A Practical Sampling Strategy for Efficient Retrieval Evaluation

Javed A. Aslam   Virgil Pavlu
College of Computer and Information Science
Northeastern University
{jaa,vip}@ccs.neu.edu

ABSTRACT

We consider the problem of large-scale retrieval evaluation, with a focus on the considerable effort required to judge tens of thousands of documents using traditional test collection construction methodologies. Recently, two methods based on *random sampling* were proposed to help alleviate this burden: While the first method proposed by Aslam et al. is very accurate and efficient, it is also very complex, and while the second method proposed by Yilmaz et al. is relatively simple, its accuracy and efficiency are significantly lower than the former.

In this work, we propose a new method for large-scale retrieval evaluation based on random sampling which combines the strengths of each of the above methods: it maintains the simplicity of the Yilmaz et al. method while achieving the performance of the Aslam et al. method. Furthermore, we demonstrate that this new sampling method can be adapted to incorporate both randomly sampled and fixed relevance judgments, as were available in the most recent TREC Terabyte track, for example.

General Terms

Theory, Measurement, Experiment, Information Retrieval

Keywords

Performance Evaluation, Sampling, Average Precision

1. INTRODUCTION

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) [19]. TREC-style evaluation is largely based on the Cranfield paradigm [10], wherein collections of retrieval systems are evaluated by (1) constructing a collection of *documents*, (2) constructing a collection of *information needs* (variously referred to as *topics* or *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 average precision, R-precision, and precisions at various rank cutoffs.

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 to 25 million documents to each of 50 or more 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 judged. 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 k documents returned by each system in response to a given query. This depth-k pooling is appealing for at least two reasons: (1) It is "fair" in the sense that every system has (at least) its top-k retrieved documents judged, and (2) for sufficiently large k and for sufficiently many input systems, the depth-k pools can be "effectively complete," in the sense that it is unlikely that a relevant document would be retrieved *deeper* than rank k by *every* system, and thus the assumption that documents outside the depth-k pool are non-relevant is reasonable. Depth-100 pools have been shown to be effective in evaluating the relative performance of retrieval systems for many TREC collections [13, 22].

Unfortunately, TREC-style retrieval evaluation can be very expensive, often requiring that tens of thousands of documents be judged in order to obtain accurate, robust, and reusable assessments. In TREC 8, for example, 86,830 relevance assessments were collected using depth-100 pooling in order to evaluate 129 system runs submitted in response to 50 queries.

A number of methods have been proposed to potentially alleviate this assessment burden, including shallower depth pools [22], greedily chosen dynamic pools [11, 2, 7], and pools with randomly assigned relevance judgments [15]. However, these methods all tend to produce biased or incomparable estimates of commonly used measures of retrieval performance (such as average precision), especially when relatively few relevance assessments are made.

Recently, Aslam et al. [4] and Yilmaz et al. [21] each proposed new methods for efficient system evaluation based on pools chosen via *random sampling*. In the former method, a carefully chosen, non-uniform distribution over the documents in the depth-100 pool is formed, and documents are sampled with replacement according to this distribution. In order to assess a search engine run, the entire sampling distribution over the depth-100 pool must be available, together with the relevance judgments and sampling counts associated with the sampled documents. While this evaluation method is very accurate and efficient, achieving assessment results essentially equivalent to TREC depth-100 pooling results using sample sizes as low as 4% of the size of the traditional depth-100 pool, it is very complex both in its conception and its implementation. Conversely, the method proposed by Yilmaz et al. is quite simple: Documents are chosen uniformly at random from the depth-100 pool, and only those judged documents (and knowledge of the depth-100 pool) are required to assess any given search engine run. The method is quite simple in its conception and implementation. Unfortunately, while more efficient than traditional depth-pooling, the method is far less accurate and efficient than the (far more complex) method proposed by Aslam et al.—in order to achieve similarly accurate results, samples sizes roughly five times as large are required.

In this work, we propose a new method for large-scale, TREC-style retrieval evaluation based on random sampling which combines the strengths of each of the above two methods: it matches the simplicity of the Yilmaz et al. method while equaling or exceeding the performance of the Aslam et al. method. Furthermore, unlike the previous two methods, we demonstrate that this new sampling method can be adapted to incorporate additional judgments obtained via deterministic methods (such as traditional depth pooling), and thus our proposed method effectively generalizes and combines both random and fixed pooling techniques.

In the sections that follow, we describe our sampling and evaluation methodology, then we discuss the results obtained from extensive experiments conducted with TREC data. We conclude with a summary and discussion of future work.

2. METHODOLOGY

In this section, we describe our sampling methodology in detail. We begin with a simple example in order to provide intuition for the non-uniform sampling strategy ultimately employed, and we then proceed to describe the specific application of this intuition to the general problem of retrieval evaluation.

2.1 Sampling Theory and Intuition

As a simple example, suppose that we are given a ranked list of documents $(d_1, d_2, \ldots)$, and we are interested in determining the precision-at-cutoff 1000, i.e., the fraction of the top 1000 documents that are relevant. Let $PC(1000)$ denote this value. One obvious solution is to examine each of the top 1000 documents and return the number of relevant documents seen divided by 1000. Such a solution requires 1000 relevance judgments and returns the *exact* value of $PC(1000)$ with *perfect certainty.* This is analogous to forecasting an election by polling each and every registered voter and asking how they intend to vote: In principle, one would determine, with certainty, the exact fraction of voters who would vote for a given candidate on that day. In practice, the cost associated with such "complete surveys" is prohibitively expensive. In election forecasting, market analysis, quality control, and a host of other problem domains, *random sampling* techniques are used instead [17].

In random sampling, one trades-off *exactitude* and *certainty* for *efficiency.* Returning to our $PC(1000)$ example, we could instead *estimate* $PC(1000)$ with some *confidence* by sampling in the obvious manner: Draw m documents uniformly at random from among the top 1000, judge those documents, and return the number of relevant documents seen divided by m — this is analogous to a random poll of registered voters in election forecasting. In statistical parlance, we have a *sample space* of documents indexed by $k \in \{1, \ldots, 1000\}$, we have a *sampling distribution* over those documents $p_k = 1/1000$ for all $1 \leq k \leq 1000$, and we have a *random variable* X corresponding to the relevance of documents,

$$x_k = rel(k) = \begin{cases} 0 & \text{if } d_k \text{ is non-relevant} \\ 1 & \text{if } d_k \text{ is relevant.} \end{cases}$$

One can easily verify that the *expected value* of a single random draw is $PC(1000)$

$$E[X] = \sum_{k=1}^{1000} p_k \cdot x_k = \frac{1}{1000} \sum_{k=1}^{1000} rel(k) = PC(1000),$$

and the Law of Large Numbers and the Central Limit Theorem dictate that the *average* of a set S of m such random draws

$$\widehat{PC}(1000) = \frac{1}{m} \sum_{k \in S} X_k = \frac{1}{m} \sum_{k \in S} rel(k)$$

will converge to its expectation, $PC(1000)$, quickly [14] — this is the essence of random sampling.

Random sampling gives rise to a number of natural questions: (1) How should the random sample be drawn? In *sampling with replacement*, each item is drawn independently and at random according to the distribution given (uniform in our example), and repetitions may occur; in *sampling without replacement*, a random subset of the items is drawn, and repetitions will not occur. While the former is much easier to analyze mathematically, the latter is often used in practice since one would not call the same registered voter twice (or ask an assessor to judge the same document twice) in a given survey. (2) How should the sampling distribution be formed? While $PC(1000)$ seems to dictate a uniform sampling distribution, we shall see that non-uniform sampling gives rise to much more efficient and accurate estimates. (3) How can one quantify the accuracy and confidence in a statistical estimate? As more samples are drawn, one expects the accuracy of the estimate to increase, but by how much and with what confidence? In the paragraphs that follow, we address each of these questions, in reverse order.

While statistical estimates are generally designed to be correct in expectation, they may be high or low in practice (especially for small sample sizes) due to the nature of random sampling. The variability of an estimate is measured by its *variance*, and by the Central Limit Theorem, one can ascribe 95% confidence intervals to a sampling estimate given its variance. Returning to our $PC(1000)$ example, suppose that (unknown to us) the actual $PC(1000)$ was 0.25; then one can show that the variance in our random variable X is 0.1875 and that the variance in our sampling estimate is $0.1875/m$, where m is the sample size. Note that the variance decreases as the sample size increases, as expected. Given this variance, one can derive 95% confidence intervals [14], i.e., an error range within which we are 95% confident that our estimate will lie.[1] For example, given a sample of size 50, our 95% confidence interval is $+/-0.12$, while for a sample of size 500, our 95% confidence interval is $+/-0.038$. This latter result states that with a sample of size 500, our estimate is likely to lie in the range

[1] For estimates obtained by averaging a random sample, the 95% confidence interval is roughly $+/-1.965$ *standard deviations*, where the standard deviation is the square root of the variance, i.e., $\sqrt{0.1875/m}$ in our example.

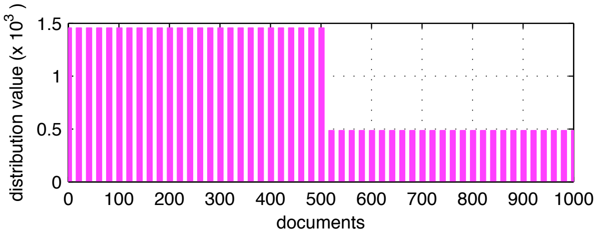

Figure 1: Non-uniform sampling distribution.

$[0.212, 0.288]$. In order to increase the accuracy of our estimates, we must decrease the size of the confidence interval. In order to decrease the size of the confidence interval, we must decrease the variance in our estimate, $0.1875/m$. This can be accomplished by either (1) decreasing the variance of the underlying random variable X (the 0.1875 factor) or (2) increasing the sample size m. Since increasing m increases our judgment effort, we shall focus on decreasing the variance of our random variable instead.

While our $PC(1000)$ example seems to inherently dictate a uniform sampling distribution, one can reduce the variance of the underlying random variable X, and hence the sampling estimate, by employing *non-uniform* sampling. A maxim of sampling theory is that accurate estimates are obtained when one samples with *probability proportional to size* (PPS) [17]. Consider our election forecasting analogy: Suppose that our hypothetical candidate is know to have strong support in rural areas, weaker support in the suburbs, and almost no support in major cities. Then to obtain an accurate estimate of the vote total (or fraction of total votes) this candidate is likely to obtain, it makes sense to spend your (sampling) effort "where the votes are." In other words, one should spend the greatest effort in rural areas to get very accurate counts there, somewhat less effort in the suburbs, and little effort in major cites where very few people are likely to vote for the candidate in question. However, one must now compensate for the fact that the sampling distribution is non-uniform — if one were to simply return the fraction of polled voters who intend to vote for our hypothetical candidate when the sample is highly skewed toward the candidates areas of strength, then one would erroneously conclude that the candidate would win in a landslide. To compensate for non-uniform sampling, one must *under-count* where one *over-samples* and *over-count* where one *under-samples.*

Returning to our $PC(1000)$ example, employing a PPS strategy would dictate sampling "where the relevant documents are." Analogous to the election forecasting problem, we do have a prior belief about where the relevant documents are likely to reside — in the context of ranked retrieval, relevant documents are generally more likely to appear toward the top of the list. We can make use of this fact to reduce our sampling estimate's variance, so long as our assumption holds. Consider the non-uniform sampling distribution shown in Figure 1 where

$$p_k = \begin{cases} 1.5/1000 & 1 \leq k \leq 500 \\ 0.5/1000 & 501 \leq k \leq 1000. \end{cases}$$

where we have *increased* our probability of sampling the top half (where more relevant documents are likely to reside) and *decreased* our probability of sampling the bottom half (where fewer relevant documents are likely to reside). In order to obtain the correct estimate, we must now "under-count" where we "over-sample" and "over-count" where we "under-sample." This is accomplished by modifying our random variable X as follows:

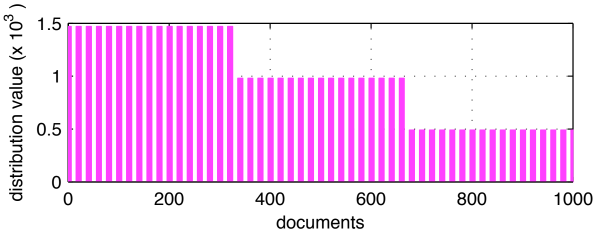

Figure 2: Non-uniform distrib. with three strata.

$$x_k = \begin{cases} rel(k)/1.5 & 1 \leq k \leq 500 \\ rel(k)/0.5 & 501 \leq k \leq 1000. \end{cases}$$

Note that we over/under-count by precisely the factor that we under/over-sample; this ensures that the expectation is correct:

$$\begin{aligned} E[X] &= \sum_{k=1}^{1000} p_k \cdot x_k = \sum_{k=1}^{500} \frac{1.5}{1000} \cdot \frac{rel(k)}{1.5} + \sum_{k=1}^{500} \frac{0.5}{1000} \cdot \frac{rel(k)}{0.5} \\ &= \frac{1}{1000} \sum_{k=1}^{1000} rel(k) = PC(1000). \end{aligned}$$

For a given sample S of size m, our estimator is then a weighted average

$$\begin{aligned} \widehat{PC}(1000) &= \frac{1}{m} \sum_{k \in S} X_k \\ &= \frac{1}{m} \left(\sum_{k \in S \,:\, k \leq 500} \frac{rel(k)}{1.5} + \sum_{k \in S \,:\, k > 500} \frac{rel(k)}{0.5} \right) \end{aligned}$$

where we over/under-count appropriately.

Note that our expectation and estimator are correct, *independent of whether our assumption about the location of the relevant documents actually holds!* However, if our assumption holds, then the variance of our random variable (and sampling estimate) will be reduced (and vice versa). Suppose that all of the relevant documents were located where we over-sample. Our expectation would be correct, and one can show that the variance of our random variable is reduced from 0.1875 to 0.1042 — we have sampled where the relevant documents are and obtained a more accurate count as a result. This reduction in variance yields a reduction in the 95% confidence interval for a sample of size 500 from $+/-0.038$ to $+/-0.028$, a 26% improvement. Conversely, if the relevant documents were located in the bottom half, the confidence interval would increase.

One could extend this idea to three (or more) strata, as in Figure 2. For each document k, let α_k be the factor by which it is over/under-sampled with respect to the uniform distribution; for example, in Figure 1, α_k is 1.5 or 0.5 for the appropriate ranges of k, while in Figure 2, α_k is 1.5, 1, or 0.5 for appropriate ranges of k. For a sample S of size m drawn according to the distribution in question, the sampling estimator would be

$$\widehat{PC}(1000) = \frac{1}{m} \sum_{k \in S} \frac{rel(k)}{\alpha_k}.$$

In summary, one can sample with respect to any distribution, and so long as one over/under-counts appropriately, the estimator will be correct. Furthermore, if the sampling distribution places higher weight on the items of interest (e.g., relevant documents), then the variance of the estimator will be reduced, yielding higher accuracy.

Finally, we note that sampling is often performed *without replacement* [17]. In this setting, the estimator changes somewhat, though the principles remain the same: sample where you think the relevant documents are in order to reduce variance and increase accuracy. The α_k factors are replaced by *inclusion probabilities* π_k, and the estimator must be normalized by the size of the sample space:

$$\widehat{PC}(1000) = \frac{1}{1000}\sum_{k\in S}\frac{rel(k)}{\pi_k}.$$

The inclusion probability π_k is simply the probability that the document k would be included in any sample of size m. In without-replacement sampling, $\pi_k = p_k$ when $m = 1$ and π_k approaches 1 as the sample size grows. Note that documents with large inclusion probabilities (i.e., those likely to be sampled) are under-counted as compared to those with small inclusion probabilities, which are appropriately over-counted, as desired. Furthermore, if the size of the sample space itself is unknown (suppose that we did not know the number of registered voters in our election forecasting analogy), then one can estimate this quantity as well. This yields the Horwitz-Thompson *generalized ratio estimator* [17]

$$\widehat{X} = \frac{\sum_{k\in S} v_k/\pi_k}{\sum_{k\in S} 1/\pi_k}$$

where v_k is the *value* associated with item k (e.g., the relevance of a document, a vote for a candidate, the size of a potential donation, etc.). PPS-without-replacement sampling is known to be much more robust and efficient than with-replacement sampling [17, 5]. However, for most sampling strategies, inclusion probabilities are notorious difficult to compute when sample size is reasonably large[5].

The generalized ratio estimator is most useful for estimating *average precision* (AP), which is the average of the precisions at relevant documents. In sampling to estimate the average precision(s) of one or more given document lists, one typically does not know the total number of relevant documents R; thus, the generalized ration estimator is applicable since it effectively estimates R as well. The "values" we wish to average are the precisions at relevant documents, and the ratio estimator for AP is thus

$$\widehat{AP} = \frac{\displaystyle\sum_{k\in S\,:\,rel(k)=1} PC(k)/\pi_k}{\displaystyle\sum_{k\in S\,:\,rel(k)=1} 1/\pi_k}$$

We proceed by breaking the discussion in three parts: the *sample*, which is the set of documents sampled together with relevance judgments and other information we need; the *evaluation* module, which given the sample produces estimates for AP and other quantities of interest; and the *sampling* module which of course produces the sample. For analogy with TREC, the sampling strategy is the equivalent of depth-pooling, the evaluation the equivalent of trec-eval program and the sample the equivalent of traditional qrel files[18], the only addition being that every sampled document is also accompanied by an inclusion probability. This methodology works very well for small sizes (Kendall's τ=.85 for less than %2 of the pool judged) and it also can be smoothly adapted towards traditional setup (depth-pooling, trec-eval, qrel) by adding depth-pooling style judged documents; at the very extreme when all documents are judged, our estimated values are identical with trec-eval outputs.

Modularity. The evaluation and sampling modules are completely independent: the sampling module produces the sample in a specific format but does not imposes or assumes a particular evaluation being used; the evaluation module uses the sample only (no knowledge/assumption of the sampling strategy used to obtain the sample, a strong improvement over method presented in [4]). In fact, the sampling technique proposed is known to work with many other estimators (evaluations) while the estimator used is known to work with other sampling strategies [5]. This flexibility is particularly important if one has reason to believe that a different sampling strategy might work better for a given instance.

2.2 The sample

The sample is the set of documents selected for judging together with all information required for evaluation: in our case that means (1) the documents ids, (2) the relevance assessments and (3) the inclusion probability for each document. Sampling is done without replacement so there are no counts (sampling with-replacement requires also the count for each document; that is, how many times each document has been sampled).

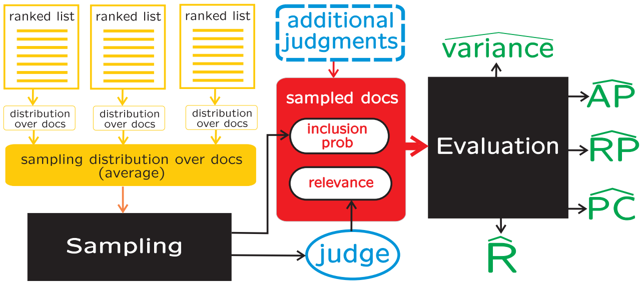

Figure 3: Sampling and evaluation design

Additional judged documents, obtained deterministically (usually with a greedy strategy like depth-pooling), can be added to the existing sample with associated inclusion probability of 1. This is a useful design feature as often in practice separate judgments are available; it makes the method especially attractive for tasks like Terabyte tracks[9, 8] or the new proposed TREC million query track. In TREC setup it can be used in two ways: first when massive central judging is done by assessors, it might be desirable to judge deterministically a given depth-pool (say top 10 documents of every participant list) and then invoke the sampling strategy to judge additional documents. Second, when a participant receives the relevance assessments from TREC ("qrel" file [18]), if more judgments are needed, one can judge either hand-picked documents and/or sampled documents and combine them with the provided qrel.

The collisions (where a document is sampled and separately deterministically judged) are handled by setting inclusion probability to 1; this is because the actual probability to have the document in the sample is 1, once it has been judged. Science evaluation module (next section) does not make any assumption of the sampling strategy used (it uses as input only the sample), at the sample level we can mix different pools of judged documents as long as we can properly update the inclusion probabilities.

2.3 Evaluation

Given a sample S of judged documents along with inclusion probabilities, we discuss here how to estimate quantities of interest (AP, R-precision, Precision at cutoff). In next section we show how to obtain such a sample S. For AP estimate, which we view as mean of a population of precision values, we adapt the generalized ratio estimator for unequal probability designs (very popular on polls, election strategies, market research etc.), as described in [17]:

$$\widehat{AP} = \frac{\sum_{d \in S: rel(d)=1} \widehat{PC}(rank(d))/\pi_d}{\sum_{d \in S: rel(d)=1} 1/\pi_d} \tag{1}$$

where $\widehat{PC}(r)$ estimates precision at rank r:

$$\widehat{PC}(r) = \frac{1}{r} \sum_{d \in S, rank(d) \le r} \frac{rel(d)}{\pi_d} \tag{2}$$

See the appendix for a discussion about inclusion probabilities, variance and confidence intervals.
Other estimates. Combining the estimates for R (number of relevant documents for the query) and for precision at cutoff (or rank), $PC(r)$, we obtain an estimate for R-precision (precision at rank R):

$$\widehat{R} = \sum_{d \in S: rel(d)=1} \frac{1}{\pi_d} \tag{3}$$

$$\widehat{RP} = \widehat{PC}(\widehat{R}) = \frac{1}{\widehat{R}} \sum_{d \in S, rank(d) \le \widehat{R}} \frac{rel(d)}{\pi_d} \tag{4}$$

2.4 Sampling strategy

There are many ways one can imagine sampling from a given distribution [5]. Essentially, we have to make two independent decisions. First, decide on a sampling distribution over documents; it should be dictated by the ranks of documents in the ranked lists and therefore naturally biased towards relevant documents. Second we have to design an actual sampling strategy that given the prior distribution produces the sample.

Sampling distribution. It has been shown that average precision induces a good relevance prior over the ranked documents of a list. The AP-prior has been used with sampling techniques[4]; in metasearch (data fusion) [3]; in automatic assessment of query difficulty [1]; and in on-line application to pooling[2]. It has also been shown that this prior can be averaged over multiple lists to obtain a global prior over documents[4]. An accurate description together with motivation and intuition can be found in [4].

For a given ranked list of documents, let Z be the size of the list. Then the prior distribution weight associated with any rank r, $1 \le r \le Z$, is given by

$$W(r) = \frac{1}{2Z}\left(1 + \frac{1}{r} + \frac{1}{r+1} + \cdots + \frac{1}{Z}\right) \approx \frac{1}{2Z} \log \frac{Z}{r}. \tag{5}$$

We used for experimentation the above described prior, averaged per document over all run lists; Note that the our sampling strategy (next section) works with any prior over documents.

Stratified sampling strategy. The most important considerations are: handle *non-uniform* sampling distribution; *without replacement* so we can easily add other judged documents; *probabilities proportional with size (pps)* minimizes variance by obtaining inclusion probabilities π_d roughly proportional with precision values $PC_{rank(d)}$; and *computability of inclusion probabilities* for documents (π_d) and for pairs of documents (π_{df}). We adopt a method developed by Stevens [5, 16], sometimes referred to as *stratified sampling*, that has all of the features enumerated above and it is very straight forward for our application.

Let W be the sampling distribution over N documents m be the sample of size desired (Figure 4,top). Stevens stratified sampling works as follows:

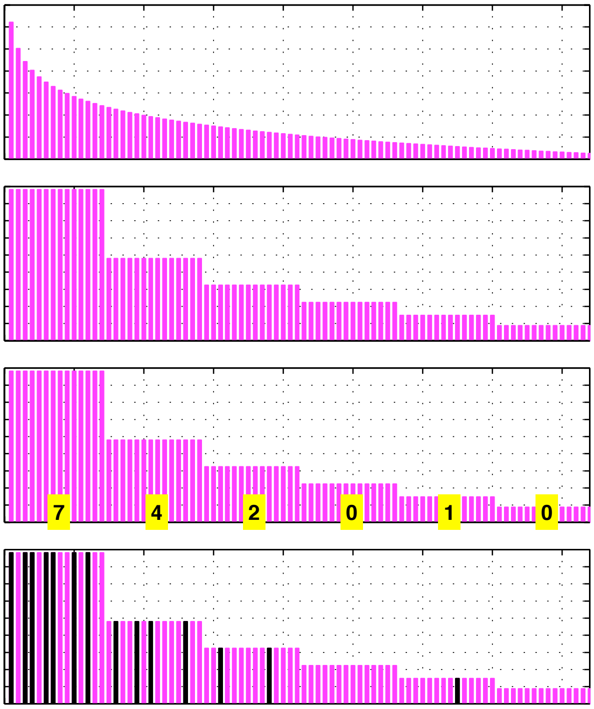

Figure 4: Average Precision induced prior W, averaged over many system lists (top). Bucketed prior (second row): Each bucket contains $m = 14$ items (in this example) and it is associated with sum of distribution weights of its items. Third: Buckets are sampled with replacement, obtaining counts 7,4,2,0,1,0 (summing to m=14). Bottom: Inside each bucket documents are sampled uniformly, without replacement: from first bucket 7 items, from second bucket 4 items, and so on.

1. Order documents by sampling weight and partition them in buckets of size m each(Figure 4, second row). First bucket will contain biggest (by W) m documents, second bucket the next m documents and so on. Last bucket might have fewer than m but that fact is negligible.

2. Pick *with replacement* the buckets m times, where each bucket has probability to be chosen (each time) the sum of W weights associated with documents in the bucket(Figure 4,third).
3. For each bucket, if it got picked k times, sample *uniformly, without replacement* k documents from the bucket (Figure 4,bottom, black bars indicate documents sampled).

Obviously, this strategy is fast and simple. Although buckets are sampled with replacement, the overall sample of documents is without replacement. Also inclusion probabilities for each document π_d are easy to compute (appendix).

3. EXPERIMENTAL RESULTS

We tested the proposed method as a mechanism for estimating the performance of retrieval systems using data from TRECs 7, 8 and 10. Using mean average precision (MAP), mean R-precision (MRP), and mean precision at cutoff 30 (MPC(30)) 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* ρ (how well the actual and estimated values fit to a straight line), and *Kendall's* τ (how well the estimated measures rank the systems compared to the actual rankings). 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 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 τ or ρ values. In fact one can show that for an estimator $\widehat{\theta}$, $RMS(\widehat{\theta}) = \sqrt{var[\widehat{\theta}] + bias^2(\widehat{\theta})}$, which directly implies that small RMS means both small bias and small variance and also that bias and variance are equally important in error measurement.

We compare the sampling estimates (various sample sizes) with actual "true" values (obtained by TREC with fully judged depth-100 pool). The size of samples are based on depth pooling equivalence; depths 1 (top document for every run) and 10(top 10 documents for every run) are displayed. TREC-style depth pooling for depths 1 and 10 correspond to 40 and 260 relevance judgments on average per query, respectively (including the pool non-participating runs). We also compare the estimated values of the measures obtained using the sampling method with the depth pooling estimates.

Since the performance of the sampling method varies depending on the actual sample, we repeated each experiment 100 times. For lineplots (Figures 10,12) we present the averaged the measurements RMS, ρ, τ over 100 trials; for scatterplots we picked a representative experiment that exhibits typical performance in terms of RMS, ρ, τ.

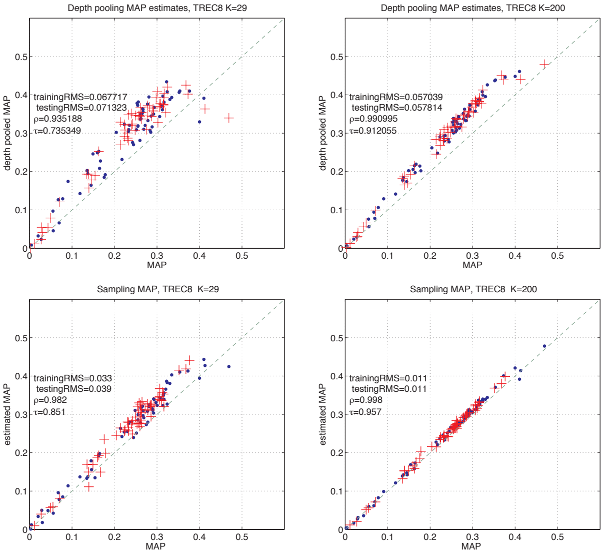

Figure 5: Sampling vs. depth pooling mean average precision estimates at depths 1 and 10 in TREC8. Each dot ($\cdot$) corresponds to a distribution-contributor run and each plus ($+$) to a distribution-non-contributor run (there are 129 runs in TREC8.)

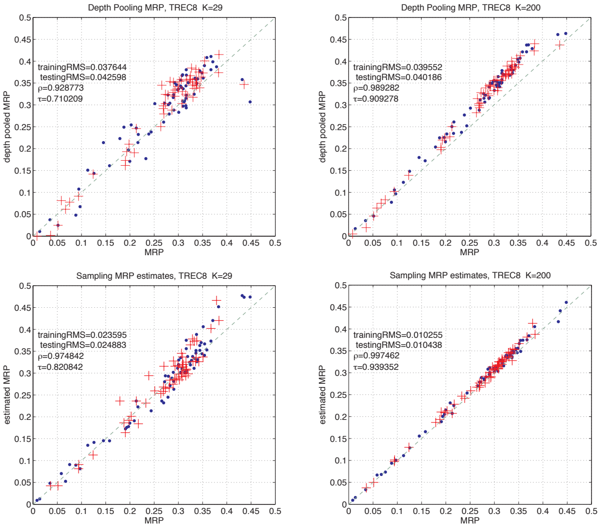

Figure 6: Sampling vs. depth pooling mean R-precision estimates at depths 1 and 10 in TREC8.

We report the results of the experiments for MAP, MRP, and MPC(30) on TREC8 in Figure 5, Figure 6, and Figure 7, 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 bias compared to depth pooling estimates; furthermore, the bottom-right plots of the figures show that 200 relevance judgments on average per query are enough to get "almost perfect" evaluations ($\tau \approx .95$); actual TREC8 evaluations use 1,737 relevance judgments on average per query.

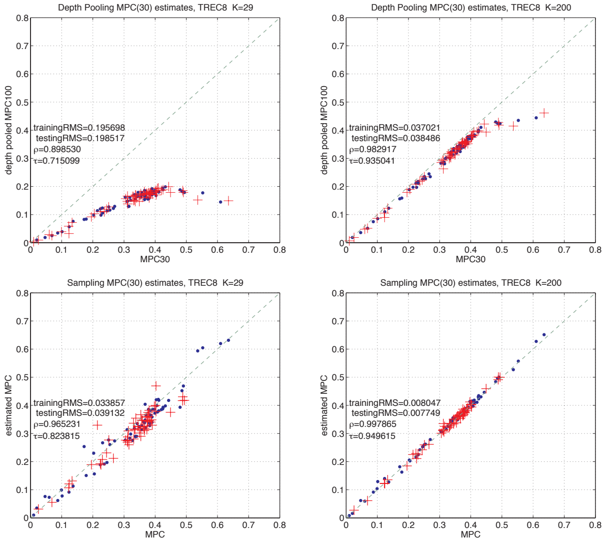

Figure 7: Sampling vs. depth pooling mean prec at cutoff 100 estimates at depths 1 and 10 in TREC8.

Figure 10 illustrates how MAP estimates using TREC-style depth pooling compare in terms of ρ and Kendall's τ with those obtained using sampling as the depth of the pool changes. For depths 1 to 10, we calculated the number of documents required to be judged using TREC-style depth pooling to determine the equivalent sample size. For each sample size (equivalent depth pool size) we repeated the experiment 100 times and then calculated the average ρ (left column), RMS (middle) and τ (right column). Along with the average displayed in the Figure 10, for ρ and τ we plot the standard deviation bar estimated unbiased from 100 values (sampling line on the plots, bar shows ±1 std). Generally speaking the results obtained are comparable or better with a previous (much more complicated) sampling method [4]. As the figure displays, the sampling method significantly outperforms the TREC-style depth pooling evaluations. For comparison purposes, we also include the average Kendall's τ value of bpref [6] and infAP [21] obtained using *random samples* of the given size to the plots in the second column. The Kendall τ values for bpref and infAP are the average values computed over 10 different random samples (infAP numbers were not calculated in this setup, instead they were obtained from infAP authors [21]).

Per query and per run results. There are certain situations when one needs the results of a single query, hence not taking advantage of the variance reduction achieved by averaging over 50 queries. It is certainly not expected to see the same kind of performance on query by query basis; however our results show definite usable query estimates (Figure 8). The method described in this paper is self-contained for a query, i.e., estimates for a query are not dependent on any data from other queries. On a different setup, one may want to analyze only one run over all queries (Figure 9).

3.1 Generalization on new runs

For each experiment we split the runs into "training" and "testing" (sometimes called distribution-contributors and distribution noncontributors respectively). The training runs are the ones pooled by TREC when the relevance assessment took place; the rest are testing runs. This distinction seems fair for comparison with TREC published run performance

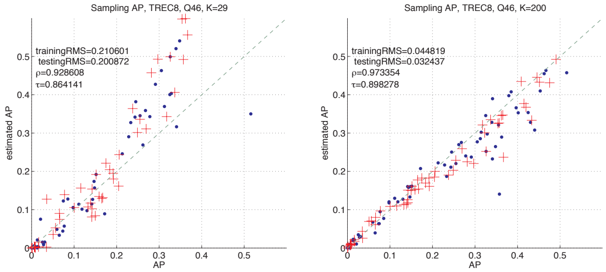

Figure 8: Sampling estimates for a query with mixed system performance. Dots (·) represent training runs; pluses (+) represent testing runs.

.

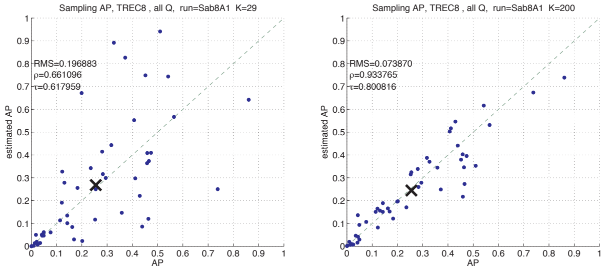

Figure 9: Sampling estimates for a fixed typical run (Sab8A1) with MAP = 0.25, all queries. Each dot (·) is an AP for a query estimate (total 50); MAP estimate is plotted as "×".

numbers. On all scatterplots, training runs are denoted by blue dots while testing ones are denoted by red crosses.

It is important that the performance of sampling over the testing runs is virtually as good as the performance over the training runs (good generalization). The trend of RMS error, as sample size increases from depth 1 to depth 10 equivalent for training and testing systems is shown in Figure 12. On x-axis the units are the depth-pool equivalent number of judgments converted into percentages of depth-100 pool.

3.2 Significance tests

Is search engine A better than search engine B? Each engine is evaluated at TREC over 50 queries, hence 50 Average Precision numbers; we can compute an overall measurement, "Mean Average Precision" (MAP) by averaging those 50 numbers for each engine. But is the difference in MAP between A and B significant? The Wilcoxon Significance Test [20] takes as input the array of 50 differences AP values and produces a *left-p-value* that can be interpreted as the probability that A is better than B; if p-value> 0.95 we conclude that the MAP-based ordering of engines is significant. Ideally, for each run-pair (A,B) with significant TREC MAP difference (and only for those pairs), sampling AP estimates would also lead to significant MAP difference.

		SAMPLING(K=29)		SAMPLING(K=200)	
		insignif	signif	insignif	signif
TREC	insignif	9970	143	9902	211
	signif	1105	5423	318	6210

Table 1: Trec8: Number of run-pairs (A,B) for each significance category. Sampling estimates are obtained with 29 judgments per query (left), and with 200 judgments per query (right).

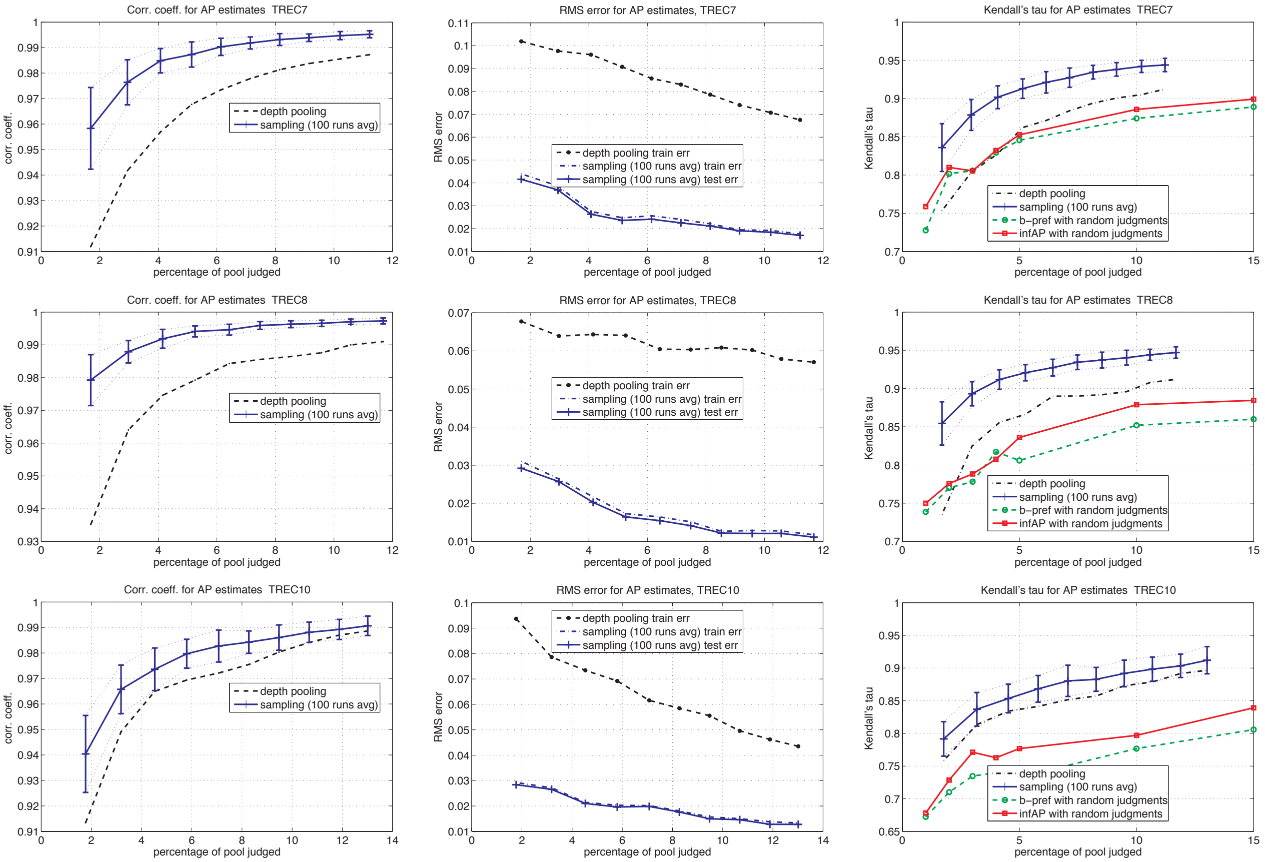

Figure 10: Linear correlation coefficient, Kendall's τ and RMS error comparisons for mean average precision, in TRECs 7, 8 and 10.

Each pair of runs (A,B) is thus classified "significant by TREC" (p-value >.95) or "insignificant by TREC" (p-value <.95); if we use as input for Wilcoxon test our AP estimates instead of TREC APvalues, we classify each pair as "significant by SAMPLING" and "insignificant by SAMPLING" (Table 1); in Trec8 there are 129 participant runs hence 129^2 ordered pairs). We note that sampling estimates are concluding "significant" on most of the run pairs that are actually significantly different; as the sample grows in size, the numbers of run pairs incorrectly considered "insignificant" decreases (Table 1, right).

3.3 Using additional judged documents

In many cases, additional judgments are available from various sources. Most important, those judgments are independent of the sampling picked judgments and some may be common (collisions). We next demonstrate that the evaluation stage of our method can use the additional judgments. In a very simple fashion, the additional judgments are added to the sample with inclusion probability 1 (each).

Figure 11 presents evaluation results when additional judgments are used. The 3 plots on the left (top to bottom: sampling, sampling + depth pooling, depth pooling) show results for a particular query on TREC8. A comparison between top plot and middle plot shows a significant improvement when additional documents are added; this corresponds to a scenario where sampling is done first and additional judgments provided later. The combined result shows significant improvement over the depth pool strategy (bottom plot), as expected and demonstrated above; this corresponds to a scenario where depth pool judgments are available to start with and sampling is used to obtain additional judgments. The right 3 plots are showing corresponding results for all queries on TREC8.

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.

The method presented has the advantages of other methods proposed [21, 4], but not their disadvantages (hard to put in practice, good but not best estimates). It also brings several novice features: independence between sampling and evaluation, ability to use existing judgments, and computability of variance of the estimators.

This work leaves open a number of question for further research. 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?

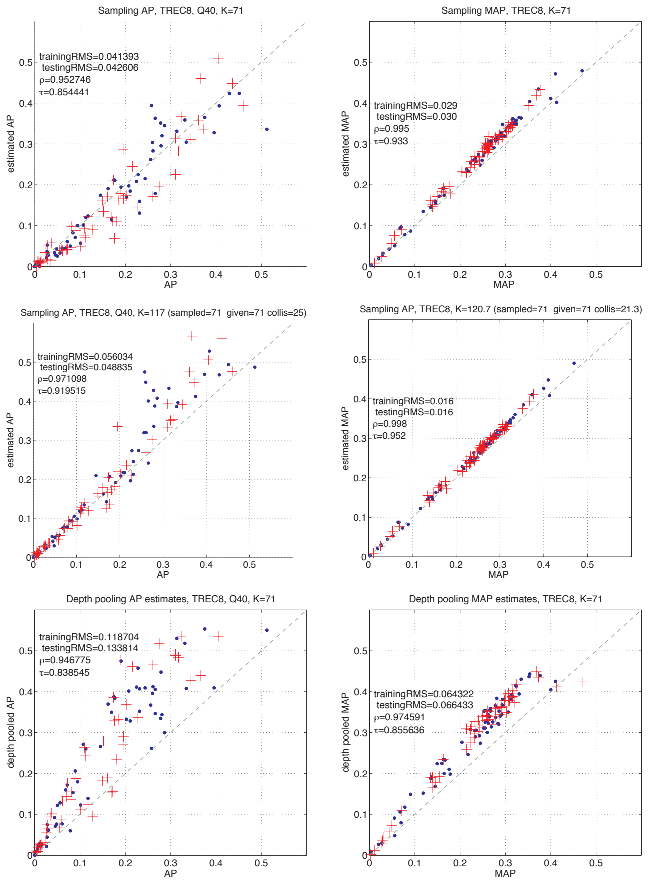

Figure 11: Sampling combined with depth pooling judgments (middle row) compared with Sampling (top) and depth pooling(bottom) evaluations on TREC8. Left plots are for a fixed query while right plots are averages for all queries.

Starting from the variance computation, one could in principle derive high probability confidence intervals for the estimates obtained, and such confidence intervals would be quite useful in practice (see apendix).

5. REFERENCES

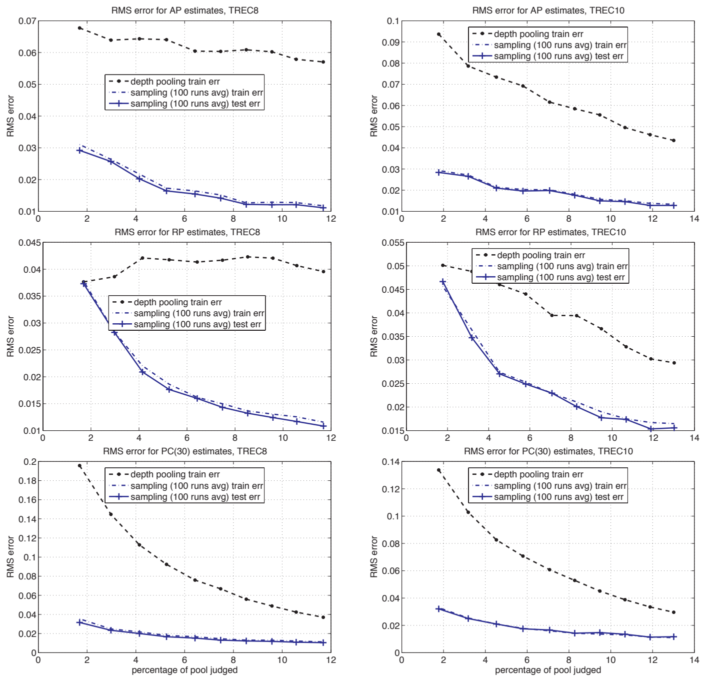

Figure 12: RMS error train/test comparisons for MAP, RP, PC(30), in TRECs 8 and 10. Equivalent depths are indicated on the plot.

[1] J. A. Aslam and V. Pavlu. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In G. Amati, C. Carpineto, and G. Romano, editors, *Advances in Information Retrieval: 28th European Conference on IR Research*, volume 4425 of *Lecture Notes in Computer Science*, pages 198–209. Springer-Verlag, 2007.

[2] J. A. Aslam, V. Pavlu, and R. Savell. A unified model for metasearch, pooling, and system evaluation. In O. Frieder, J. Hammer, S. Quershi, and L. Seligman, editors, *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 484–491. ACM Press, November 2003.

[3] J. A. Aslam, V. Pavlu, and E. Yilmaz. Measure-based metasearch. In G. Marchionini, A. Moffat, J. Tait, R. Baeza-Yates, and N. Ziviani, editors, *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 571–572. ACM Press, August 2005.

[4] J. A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In S. Dumais, E. N. Efthimiadis, D. Hawking, and K. Jarvelin, editors, *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 541–548. ACM Press, August 2006.

[5] K. R. W. Brewer and M. Hanif. *Sampling With Unequal Probabilities*. Springer, New York, 1983.

[6] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–32, 2004.

[7] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 268–275, 2006.

[8] C. Clarke, N. Craswell, and I. Soboroff. The TREC terabyte retrieval track. 2004.

[9] C. L. A. Clarke, F. Scholer, and I. Soboroff. The TREC 2005 terabyte track. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, 2005.

[10] C. Cleverdon. The cranfield tests on index language devices. In *Readings in Information Retrieval*, pages 47–59. Morgan Kaufmann, 1997.

[11] G. V. Cormack, C. R. Palmer, and C. L. A. Clarke. Efficient construction of large test collections. In Croft

et al. [12], pages 282–289.

[12] W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors. *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Aug. 1998.

[13] D. Harman. Overview of the third text REtreival conference (TREC-3). In D. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 1–19. U.S. Government Printing Office, Apr. 1995.

[14] J. A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, second edition, 1995.

[15] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 66–73, Sept. 2001.

[16] W. L. Stevens. Sampling without replacement with probability proportional to size. *Journal of the Royal Statistical Society. Series B (Methodological), Vol. 20, No. 2. (1958), pp. 393-397.*

[17] S. K. Thompson. *Sampling*. Wiley-Interscience, second edition, 2002.

[18] E. M. Voorhees and D. Harman. Overview of the Eighth Text REtrieval Conference (TREC-8). In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 1–24, 2000.

[19] E. M. Voorhees and D. K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.

[20] D. D. Wackerly, W. Mendenhall, and R. L. Scheaffer. *Mathematical Statistics with Applications*. Duxbury Advanced Series, 2002.

[21] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In P. S. Yu, V. Tsotras, E. Fox, and B. Liu, editors, *Proceedings of the Fifteenth ACM International Conference on Information and Knowledge Management*, pages 102–111. ACM Press, August 2006.

[22] J. Zobel. How reliable are the results of large-scale retrieval experiments? In Croft et al. [12], pages 307–314.

APPENDIX

Inclusion Probabilities. Say document d belongs to a bucket (of size m) with cumulative probability g and lets denote T the random variable indicating how many times the this bucket is picked (in the with-replacemnt bucket sampling, out of m times). Then

$$\pi_d = \sum_{k=1}^{m} Prob[T = k]\frac{k}{m} = \frac{1}{m}E[T] = \frac{1}{m}mg = g \qquad (6)$$

This shows that the sampling strategy is *probability proportional to size or "pps"*, meaning inclusion probabilities are approximative proportional with sampling probabilities. While g is the inclusion probability for all documents in the bucket (which originally don't have identical sampling probabilities, but close due to bucketing after ordering), for small m it is safe to say the method is pps. For large sample sizes, there would be small number of buckets, each with a uniform distribution over its documents. As an extreme example, when sample size is half of the number of total documents considered for sampling, we would have 2 buckets (first one having most of the cumulative weight) and sampling becomes close to uniform; but then due to a large sample, most relevant documents would be sampled and therefore estimates are accurate.

W compute inclusion probability for each pair of documents (d,f) (the probability that both documents are included in the sample) by distinguishing two cases. If d and f belong to the same bucket with total probability g then:

$$\begin{aligned}\pi_{df} &= \sum_{1\leq k\leq m} \frac{k(k-1)}{m(m-1)} Prob[T = k] \\ &= \sum_{1\leq k\leq m} \frac{k(k-1)}{m(m-1)} g^k(1-g)^{m-k}\binom{m}{k} \\ &= g^2 = \pi_d\pi_f\end{aligned}$$

If documents d and f belong to different buckets with total probability g and h respectively and T, U are random variables indicating how many times each bucket got picked, then

$$\begin{aligned}\pi_{df} &= \sum_{1\leq k+l\leq m} \frac{kl}{m^2} Prob[T = k; U = l] \\ &= \sum_{1\leq k+l\leq m} \frac{kh}{m^2} g^k h^l (1-g-h)^{m-k-l}\binom{m}{k, l, m-l-k} \\ &= \frac{m-1}{m} gh = \frac{m-1}{m}\pi_d\pi_f\end{aligned}$$

Variance. We show next how compute the variance of the AP estimator. If we denote $y_d = \widehat{PC}(rank(d)) - \widehat{AP}$, we obtain the following formula for estimated variance, adapted from[17]:

$$\widehat{var}(\widehat{AP}) = \frac{1}{\widehat{R}^2}\left(\sum_{d\in S,rel}\frac{1-\pi_d}{\pi_d^2}y_d^2 - \frac{1}{|S|-1}\sum_{d,f\in S,rel}\frac{y_d\cdot y_f}{\pi_d\cdot\pi_f}\right)$$

While $\widehat{PCR}_c$ and $\widehat{R}$ are unbiased estimators, the ratio estimator $\widehat{AP}$ it is not guaranteed to be unbiased. Some of our results have a small positive bias but it is negligible for any practical situation and definitely a large improvement compared to depth-pooling evaluation bias.

Confidence intervals. Under the assumption that the sample is large enough so that we can approximate the estimator distribution with a Gaussian, we compute a 95% confidence interval as the interval of possible values of the estimator, around the mean of the Gaussian and of length about two standard deviations each side. So the length of the confidence interval is about 4σ :

$$CI_{length} = 4\sigma = 4\sqrt{\widehat{var}(\widehat{AP})}$$

An important question (to be answered in future work) is, given a target confidence interval, a given prior distribution over documents and fixing the estimator and the sampling strategy to be the ones proposed here, *how many documents one needs to sample?* Thats is, solve for the size of the sample by setting 2σ to be a specific target; one possible complication are estimated quantities for precision numbers in the variance formula.