# A Statistical Method for System Evaluation Using Incomplete Judgments

Javed A. Aslam[*] Virgil Pavlu Emine Yilmaz
College of Computer and Information Science
Northeastern University
360 Huntington Ave, #202 WVH
Boston, MA 02115
{jaa,vip,emine}@ccs.neu.edu

## ABSTRACT

We consider the problem of large-scale retrieval evaluation, and we propose a statistical method for evaluating retrieval systems using incomplete judgments. Unlike existing techniques that (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 statistical technique produces unbiased estimates of the standard measures themselves.

Our proposed technique is based on random sampling. While our estimates are unbiased by statistical design, their variance is dependent on the sampling distribution employed; as such, we derive a sampling distribution likely to yield low variance estimates. We test our proposed technique using benchmark TREC data, demonstrating that a sampling pool derived from a set of runs can be used to efficiently and effectively evaluate those runs. We further show that these sampling pools generalize well to unseen runs. Our experiments indicate that highly accurate estimates of standard performance measures can be obtained using a number of relevance judgments as small as 4% of the typical TREC-style judgment pool.

## Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *Performance evaluation*

## General Terms

Theory, Measurement, Experimentation

## Keywords

Evaluation, Sampling, Average Precision

[*]We gratefully acknowledge the support provided by NSF grant CCF-0418390.

## 1. INTRODUCTION

We consider the problem of large-scale retrieval evaluation and propose a statistical technique for efficiently and effectively estimating standard measures of retrieval performance via random sampling. Standard methods of retrieval evaluation can be quite expensive when conducted on a large-scale. For example, in many years of the annual Text REtrieval Conference (TREC), upwards of 100 runs each consisting of a ranked list of up to 1,000 documents were submitted with respect to each of 50 or more topics. In principle, each document in the collection would need to be assessed for relevance with respect to each query in order to evaluate many standard retrieval measures such as average precision and R-precision; in practice, this is prohibitively expensive. TREC instead employs *depth pooling* wherein the union of the top $k$ documents retrieved in each run corresponding to a given query is formed, and the documents in this *depth* $k$ pool are judged for relevance with respect to this query. In TREC, $k = 100$ has been shown to be an effective cutoff in evaluating the relative performance of retrieval systems [**?**, **?**], and while the depth 100 pool is considerably smaller than the document collection, it still engenders a large assessment effort: in TREC8, for example, 86,830 relevance judgments were used to assess the quality of the retrieved lists corresponding to 129 system runs in response to 50 topics [**?**]. The table given below shows the relationship between pool depth and the number of judgments required per topic on average for various TRECs.

| Pool | TREC | | |
|---|---|---|---|
| Depth | 7 | 8 | 10 |
| 1 | 27 | 29 | 25 |
| 3 | 65 | 71 | 64 |
| 5 | 98 | 110 | 100 |
| 10 | 179 | 200 | 184 |
| 20 | 337 | 379 | 339 |
| 100 | 1596 | 1712 | 1414 |

Shallower pools [**?**] and greedily chosen dynamic pools [**?**, **?**] have also been studied in an attempt to alleviate the assessment effort; however, such techniques tend to produce biased estimates of standard retrieval measures, especially when relatively few relevance judgments are used. Recently, Buckley and Voorhees proposed a new measure, *bpref*, and they show that when bpref is employed with a judged sample drawn uniformly from the depth 100 pool, the retrieval systems can be effectively ranked [**?**]. However, bpref is not designed to approximate any standard measure of retrieval performance, and while bpref is correlated with average precision, especially at high sampling rates, these correlations and the system rankings produced degenerate at low sampling rates.

Our goal in this work is to efficiently and accurately estimate standard measures of retrieval performance. Unlike previously proposed methodologies which tend to produce biased estimates of standard measures using few relevance judgments or methodologies based on estimating measures other than the most widely reported standard measure, 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 is proportional 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. By statistical design, such estimates will be *unbiased*. Furthermore, through the use of the statistical estimation technique of *importance sampling* [**?**], we show how *low variance* estimates of multiple retrieval measures can be simultaneously estimated for multiple runs given a single sample. In sum, we show how both *efficient* and *effective* estimates of standard retrieval measures can be inferred from a random sample, thus providing an alternative to large-scale TREC-style evaluations.

We tested our methodology using the benchmark TREC data collections, and results are reported for TRECs 7, 8, and 10. We compare the performance of TREC-style depth pools to sampling pools of an equivalent size. Generally speaking, as the number of relevance assessments is decreased (either by considering a lower depth pool, in the case of TREC, or by sampling at a lower rate, in the case of the proposed technique), the bias and variance of the estimates inferred from TREC-style depth pools increases, while only the variance of estimates inferred from sampling pools increases. At equivalent levels of judgment effort, the estimates produced by sampling are consistently better than those produced by depth pooling, often dramatically so.

While TREC-style depth pools are used to evaluate the systems from which they were generated, they are also often used to evaluate new runs which did not originally contribute to the pool. Depth 100 TREC-style pools generalize well in the sense that they can be used to effectively evaluate new runs, and this has been a tremendous boon to researchers who use TREC data. We also show that random sampling pools generalize well, achieving estimation errors on unseen runs comparable to the estimation errors on runs from which the sample was drawn.

In the sections that follow, we describe our methodology and the results from experiments using the TREC collections. We conclude with a summary and directions for future research.

## 2. METHODOLOGY

In this section, we sketch our proposed statistical technique for efficiently and effectively estimating standard retrieval measures from random samples.[1] While many of the details are somewhat complex and/or necessarily omitted for space considerations, the basic ideas can be summarized as follows:

1. For each measure, we derive a *random variable* and associated *probability distribution* such that the value of the measure in question is proportional to 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 precision-at-cutoff $R$, an identical strategy holds, given the value or an estimate of $R$.) 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. To implement this methodology *efficiently*, one would like to estimate all retrieval measures for all runs simultaneously using a single judged sample. 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 employ *scaling factors* used in the statistical estimation field of *importance sampling* [**?**] in order to facilitate this goal.

3. Finally, to implement this methodology *effectively*, one desires low variance unbiased estimators so that the computed empirical means will converge to their true expectations quickly. While we show that any known sampling distribution can be used to yield unbiased estimators for retrieval measures, we also describe a heuristic for generating a specific sampling distribution which is likely to yield low variance estimators.

In the sections that follow, we describe each of these ideas in more detail, and we then present extensive experiments using TREC data.

### 2.1 Sum 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 is empirically quite useful for estimating the other measures of interest.

By definition, average precision ($AP$) is the average of the precisions at all relevant documents, or equivalently, the

[1]A somewhat different treatment of this material may be found in our recent workshop work-in-progress report [**?**].

|  | 1 | 2 | 3 | … | $Z$ |
|---|---|---|---|---|---|
| 1 | 1 |  |  |  |  |
| 2 | 1/2 | 1/2 |  |  |  |
| 3 | 1/3 | 1/3 | 1/3 |  |  |
| ⋮ |  |  |  |  |  |
| $Z$ | 1/Z | 1/Z | 1/Z | … | 1/Z |

|  | 1 | 2 | 3 | … | $Z$ |
|---|---|---|---|---|---|
| 1 | 2 | 1/2 | 1/3 | … | 1/Z |
| 2 | 1/2 | 1 | 1/3 | … | 1/Z |
| 3 | 1/3 | 1/3 | 2/3 | … | 1/Z |
| ⋮ |  |  |  |  |  |
| $Z$ | 1/Z | 1/Z | 1/Z | … | 2/Z |

**Table 1: (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 $JD$.**

sum of the precisions at all relevant documents divided by $R$, the number of documents relevant to the query. Let $SP$ be this sum of precisions at all relevant documents. In what follows, we first discuss estimating $SP$, and later we discuss estimating $R$; our estimate of $AP$ will be the ratio of the estimates of $SP$ and $R$.

One can compute sum 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 documents relevant to the query.

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

Thus, in order to evaluate $SP$, 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 **??**, 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 **??** (left); for convenience, we shall instead define a symmetrized version of these weights (see Table **??** (right)) and the corresponding joint distribution $JD$ (appropriately normalized by $2Z$). It is not difficult to see that

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

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 $SP$

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

## 2.2 Simultaneous estimation for multiple runs

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 runs to be evaluated, judges that pool, and evaluates all of the systems with respect to this single judged

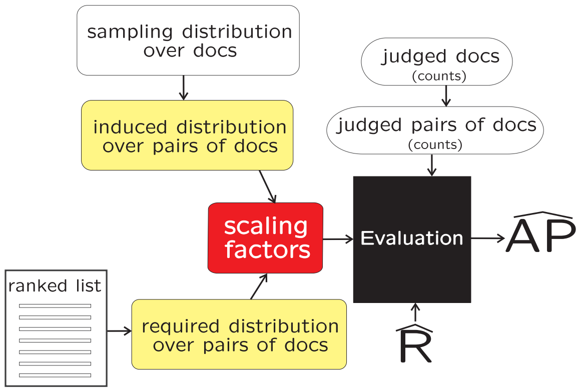

sampling distribution over docs
judged docs (counts)
judged pairs of docs (counts)
induced distribution over pairs of docs
scaling factors
Evaluation
$\widehat{AP}$
ranked list
required distribution over pairs of docs
$\widehat{R}$

**Figure 1: AP evaluation diagram**

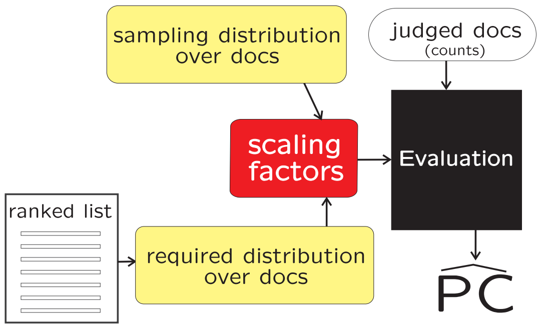


**Figure 2: PC evaluation diagram**

pool. In order to combat this potential inefficiency, we shall construct a *single* distribution over pairs of documents derived from the joint distributions $JD_s$ associated with every system $s$.

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

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

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 Deriving the sampling distribution

The technique described above can be used to estimate the average precision of one or more retrieval runs with respect to any given query. However, it is relatively inefficient: (1) On a per run basis, independent and identically distributed (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. In order to combat this potential inefficiency, we shall instead draw a sample from a distribution over *documents* and consider all *induced pairs* of judgments.

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 $SP = Z_s \cdot \mathbf{E}_{D_s}[Y]$. Since $Z_s$ and $R$ are fixed, in order to minimize the variance of $AP$, we must minimize the variance of $Y$.

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

To minimize this variance, it is enough to minimize the first term since $SP/Z_s$ is fixed. Employing *importance sampling* techniques for minimizing the variance of Monte Carlo estimators, one can derive that the best sampling distribution $D$ is the distribution induced by $JD_s$ over relevant documents. (See Anderson [**?**] for an example of such a derivation.) 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 [**?**], 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 in low variance. (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 $s$.

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

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

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.

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:

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

Finally, we derive the exact form of the multinomial sampling distribution over induced pairs. Sampling $K$ documents (with replacement) from a distribution $M$ and forming all $K^2$ pairs of documents yields a *multinomial distribution* $I$ over the possible outcomes. Let $\vec{k} = (k_1, k_2, \ldots, k_W)$ correspond to counts for the sampled documents where $k_1 + k_2 + \cdots + k_W = K$ and $k_i$ is the count associated with document $i$. Then $\Pr(\vec{k}) = \binom{K}{k_1, k_2 \ldots k_W} \cdot \prod_d M(d)^{k_d}$.

For $i \neq j$ and $k > 2$, the induced pairs distribution is derived as follows

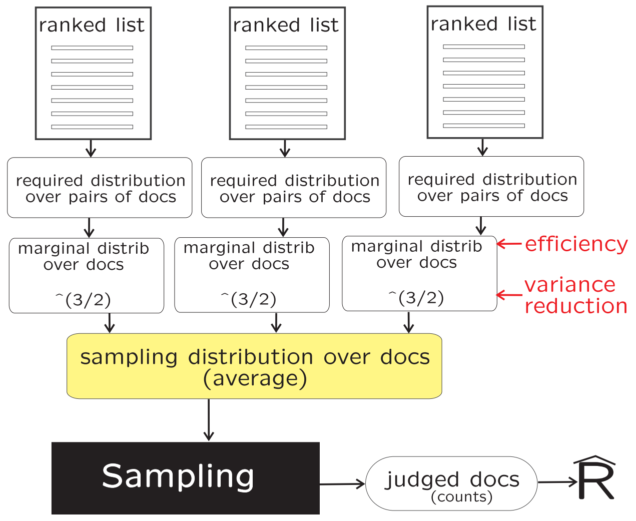
$$\begin{aligned}
I(i,j) &= \sum_{\vec{k}=(\ldots,k_i,\ldots,k_j,\ldots)} \tfrac{k_i \cdot k_j}{K^2} \cdot \Pr(\vec{k}) \\
&= \tfrac{1}{K^2} \sum_{k_i+k_j \le K} \Big( k_i k_j \tbinom{K}{k_i, k_j, \ldots} M(i)^{k_i} M(j)^{k_j} \cdot \\
&\quad (1 - M(i) - M(j))^{K-k_i-k_j} \Big) \\
&= \tfrac{1}{K^2} \sum_{k_i+k_j \le K} \Big( \tfrac{(K-2)!(K-1)K}{(k_i-1)!(k_j-1)!(K-k_i-k_j)!} M(i)M(j) \cdot \\
&\quad M(i)^{k_i-1} M(j)^{k_j-1} (1 - M(i) - M(j))^{K-k_i-k_j} \Big) \\
&= \tfrac{K(K-1)M(i)M(j)}{K^2} \sum_{k_i+k_j \le K} \Big( \tbinom{K-2}{k_i-1, k_j-1, \ldots} \cdot \\
&\quad M(i)^{k_i-1} M(j)^{k_j-1} (1 - M(i) - M(j))^{K-k_i-k_j} \Big) \\
&= \frac{K-1}{K} M(i)M(j)
\end{aligned}$$

When $i = j$, a similar derivation yields

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

## 2.4 Estimating R and AP

To obtain an estimate for $AP$, we must know or obtain estimates for $SP$, $R$ and $Z_s$, and the expectation described above. We have described in detail how to estimate $SP$, and $Z_s$ is a known quantity (the length of the system's returned list). However, $R$, the total number of documents



ranked list
ranked list
ranked list
required distribution over pairs of docs
required distribution over pairs of docs
required distribution over pairs of docs
marginal distrib over docs ^(3/2)
marginal distrib over docs ^(3/2)
marginal distrib over docs ^(3/2)
efficiency
variance reduction
sampling distribution over docs (average)
Sampling
judged docs (counts)
$\widehat{R}$

**Figure 3: Sampling diagram**

relevant to the given query, is not typically known and must also be estimated. Sophisticated approaches for estimating $R$ exist [?]; 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 and RP

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

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 $PC(\widehat{R})$.

## 2.6 Practical summary

Below we give a summary for implementing the sampling method. Figure **??** can serve as a guide.

(1) For each run $s$, use the joint distribution over pairs of documents, $JD_s(i,j)$, dictated by $SP$ such that sampling pairs of documents according to $JD$ would yield $SP$ in expectation.

(2) To minimize variance, we introduce a prior over relevant documents (Figure **??**, step 2). Let $MD_s(i)$ be the marginal distribution of $JD_s(i,j)$, and let $D_s(i,j)$ be the (appropriately normalized) joint distribution corresponding to $(MD_s(i) \cdot MD_s(j))^{3/2}$. Let $MD'_s(i)$ be the marginal distribution of $D$. This is the sampling distribution over documents that would be used for run $s$.

(3) Over all runs, compute $M(i)$, the average of these sampling distributions over documents. $M(i)$ is our single, final sampling distribution for the query (all runs).

(4) Sample $K$ documents with replacement according to $M(i)$ (Figure **??**, black box) until $T$ unique documents are drawn; judge these documents. ($T$ is the desired *a priori* judgment effort.) Generate all $K^2$ pairs of judged documents.

(5) Compute the multinomial induced pairs distribution (Figure **??**, step 1), $I(i,j)$; this is the "effective" distribution over pairs of documents from which we sampled. From $I(i,j)$ and $JD_s(i,j)$, compute the required scaling factors.

(6) From the induced pairs and the scaling factors, compute the estimates of $SP$ for each run (Figure **??** black box).

(7) Estimate $R$ using the sampled documents drawn according to $M(i)$ and appropriate scaling factors (Figure **??**).

(8) Estimate $AP$ by the ratio of the estimates for $SP$ and $R$. Note that the estimate of a ratio is not necessarily the ratio of estimates. More accurate ratio estimates derived via a second order Taylor series approximation [?] were tested, and they were generally found to be of little benefit for the computational effort required.

## 2.7 New retrieval runs

A question of particular importance is how can we use the samples generated by our method to evaluate a new run, i.e. a run that did not contribute to the sampling distribution. In order for sampling to work correctly, there should be sufficient sampled documents in the new run so that the evaluation using sampling is meaningful.

The evaluation (estimation) methodology is independent of the fact that the run participated to the sampling process; therefore it can be applied to the new runs in the same way as for the runs used in producing the sample. On the scale factor computation, the numerator is a function of the ranks of sampled documents in the new list and the denominator is computed based on the sampling distribution conditioned to the new run.

In TREC data, it is already the case that the actual pools are created from only a subset of the submitted runs. In all our experiments, to computing the average sampling distribution, we only use the runs that contributed to the pool (*training systems*) and use the runs that did not contribute to the pool as *testing systems*. In the plots that follow, the dots $(\cdot)$ in the plots correspond to the training systems, and the pluses $(+)$ correspond to the testing systems.

## 3. EXPERIMENTAL 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 (how different the estimated values are from the actual values, i.e. $RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(a_i - e_i)^2}$, where $a_i$ are the actual and $e_i$ are the estimates values), linear correlation coefficient $\rho$ (how well the actual and estimated values fit to a straight line), and Kendall's $\tau$ (how well the estimated measures rank the systems compared to the actual rankings).

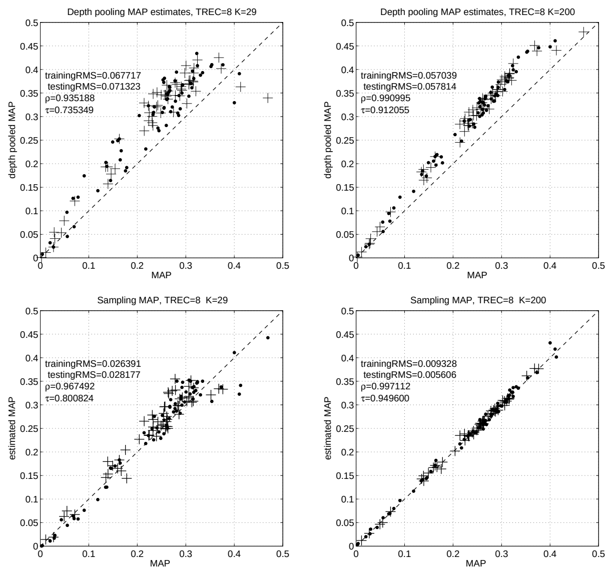


**Figure 4: Sampling vs. depth pooling mean average precision estimates at depths 1 and 10 in TREC8. Each dot (·) corresponds to a distribution-contributor run and each plus (+) to a distribution-non-contributor run (there are 129 runs in TREC8.)**

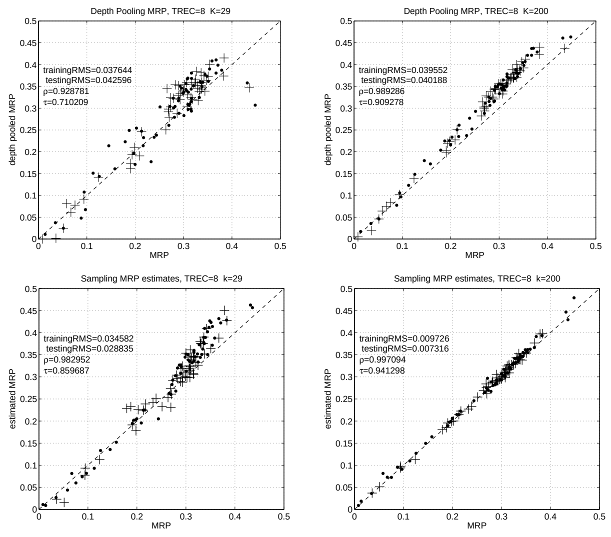


**Figure 5: Sampling vs. depth pooling mean R-precision estimates at depths 1 and 10 in TREC8.**

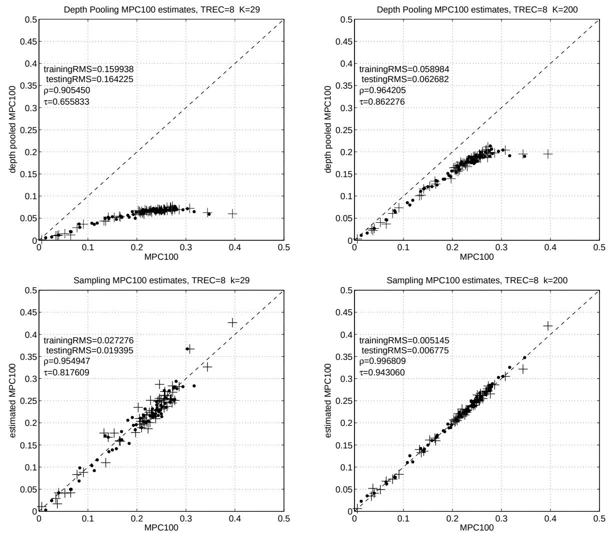


**Figure 6: Sampling vs. depth pooling mean prec at cutoff 100 estimates at depths 1 and 10 in TREC8.**

Note that in contrast to the RMS error, Kendall's $\tau$ and $\rho$ do 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 $\tau$ or $\rho$ values. Because of this, we mainly focus on the RMS error values when evaluating the performance of the sampling method.

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) on TREC8 in Figure **??**, Figure **??**, and Figure **??**, respectively. As can be seen, on TREC8 , for both depth=1 (on avg 29 judgments/query) and depth=10 (on avg 200 judgments/query), there is a significant improvement in all three statistics when sampling is used versus the TREC-style pooling for all the measures. 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 200 relevance judgments on average per query, the sampling method can very accurately estimate the actual measure values which were obtained using 1,737 relevance judgments on average per query.

Figure **??** illustrates how MAP estimates using TREC-style depth pooling compare in terms of $\rho$ and Kendall's $\tau$ with those obtained using sampling as the depth of the pool changes. 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 $\rho$ (left column) and $\tau$ (right column). As can be seen in the figure, for all TRECs the sampling method significantly outperforms the TREC-style depth pooling method at all depths.

For comparison purposes, we also include the average Kendall's $\tau$ value of bpref obtained using *random samples* [**?**] of the given size to the plots in the second column. The Kendall $\tau$ values for bpref are the average values computed over 10 different random sampling distributions.

## 3.1 Per query and per run results

While the MAP (and MRP and MPC) show improved performance over depth-style pooling in both mean and bias, there are certain situations when one needs the results of a single query, hence not taking advantage of the massive

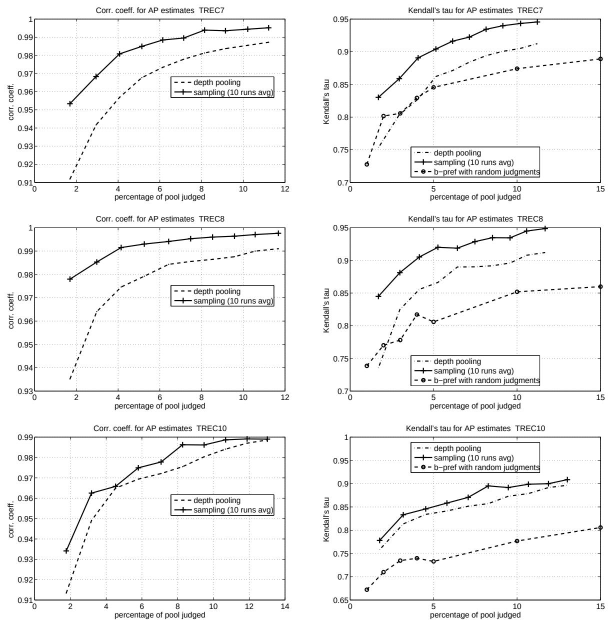


**Figure 7: Linear correlation coefficient and Kendall's $\tau$ error comparisons for mean average precision, in TRECs 7, 8 and 10.**

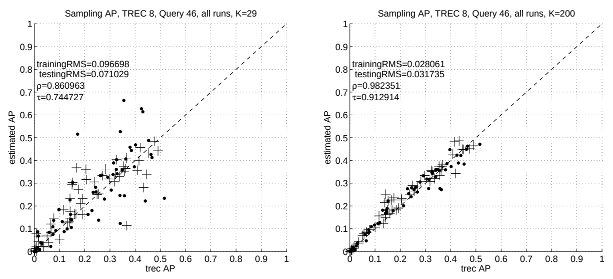


**Figure 8: Sampling estimates for a query with mixed system performance. Dots $(\cdot)$ represent training runs; pluses $(+)$ represent testing runs.**

.

variance reduction achieved by averaging over 50 queries. It is certainly not expected to see the same kind of performance on per query basis; however our results show definite usable query estimates (Figure **??**). The method described in this paper is self-contained for a query, i.e. estimates for a query are not dependant on any data from other queries.

On a different setup, one may want to analyze only one run over all queries (Figure **??**) Note that this setup is not self contained as the sampling method requires a set of runs (not only the one plotted) and associated judgments.

## 3.2 Generalization on new runs

It is important that the performance of sampling over the testing runs is virtually as good as the performance over the training runs. Note that the testing systems do not directly contribute to the sampling pool; their documents are sampled only because they happen to appear in training runs.

The trend of RMS error, as sample size increases from

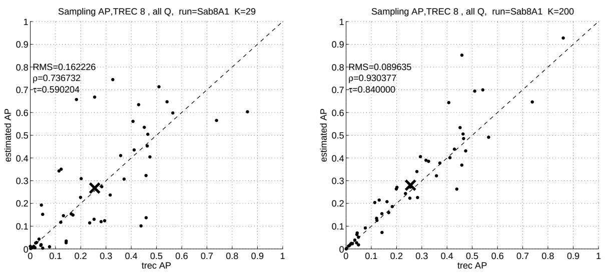


**Figure 9: Sampling estimates for a fixed typical run (Sab8A1) with MAP = 0.25, all queries. . Each dot $(\cdot)$ is an AP for a query estimate (total 50); MAP estimate is plotted as "$\times$".**

depth 1 to depth 10 equivalent for training and testing systems is shown in Figure **??**. On $x$-axis the units are the depth-pool equivalent number of judgments converted into percentages of depth-100 pool.

## 4. CONCLUSIONS AND FUTURE WORK

We propose a statistical technique for efficiently and effectively estimating standard measures of retrieval performance from random samples, and we demonstrate that highly accurate estimates of standard retrieval measures can be obtained from judged subsamples as small as 4% of the standard TREC-style depth 100 pool.

This work leaves open a number of question for further research: (1) In standard TREC settings, all documents in the depth 100 pool are judged and no documents outside the pool are judged. Our work indicates that *more* judging effort should be placed on documents near the top of ranked lists (they have high sampling probabilities) and *less* judging effort should be placed on documents near the bottom of ranked lists (they have low sampling probabilities). What is the optimal sampling distribution, and how does it change as a function of the collection or systems to be evaluated? Consider evaluating retrieval system on the web or with respect to the TREC Terabyte track, for example. (2) Given that our technique is based on random sampling, one could in principle derive high probability confidence intervals for the estimates obtained, and such confidence intervals would be quite useful in practice. (3) Finally, we have preliminary results which show that given accurate estimates of retrieval measures obtained from a judged subsample, one can accurately *infer* relevance assessments for the remaining *unjudged* documents. Such a technique would have obvious benefits in the production of large test collections.

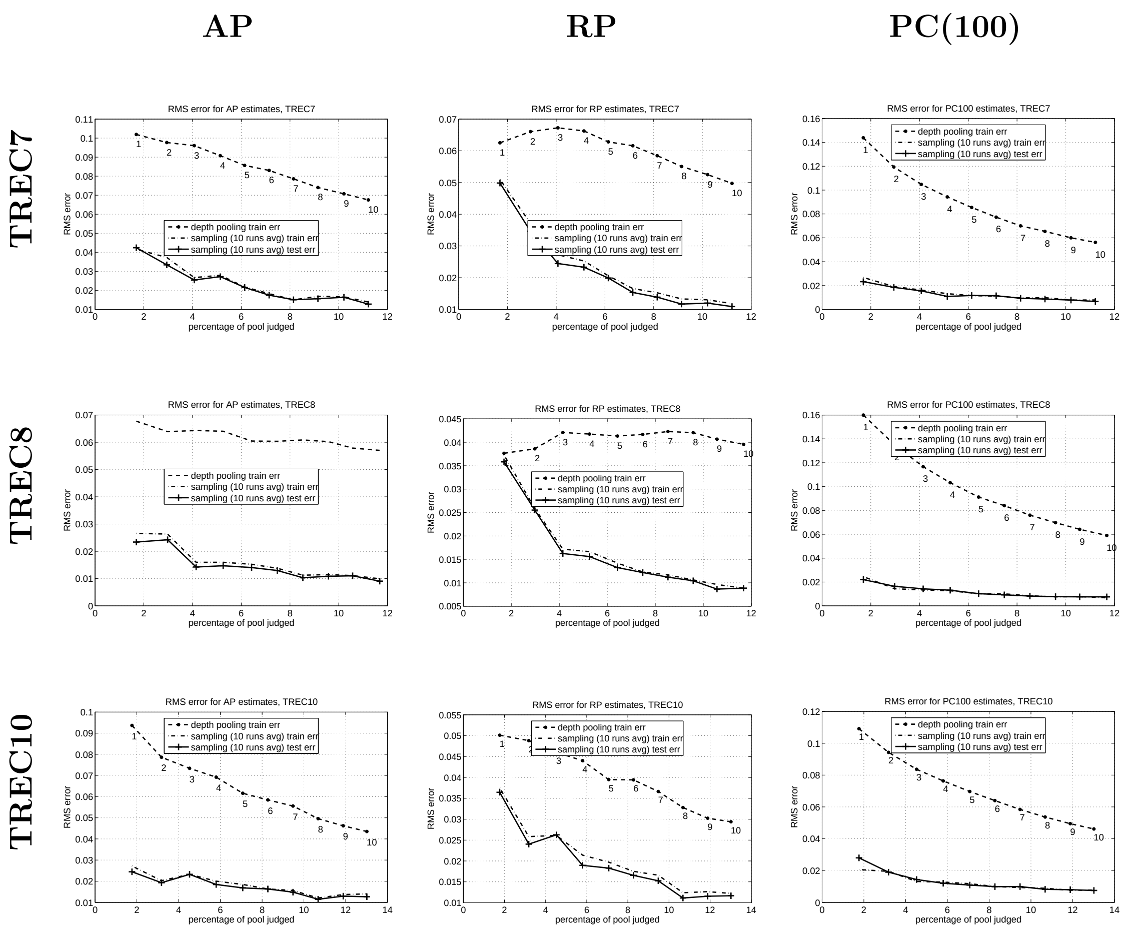


**Figure 10: RMS error train/test crossvalidation comparisons for MAP, RP, PC(100),in TRECs 7, 8 and 10. Equivalent depths are indicated on the plot.**