notes for Statistical Genetics.

Aslam, J. A., Pavlu, V., & Savell, R. (2003). A unified model for metasearch, pooling, and system evaluation. *Proceedings of the Twelfth International Conference on Information and Knowledge Management* (pp. 484–491). ACM Press.

Aslam, J. A., Pavlu, V., & Yilmaz, E. (2005a). Measure-based metasearch. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press. To appear.

Aslam, J. A., Yilmaz, E., & Pavlu, V. (2005b). The maximum entropy method for analyzing retrieval measures. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press. To appear.

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

Buckley, C., & Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. *Proceedings of the Twenty-seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 25–32). ACM Press.

Cormack, G. V., Palmer, C. R., & Clarke, C. L. A. (1998). Efficient construction of large test collections. In (Croft et al., 1998), 282–289.

Croft, W. B., Moffat, A., van Rijsbergen, C. J., Wilkinson, R., & Zobel, J. (Eds.). (1998). *Proceedings of the 21th annual international ACM SIGIR conference on research and development in information retrieval*. Melbourne, Australia: ACM Press, New York.

Harman, D. (1995). Overview of the third text REtreival conference (TREC-3). *Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 1–19). Gaithersburg, MD, USA: U.S. Government Printing Office, Washington D.C.

Kantor, P., Kim, M.-H., Ibraev, U., & Atasoy, K. (1999). Estimating the number of relevant documents in enormous collections. *Proceedings of tthe 62nd Annual Meeting of the American Sociaty for Information Science* (pp. 507 – 514).

Stuart, A. (1983). Kendall's tau. *Encyclopedia of Statistical Sciences* (pp. 367 – 369). John Wiley & Sons.

Voorhees, E., & Harman, D. (2000). Overview of the Eighth Text REtrieval Conference (TREC-8). *The Eighth Text REtrieval Conference (TREC-8)*. Gaithersburg, MD, USA: U.S. Government Printing Office, Washington D.C.

Wackerly, D. D., Mendenhall, W., & Scheaffer, R. L. (2002). *Mathematical statistics with applications*. Duxbury Advanced Series.

Zobel, J. (1998). How reliable are the results of large-scale retrieval experiments? In (Croft et al., 1998), 307–314.