UCL DEPARTMENT OF
COMPUTER SCIENCE

Research Note
RN/11/16

Prioritizing Relevance Judgments to Improve the Construction of IR Test Collections

<10 Jun 2011>

Mehdi Hosseini
Ingemar J.Cox
Trevor Sweeting
Natasa Milic-Frayling
Vishwa Vinay

Abstract

We consider the problem of optimally allocating a fixed budget to construct a test collection with associated relevance judgements, such that it can (i) accurately evaluate the relative performance of the participating systems, and (ii) generalize to new, previously unseen systems. We propose a two stage approach. For a given set of queries, we adopt the traditional pooling method and use a portion of the budget to evaluate a set of documents retrieved by the participating systems. Next, we analyze the relevance judgments in order to prioritize the queries and associated documents for further refinement of the test collection. Our objective is to increase the effectiveness of the test collection for comparative evaluation and extendibility to new systems. The query prioritization is formulated as a convex optimization problem, thereby permitting efficient solution and providing a flexible framework to incorporate various constraints. We use the remaining budget to evaluate query-document pairs with the highest *priority scores*. The budgets for the initial and the refinement phase are expended during the construction of the test collection and consider only the documents that have been retrieved by the participating systems. We evaluate our resource optimization approach on two TREC test collections namely TREC 8 and TREC 2004 Robust Track. We demonstrate that our optimization techniques are cost efficient and yield a significant improvement in the reusability of the test collections.

Prioritizing Relevance Judgments to Improve the Construction of IR Test Collections

Mehdi Hosseini[1], Ingemar J.Cox[1], Trevor Sweeting[1], Natasa Milic-Frayling[2], Vishwa Vinay[2]

[1]University College London, [2]Microsoft Research Cambridge

{m.hosseini,i.cox}@cs.ucl.ac.uk, trevor@stats.ucl.ac.uk; {natasamf,vvinay}@microsoft.com

ABSTRACT

We consider the problem of optimally allocating a fixed budget to construct a test collection with associated relevance judgements, such that it can (i) accurately evaluate the relative performance of the participating systems, and (ii) generalize to new, previously unseen systems. We propose a two stage approach. For a given set of queries, we adopt the traditional pooling method and use a portion of the budget to evaluate a set of documents retrieved by the participating systems. Next, we analyze the relevance judgments in order to prioritize the queries and associated documents for further refinement of the test collection. Our objective is to increase the effectiveness of the test collection for comparative evaluation and extendibility to new systems. The query prioritization is formulated as a convex optimization problem, thereby permitting efficient solution and providing a flexible framework to incorporate various constraints. We use the remaining budget to evaluate query-document pairs with the highest *priority scores*. The budgets for the initial and the refinement phase are expended during the construction of the test collection and consider only the documents that have been retrieved by the participating systems. We evaluate our resource optimization approach on two TREC test collections namely TREC 8 and TREC 2004 Robust Track. We demonstrate that our optimization techniques are cost efficient and yield a significant improvement in the reusability of the test collections.

2. INTRODUCTION

Test collections are needed to measure both the absolute and relative performance of systems. A test collection consists of (i) a document collection, (ii) a set of test queries, and (iii) a set of corresponding relevance judgements. Ideally, every document in the collection would be judged relevant or non-relevant with respect to every query in the test set. In practice this is infeasible due to economic constraints. Instead, an IR test collection is typically constructed in conjunction with a set of participating IR systems. Each participating system retrieves a set of documents in response to each test query and these sets are pooled together. Relevance judgments are then obtained only for documents in the pool and specific metrics are used to compare systems performance. While the number of relevance judgments needed is greatly reduced, economic constraints may still prevent exhaustive judgments of all documents in the pool.

Conference'04, Month 1–2, 2004, City, State, Country.

In this paper, we consider how to prioritize query-document pairs for relevance judgments, when budget constraints preclude obtaining relevance judgments for all documents. We formulate the question as an optimization problem in which, for a given budget, we seek to identify a set of *n* query-document pairs that most accurately rank the participating systems and provide the best generalization to yet unseen systems. The latter refers to systems that have not contributed to the pool of evaluated documents. Section 3 provides a precise definition of what is meant here.

The main contributions of this paper are (i) explicitly incorporating a cost constraint within the optimization, which we believe has not been previously considered, (ii) formulation of the optimization problem as a convex optimization, for which computationally efficient algorithms exist for finding a globally optimum solution, (iii) the incorporation of a generalization constraint based on the estimated number of un-judged relevant documents for each query, and (iv) an extension of our base algorithm to provide a biased estimator when new systems are expected to be significantly better than participating systems.

In Section 2 we discuss related work. Particular attention is given to the work of Weber and Park [1], against which we compare our algorithm. Section 3 then provides a detailed description of our algorithm, while Section 4 describes specific implementation issues. Section 5 provides experimental results on both the TREC-8 and Robust TREC test collections. Finally, Section 6 provides a summary of our results and suggestions for future research directions.

3. RELATED WORK

Sparck-Jones and Van Rijsbergen proposed the pooling technique [2] as a means of creating an effective sample of judged documents to enable comparative performance evaluation of a set of retrieval systems. The National Institute of Standard and Technology (NIST) adopted this method in most of the TREC tracks. For example, in TREC AdHoc and Routing tasks, each participating system adds the top-100 ranked documents per query to the pool. All the documents in the joint pool are then labeled by human assessors. This enables NIST to compute a system's effectiveness metrics, such as interpolated average precision and recall by considering the top-1000 retrieved documents. The assumption is that any relevant documents ranked between 101 and 1000 are likely to be retrieved by some of the other systems within the top-100 documents and therefore assessed for relevance. Consequently, if the document did not have relevance label associated with it, it is deemed non-relevant for the purpose of comparative evaluation. This approach has raised a number of issues which have been further explored in the IR literature.

Pooling bias. While studies showed that the number of pooled documents in the early TREC experiments was sufficient to rank the systems performance reliably, it also transpired that a considerable number of relevant documents remained undiscovered [3]. Thus, several alternative approaches have been suggested in order to judge more relevant documents. For example, Zobel [3] suggested that, instead of applying uniform pooling of documents across systems and queries, the systems with higher performance should contribute more documents to the pool. Otherwise, the highly performing systems that identify more relevant documents within the top-100 get little benefit from the other participating systems. Cormak et al. [4] also proposed the Move-to-Front pooling technique with a variable number of pooled documents across systems. However, in practice, it is difficult to distinguish good or bad performing systems in advance. As a result, the NIST continues to pool the same number of documents from participating systems in the attempt to avoid a possible bias in favor or against any specific type of participating systems.

Weber and Park [1] estimate the bias that the uniform pooling and incomplete judgments introduce when un-judged documents are considered as non-relevant and when they are simply omitted from the computation of the performance scores. For each participating system they consider the discrepancy in a system's performance score when the pool first excludes and then includes documents uniquely retrieved by that system. This provides the *mean absolute error* across all the participating systems that can be used as a correction factor when evaluating new systems for which the set of unique relevant documents is not judged. This relies upon the assumption that the new system is not radically different in the sense that the proportion of the pooled and judged documents vs. its unique documents is similar to other systems.

Weber and Park [1] partially address that issue by considering a more precise error estimation based on a set of *common topics* for which existing systems and a new one are fully assessed. By removing the uncertainty of the un-judged documents they propose an *adjusted estimator* that can be extrapolated to new topics and new systems. Their experiments demonstrate the effectiveness of the estimator with different sizes of common topics sets. However, they do not provide criteria for topic selection nor prioritization of new documents for relevance assessment when these are required to evaluate new systems. Research presented in this paper addresses these issues and explicitly models the query and document selection process in relation to the fixed budget constraints.

Document selection. Many of TREC test collections contain only 50 queries. Using a relatively small query set allows NIST to judge many documents per query and still stay within the available budget for relevance assessments. This increases the reusability of a test collection for other tasks and systems. On average about 2000 documents are judged per query. In total, about 100,000 documents are judged for 50 queries and involve a considerable effort from the assessors. Sanderson and Zobel [6] suggested an alternative and less costly approach. They showed that if NIST evaluated systems by using a significantly larger set of queries, i.e., much larger than sets of 50 queries, and shallower pools of candidate documents, much smaller than 100 documents per query, then the assessors' effort would be greatly reduced without compromising the accuracy of evaluation. Carterette and Smucker [7] supported this suggestion by using statistical tests. The idea of evaluating by large number of queries with shallow judgments motivated a variety of approaches for selecting a subset of documents for assessments and defining evaluation metrics for partially judged result sets, such as statAP [9] or MTC [8].

Following the belief that a larger query set is desirable, the TREC 2007 Million Query track [10] was the first to include thousands of queries. The organizers made use of recent document selection methods to collect few judgments per query. However, due to the small number of documents assessed per query, the reusability of such a test collection still remains questionable [11]. This raises a fundamental question of how many and which documents should be assessed per query to achieve an optimal trade-off between the evaluation accuracy and the limited budget that is available for document assessments. In our work we give a mathematical formulation of this problem that is tractable and extendible to include various refinements

The awareness of cost factors in IR system evaluation and relevance assessments has increased in recent years with the use of crowdsourcing services, such as Mechanical Turk (www.mturk.com) provided by Amazon, to supplement the traditional approaches to relevance assessments. Indeed, relevance assessment tasks can be expressed in terms of Human Intelligence Tasks (HITs) ([5] [12]) and presented to the crowd to solicit relevance labels in return for a specified fee. The direct cost is then captured in the pay to the workers through micropayment facilities that the crowdsourcing services provide. The effectiveness of the crowdsourcing approach is being investigated in terms of various factors ([13] [5]), including the cost overhead caused by redundant relevance assessments that are needed for quality assurance ([14] [15] [5]). Our work can aid such efforts by providing an optimization mechanism that explicitly considers the cost per relevance assessment and provides criteria for selecting query-document pairs that most contribute to the accuracy of the system evaluation.

In summary, a large body of research has focused on various ways to reduce the effort of collecting relevance assessments while maintaining specific qualities of a test collection such as coverage of relevant documents and effectiveness in ranking of participating systems. We extend the existing research in two directions: (1) we devise a method for incrementally acquiring relevance judgment for a test collection and (2) we formulate and evaluate cost optimization problems that control the effectiveness of relevance assessments under the constraint of limited budget.

4. PROBLEM FORMULATION

The main objective of our research is to devise an effective method for growing a test collection that explicitly takes into account the cost of acquiring relevance judgments and aims to maximize the accuracy of system evaluation and extensibility to new, possibly rather different systems.

Let $\boldsymbol{S}$ denote the population of all IR systems. Although the distribution of $\boldsymbol{S}$ is unknown, we assume that all, past, present and future systems are drawn from this distribution. We acknowledge that this is a simplifying assumption but a good starting point for developing the mathematical model.

For a given document corpus $\mathcal{D}$ and a set of N queries$\boldsymbol{Q}_N = \{q_1, q_2, \dots, q_N\}$, we aim to gather relevance judgments and create a test collection that is effective in evaluating the performance of retrieval systems. We assume that there is a set $\boldsymbol{S_L}$ of L participating systems ($\boldsymbol{S_L} \subset \boldsymbol{S}$), each of which returns a number of retrieved documents for each of N queries. From the retrieved documents we create a *common pool* of documents to be used for

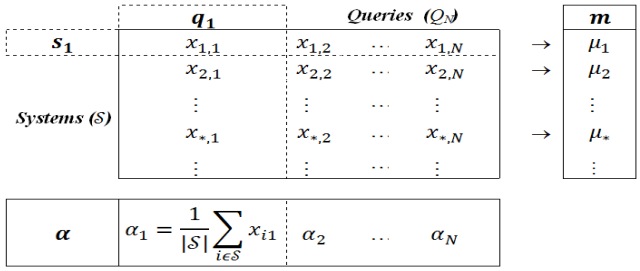

Figure 1. The per-query effectiveness score matrix *X*, for systems in *S* and queries in Q_N. *M* is the vector of average system performance across queries, and α is the (column) vector of average query performance values measured across systems.

comparative evaluation of the systems. Let Ω denote the desired budget required to build complete relevance judgments over the pooled documents. For a given budget *B* that is much smaller than Ω ($B \ll \Omega$), we seek to collect relevance judgments for a subset of query-document pairs in order to accurately evaluate the performance of the participating systems and reliably estimate the performance of yet unseen systems. We propose a two-stage process to allocate the limited budget *B*, which we outline next.

Stage 1. *– Acquire relevance judgments for an initial set of documents*

In the first stage we allocate a portion B_1 of the budget *B* to assess the relevance of some of the documents that are contributed to the common pool by the participating systems $\boldsymbol{S_L}$. A number of methods have been proposed to select documents for relevance assessment e.g. [8]. Generally, the selection methods assign a priority value w_d to each document and process them accordingly.

Given a limited budget, the simplest allocation strategy is to divide the budget equally among *N* queries and, for each query, select a subset of documents with the highest priority scores. In the standard pooling technique that is done by ranking documents based on the query relevance and choosing the uniform pool depth across queries to fit the available budget B_1. Thus, w_d score is 1 for documents above the cut-off rank and, thus, included into the pool and 0 for those that are not.

Stage 2. *– Selectively expand relevance judgments*

In the second stage we utilize the remaining budget, B_2 ($B=B_1+B_2$), to extend the pool of relevance judgments in order to improve the accuracy of the performance metrics for the participating systems and the reusability of the test collection for evaluating new, yet unseen systems.

We formulate the optimization problems that reflect our goal to allocate budget per query and acquire additional relevance judgments in order to achieve maximum agreement with the evaluation of $\boldsymbol{S_L}$ systems using the full set of common documents. Since the distribution of relevant documents across queries may vary, we include the estimated number of relevant documents into the optimization model. This enables us to make allocation of budget per queries that reflect the potential of selecting and assessing documents that are relevant. Increasing the pool of relevant documents has a dual benefit: it improves the evaluation accuracy of existing systems and it enables evaluation of new systems.

Before we describe in detail the method for prioritizing queries and documents, we first introduce the mathematical notation and formulation of the model.

4.1 Concepts and Notation

For the population of all IR systems $\boldsymbol{S}$, we observe the retrieval performance of each system over a finite set of *N* test queries. The performance measurements are represented in the form of a performance matrix $\boldsymbol{X}$, as depicted in Figure 1.

Each row corresponds to a system and each column to a query[1]. An entry $x_{i,j}$ in $\boldsymbol{X}$ denotes the *performance score* of the *i-th* system on the *j-th* query. We refer to a column of the matrix $\boldsymbol{X}$ as a *query-system vector* comprising the performance scores of all the systems for a given query. The column vector $\boldsymbol{m}$ is the average of all query-systems vectors across queries. Let μ denote the average performance across all the queries for a randomly selected system in $\boldsymbol{S}$. If $\boldsymbol{x}$ is the system row in the matrix $\boldsymbol{X}$, then

$$\mu = N^{-1}\boldsymbol{xe}$$

where $\boldsymbol{e} = \{1\}^{N\times 1}$ is the vector of *N* components, each equal to 1. We are interested in the expectation and variance of μ across all the systems. For that we define $\boldsymbol{\alpha} \in R^{1\times N}$ to be the vector of average performance scores for an individual query across the systems, as shown in Figure 1. Further, let $\boldsymbol{\Sigma}$ denote $N \times N$ covariance matrix of the *N* query-systems vectors. Then the expectation and the variance of μ across queries are given by

$$E(\mu) = N^{-1}\boldsymbol{\alpha e}\,, \qquad var(\mu) = N^{-2}\boldsymbol{e^T \Sigma e}$$

In a more general case, the performance μ of a system in $\boldsymbol{S}$ is expressed as a linear combination of the effectiveness scores $x_{i,j}$, associating a weight with each query q_j. Let $\boldsymbol{\beta} \in [0,1]^{N\times 1}$ be a the weight vector with real values in [0,1]. Then the weighted average is expressed as

$$\mu_\beta \cong \boldsymbol{x\beta}$$

and the expectation and the variance of μ_β across queries are given by

$$E(\mu_\beta) = \boldsymbol{\alpha\beta}\,, \qquad var(\mu_\beta) = \boldsymbol{\beta^T \Sigma \beta}$$

We now show that $\boldsymbol{\beta}$ determines the priority scores of queries under specified conditions when expanding relevance judgments in stage 2.

4.2 Prioritizing Query-Document Pairs

In practice, it has been shown that some documents are more effective than others in discriminating systems' performance for a given query (e.g., [8] & [16]). Similarly, some queries are more effective than others in characterizing individual system's performance and facilitating comparative performance of a group of systems [17]. Thus, it is useful to define a query-document priority score s_{qd} as $s_{qd} = w_q \times w_d$ where w_q and w_d are weight coefficients for queries and documents, respectively.

While there are many ways to prioritize documents e.g. [8], for simplicity, we adopt the standard pooling method for the document selection and focus our attention on query prioritization. The prioritization of documents is determined by

[1] For simplicity we shall denote the row and the column vector in the same manner; it will be clear from the context which operation is being performed with the vectors.

the pool depth and is adjusted according to the available budget. Thus, the document weight w_d=1 if a document is in the pool and w_d=0 otherwise.

We consider a query *j* informative if its performance across systems, i.e., the *j-th* column of $\boldsymbol{X}$, is similar to the average performances of systems across all the test queries, i.e., the vector $\boldsymbol{m}$. Our objective is to determine the set of most informative queries based on several criteria. We formalize that by defining the vector $\boldsymbol{m}_\beta \in R^{L\times 1}$ to represent the weighted average performance of the systems across queries. The weights are given by coefficients $\boldsymbol{\beta}$ and $\boldsymbol{m}_\beta[\mathrm{i}] = \mu_{\beta i}$, $i \in \{1, \dots, L\}$, is the weighted average performance of the system i across queries.

In order to determine the coefficients $\boldsymbol{\beta}$ so that vector $\boldsymbol{m}_\beta$ is close to the vector $\boldsymbol{m}$, we consider an objective function $f(\boldsymbol{\beta})$ that defines the distance criteria between $\boldsymbol{m}_\beta$ and $\boldsymbol{m}$. In the context of IR systems evaluation, two criteria naturally present themselves: (i) the similarity in the ranking of the systems and (ii) the similarity in the absolute values of performance, i.e., $\mu_{\beta i} \approx \mu_i$.

When dealing with a group of systems, we often want the relative ordering of systems induced by $\boldsymbol{m}_\beta$ to be close to the ordering induced by $\boldsymbol{m}$. The closeness of two orderings is usually measured using Kendall-τ. Unfortunately, using such a measure leads to computationally inefficient solutions. For example, applying a greedy algorithm results in prioritizing queries that are highly dependent on the set of participating systems and do not generalize to new systems [18]. Consequently, we did not consider this similarity measure further in the context of the query-document prioritization. However, we do use Kendall-τ as an evaluation measure in our experiments to assess the quality of the optimization method.

4.2.1 Performance Score Similarity

There are many ways to characterize similarity in values between $\boldsymbol{m}_\beta$ and $\boldsymbol{m}$ such as the mean square error or correlation. In our experiments and analysis we measure and report on correlation.
The linear correlation measure ρ_β, between μ_β and μ is given by

$$\rho_\beta = \frac{cov\,(\mu,\ \mu_\beta)}{var(\mu)^{1/2} var(\mu_\beta)^{1/2}} = \frac{\boldsymbol{e^T \Sigma \beta}}{(\boldsymbol{e^T \Sigma e})^{1/2}\,(\boldsymbol{\beta^T \Sigma \beta})^{1/2}} \qquad (2)$$

where the covariance between μ and μ_β is given by

$$cov\,(\mu, \mu_\beta) = N^{-1}\,\boldsymbol{E}\{\boldsymbol{e^T(x-\alpha)^T\,(x-\alpha)\beta}\} = N^{-1}\,\boldsymbol{e^T \Sigma \beta}$$

where $\boldsymbol{x} \in R^{1\times N}$ represent a system row in the matrix $\boldsymbol{X}$. We seek a set of $\boldsymbol{\beta}$ coefficients that maximizes ρ_β. Reordering Equation (2) gives

$$\gamma_\beta \equiv (\boldsymbol{e^T \Sigma e})^{1/2} \rho_\beta = \frac{\boldsymbol{e^T \Sigma \beta}}{(\boldsymbol{\beta^T \Sigma \beta})^{1/2}} \qquad (3)$$

Maximizing ρ_β is equivalent to maximizing γ_β since $(\boldsymbol{e^T \Sigma e})^{\frac{1}{2}}$ is a constant. Defining $\boldsymbol{h} \in R^{N\times 1}$ as a column vector where $\boldsymbol{h^T} = \boldsymbol{e^T \Sigma}$, Equation (3) can be re-written as $\frac{\boldsymbol{h^T \beta}}{(\boldsymbol{\beta^T \Sigma \beta})^{1/2}}$ and, hence, its maximum value can be approximated by the minimization problem that is expressed in a quadratic programming form[2] [19]:

$$min_\beta \quad f(\beta) = \boldsymbol{\beta^T \Sigma \beta} - \tfrac{1}{2}\boldsymbol{h^T \beta} \qquad (4)$$

Hence, minimizing $f(\beta)$ is equivalent to maximizing γ_β.

4.3 Constraints

We define the constraints to be used in conjunction with the objective function $f(\boldsymbol{\beta})$ of Equation (4). First we consider a simple constraint that focuses on the budget limit and controls the number of queries that are selected to build additional relevance judgments. The second constraint is intended to increase the coverage of the relevant documents that are assessed and ensure that the selected queries provide effective evaluation of new, previously unseen systems. We refer to the latter as the *generalizability* of the test collection.

During Stage 2, we assume that a fixed budget B_2 is available for relevance judgments. Previous work has not considered a budget constraint in the context of query-document selection. It is natural to assume that, if the query has a high priority score, the allocation of budget would be proportional to the query importance for evaluation. We can, then, without loss of generality, take the query weight β coefficients to represent the proportion of the available budget that will be allocated to individual queries. In other words, if query *j* has a corresponding weight β_j>0, we will expend a proportion of the budget that is a function of β_j and $\sum_{j=1}^{N} \beta_j = \frac{B_2}{\Omega - B_1}$. The number of 'active' queries (β_j >0), is then based on the optimization B_2:

$$min_\beta\ f(\boldsymbol{\beta}) \qquad subject\ to: \begin{cases} \sum_{j=1}^{N} \beta_j = \frac{B_2}{\Omega - B_1} \\ \forall\, j : 0 \le\ \beta_j \le 1 \end{cases} \qquad (5)$$

4.3.1 Generalizability Constraint

If all the relevant documents for each query in the test collection are identified, then the test collection generalizes to any system. Unfortunately, we can guarantee to identify all relevant documents only if we judge all the documents in the collection, which is prohibitively costly. Pooling documents significantly reduces the number of documents we need to judge, as discussed earlier. However, pooling does not guarantee that all the relevant documents have been identified. Clearly, the fewer unidentified relevant documents in the test collection, the more generalizable the test collection is. Thus, we define a cost function that not only minimizes the difference between $\boldsymbol{m}_\beta$ and $\boldsymbol{m}$, but also minimizes the number of un-judged relevant documents.

Let r_j be the expected number of un-judged relevant documents for query q_j. Given that we allocate $\beta_j$of the B_2 budget to a query q_j then at the end of the second stage, the number of newly judged relevant documents will be proportional to $\beta_j r_j$. The total number of relevant documents judged in the second stage is simply $\sum_{j=1}^{N} \beta_j r_j$, ignoring the constant of proportionality. Clearly, we want to maximize the total number of relevant documents in order to achieve maximum generalizability. Using a Lagrange multiplier λ we combine the constraint and the objective function $f(\boldsymbol{\beta})$ to obtain

$$\min_\beta \left[\tfrac{1}{2} f(\beta) - \lambda \sum_{j=1}^{N} \beta_j r_j \right] \text{ subject to: } \begin{cases} \sum_{j=1}^{N} \beta_j = \frac{B_2}{\Omega - B_1} \\ \forall\, j : 0 \le\ \beta_j \le 1 \end{cases} \qquad (6)$$

The above optimization function is convex and we solve it using a sequential quadratic programming algorithm [20]. When the

[2] The optimization form in Equation 3 is in *convex-fractional form* and is optimized by transferring it to quadratic programming form [19].

Lagrange multiplier $\lambda = 0$, the formulation (6) is reduced to (5). Of course, the above discussion begs the question of how to estimate r_j. This is discussed in Section 4.3.

5. IMPLEMENTATION DETAILS

Before describing the experiments, we discuss a number of implementation issues. Note, however, that the setting of λ is discussed in Section 5.

5.1 Random Sampling of Systems

In practice, the mean vector $\boldsymbol{\alpha}$ and covariance matrix $\boldsymbol{\Sigma}$ are unknown to us because $\boldsymbol{S}$ is unknown, i.e. there is no information about unseen systems. Instead, we have a sample of $x_1, \ldots, x_L$ of multivariate scores (x_i is a row of matrix $\boldsymbol{X}$) of $\boldsymbol{L}$ participating systems by which we can estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$. When the set of participating systems is uniformly sampled from the population of systems, $\boldsymbol{S}$, the standard unbiased estimators of $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$, denoted as $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\Sigma}}$, are given by

$$\widehat{\boldsymbol{\alpha}} = \overline{\boldsymbol{x}} \equiv L^{-1}\sum_{i=1}^{L}\boldsymbol{x_i} \ , \quad \widehat{\boldsymbol{\Sigma}} = (L-1)^{-1}\sum_{i=1}^{L}(\boldsymbol{x_i}-\widehat{\boldsymbol{\alpha}})^{\boldsymbol{T}}(\boldsymbol{x_i}-\widehat{\boldsymbol{\alpha}})$$

If L is large and the sample of participating systems forms a diverse set of retrieval systems, we can get reliable estimations of $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$.

5.2 Non-random Sampling of Systems

The unbiased estimators ensure that all participating systems contributing equally to estimate priority scores $\boldsymbol{\beta}$. Hence, the unbiased estimators provide maximum likelihood estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$ when a new IR system is randomly sampled from $\boldsymbol{S}$. However, in practice, the new systems may not be considered as drawn from a random sample. They may, in fact, be variations and improvements of the already participating systems. In that case, allowing poor and good performing systems to contribute equally to prioritizing queries may not result in the best choice. Instead, better performance may be achieved by prioritizing queries based on participating systems that are similar to the new system.

Unfortunately, no information regarding new systems is available during the construction of the test collection. However, if we assume that new systems will have high performance, we can weigh the participating systems such that higher performing participating systems contribute more to the prioritization of queries. In this case, we can use biased estimators to approximate $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$, as explained next.

Let $\{p_1, p_2, \ldots, p_L\}$ be a set of weights assigned to the L participating systems such that p_i indicates the degree of contribution for *i-th* participating system in prioritizing queries and $\sum_{i=1}^{L} p_i = 1$. The biased estimators of $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$ are then

$$\hat{\alpha} = \sum_{i=1}^{L} x_i p_i \ , \quad \hat{\Sigma} = \frac{1}{(1-\sum_{i=1}^{L} p_i^2)}\sum_{i=1}^{L}(x_i-\hat{\alpha})^T(x_i-\hat{\alpha})\,p_i$$

In Section 5.3, we describe a simple selection method for weights p_i and investigate the use of biased estimators with a set of new systems that are expected to have higher performance than participating systems.

5.3 Estimating the Number of Unseen Relevant Documents

It is difficult to determine whether or not all relevant documents for a query have been judged. However, the prior work of Zobel [3] suggests that some degree of estimation is possible, given an initial set of relevance judgments. Experimental results in [3] demonstrated high prediction accuracy when estimating the *total* number of unseen relevant documents retrieved for all queries in a test collection. However, when predicting relevant documents for a single query, there was a large uncertainty in the estimates.

Given a set of initially judged documents, Carterette et al. [21] applied logistic regression to calculate the probability of relevance of unjudged documents. We use the same method to partition unjudged documents into relevant and non-relevant categories. Specifically, given an initial set of judged documents for a query, the relevance of a document d_i to query q_j is estimated by:

$$R(d_i, q_j) = \frac{1}{1+\exp\left(-\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{F}\right)}$$

where $\boldsymbol{w}$ is the parameter vector of the model and $\boldsymbol{F}$ is a feature vector. The feature vector uses the same features as introduced in [21].

In order to train the model, we first extract features from each of the judged documents. The output of the model for a judged document is 0 when the document is non-relevant and 1 when the document is relevant. After learning the parameter vector $\boldsymbol{w}$ by using the logistic regression, the trained model is used to estimate the relevance of unjudged documents. For an unjudged document d_i, retrieved for query q_j, we label it as relevant if the probability of relevance $R(d_i, q_j) > 0.5$; otherwise d_i is labelled as non-relevant. Hence, the expected number of relevant unseen documents for queries j is the number of unjudged documents with $R(d_i, q_j) > 0.5$.

6. EXPERIMENTAL EVALUATION

In this Section, we describe a set of experiments that we conducted in order to evaluate our two stage approach to creating effective test collections. The novelty of our work is in the explicit modeling of the budget constraints for relevance assessments and prioritization of queries to maximize the accuracy of the system performance. In our experiments

1. We apply the optimization model expressed in (6), considering both the accuracy of evaluating existing systems and expanding the set of known relevant documents to, secure generalization to new systems. The model in (6) requires
   a. Computing the Lagrange Multiplier for the collected data.
   b. Estimation of the number of relevant documents *r* for a given query (Section 4.3).
2. We compare all our results with three benchmark systems that promote specific budget allocation method: uniform allocation, random allocation, and score adjustment allocation.

In the evaluation of our approach we consider both the accuracy of the performance evaluation with additional relevance judgments and generalization to unseen systems. The latter requires careful characterization of the systems. We differentiate between (1) new systems that are similar to the participating one, i.e., drawn from the homogeneous set of systems and (2) new systems that are markedly different, i.e., drawn from a heterogeneous set of systems. This has two implications for our experimental work:

- The statistical treatment is different for homogeneous and heterogeneous systems; the latter requires the use of biased estimators of $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$.

- We need to define the criteria for identifying radically different systems. In our work we examine the *mean average reuse* (MAR) of individual systems and select a group of 'new' systems that have low MAR. Furthermore, we use manual runs as new systems since they are markedly different from the automatic ones.

Finally, all the evaluations are conducted by comparing the standard ranking of the systems, based on the full set of queries and relevance judgments, with the system ranking obtained based on the weighted query ranking and incremental relevance judgments supplied by our method. The comparison is based on two metrics: Kendall-t and RMSE (Root Mean Square Error).

6.1 Benchmarks

Our experiments assume that there is a budget *B* available to build relevance judgments. The budget is expressed in terms of the number of documents judged. We divide the budget in two parts, corresponding to the two-phase approach: (i) B_1, which is uniformly allocated between all queries in the test collection, and (ii) B_2: which is allocated based on our *Query Prioritization* (*QP*) and resource allocation method. In order to avoid any bias against individual systems, all participating systems contribute equally to the pooling of documents in both phases. We consider three resource allocation methods as baselines for comparison with *QP*:

(i) ***Uniform Allocation (UN)***, in which available budget is uniformly allocated across queries. For example, if the budget can cover only 200 new judgments and there are 100 queries, we judge two new documents per query.

(ii) ***Random Allocation (RA)***, in which a random set of *n* queries is selected and the budget B_2 is uniformly allocated across the selected queries. In our experiments we use *n* that corresponds to the number of queries selected by our optimization method. We repeat the random query sampling for 1000 trials and report the average of the corresponding results.

(iii) ***Score Adjustment (SA)***, in which a random set of *n* queries is selected and the budget B_2 is uniformly allocated across the selected queries. Once the new relevance judgments are acquired, one can compare the difference between the original and new performance metrics and use the average bias as a correction term for both queries and systems, as proposed by Webber and Park [1]. Note that in the original algorithm by Webber and Park [1] it is assumed that relevance judgments are rendered for documents retrieved by new systems, not participating in the construction of the common pool of documents. However, in our context, we implemented score adjustment based on relevance judgments of documents within the common pool of participating systems. Our task is focused on incrementally building relevance judgments for the commonly pooled set of document. In our experiments, we apply the ***SA*** method for 1000 trials of random query sampling and report the average of the corresponding results.

All three baseline methods are compared with the ***Query-Document prioritization (QP)*** outlined in Section 3.2. The results can be found in Table 1 and 2.

6.2 Data Sets and Parameter Settings

Our experimental investigations were performed using two test collections: i) the TREC 2004 Robust track and (ii) the TREC-8 test collection. Normally, organizations participating in TREC register as *sites* and submit a number of experimental *runs* for evaluation. These runs often represent variations of the same IR system. For our purposes we consider each *run* as an individual IR system but take special care when considering IR systems from the same site. In particular, when experiments require that we exclude some of the systems in order to treat them as new, yet unseen systems, we hold out not only individual runs but the entire set of runs from the same site. Furthermore, during the computation of performance metrics, we remove documents that are uniquely retrieved by the held-out systems when that is required.

The TREC 2004 Robust track consists of 249 queries, 14 sites with a total of 110 automatic runs, and 311,410 relevance judgments. Since all the submissions are automatic runs, we treat them as a homogeneous set of systems, drawn from the same distribution.

The TREC-8 consists of 50 queries, 39 sites with 13 manual runs and 116 automatic runs, and 86,830 relevance judgments. Automatic runs use automatic query formulation, while manual runs allow human to formulate queries. The latter queries typically perform better. Because of the existence of both automatic and manual runs, we treat TREC-8 as a heterogeneous set of systems, in the sense that they are *not* all drawn from the same distribution. Thus, we apply the biased estimator of Section 4.2 when defining the set of new systems to consist of the manual runs. Both test collections use TREC Disks 4 & 5, excluding the Congressional Record sub-collection.

Comparative evaluation of TREC runs is conducted based on the *average precision* (AP). However, since we use an incomplete set of relevance judgments at Stage 1, many documents remain unjudged. Consequently, the AP scores measured for participating systems are uncertain and the performance matrix $\boldsymbol{X}$ is noisy. For that reason, in our experiments we use *infAP* rather than AP to measure systems effectiveness with respect to initial judgments. The *infAP* scores provide a better approximation of the true AP scores [22] and, hence, a less noisy performance matrix *X*. In addition, *infAP* allows us to measure the confidence interval for estimates of a system's performance. This helps investigate the minimum budget required to evaluate systems with a high confidence.

6.3 Experimental Setup

In order to test the generalization and robustness of the three methods to evaluate new systems, we first divide the TREC runs into participating systems and new, still unseen systems that contribute new search results. To simulate Stage 1, we randomly select a few sites and use their corresponding runs as participating IR systems. Using the pooling technique we select and evaluate the set of documents retrieved by these participating systems. The pool depth is adjusted to fit the budget allocated to the Stage 1.
We split the held-out systems into two groups. For each held-out system, and each query, we compute the *average reuse* (AR) [21]. This measures the overlap between the documents retrieved by a held-out system and the judged documents. The average reuse for query *q* is defined as:

Table 1. Result for Robust TREC 2004 runs evaluated by MAP. The first two columns report experimental parameters. The next columns report the Kendall-τ and Root Mean Square (RMSE) of (i) participating systems, and (ii) previously unseen systems for each resource allocation.

#	(s_1, s_2) %	(B_1, B_2) $\times 10^3$	Kendall-τ								RMSE							
			participating systems				new systems				participating systems				new systems			
			UN	*RA*	*SA*	*QP*	*UN*	*RA*	*SA*	*QP*	*UN*	*RA*	*SA*	*QP*	*UN*	*RA*	*SA*	*QP*
1	(10, 50)	(2,8)	0.63	0.58	0.65	0.68	0.54	0.51	0.59	0.58	0.17	0.21	0.14	0.15	0.19	0.27	0.15	0.16
2		(5,5)		0.61	0.7	0.78		0.52	0.66	0.71		0.19	0.12	0.1		0.25	0.12	0.11
3		(8,2)		0.63	0.67	0.79		0.52	0.63	0.74		0.18	0.1	0.09		0.24	0.11	0.11
4	(10, 40)	(4,16)	0.72	0.66	0.76	0.9	0.68	0.62	0.7	0.76	0.137	0.17	0.09	0.04	0.12	0.21	0.12	0.1
5		(10,10)		0.68	0.79	0.89		0.65	0.77	0.81		0.15	0.07	0.04		0.17	0.09	0.08
6		(16,4)		0.74	0.81	0.91		0.67	0.74	0.83		0.15	0.08	0.03		0.15	0.1	0.09
7	(20, 40)	(4,16)	0.79	0.69	0.83	0.91	0.8	0.66	0.74	0.84	0.12	0.15	0.11	0.06	0.1	0.22	0.1	0.09
8		(10,10)		0.75	0.82	0.89		0.67	0.8	0.9		0.14	0.08	0.04		0.19	0.08	0.06
9		(16,4)		0.77	0.83	0.91		0.7	0.81	0.91		0.11	0.1	0.05		0.15	0.09	0.05

$$AR(q) = \left(\frac{1}{judged\ (q)}\right)\sum_i \left(\frac{judged\ @\ i\ (q)}{i}\right)$$

where *judged @i (q)* is the number of judged documents in the top i results retrieved by a held-out system for query *q*, and *judged (q)* is the total number of documents judged for query *q*. We then define the *mean average reuse* (MAR) for a held-out system as the average of AR values over the full set of queries.

Based on the MAR values, we split the held-out systems into two groups. The first group consists of systems with high MAR across runs. These systems can be evaluated using the existing relevance judgments. The second group, referred to as the *new* set, consists of runs that have low MAR. These systems require additional relevance judgments in order to be evaluated.

The full experiment comprises the following steps:

1. Pick s_1 percent of sites at random, these are the *held-in* sites.
2. For each query, construct the training pool of the top k_0 documents using documents retrieved by the held-in runs and collect the associated relevance judgments. Compute the performance matrix $\boldsymbol{X}$. The value of k_0 is determined based on the budget allocated to Stage 1. The budget is uniformly distributed across queries.
3. Compute the MAR for the held-out runs. Average the MAR scores across runs from the same site and produce average reuse score for each site.
4. Pick s_2 percent of sites with low MAR scores and treat their runs as *new* systems. The remaining runs are evaluated with the existing relevance judgments and their performance values are added to the matrix $\boldsymbol{X}$. Note, however, that the remaining runs do not contribute to the document pool.
5. Prioritize queries using *QP* method (using the optimization defined in Equation 6).
6. For the *RA* and *SA* method, given that *n* queries are activated at step 5 (have non-zero β coefficients), randomly select a subset of *n* queries from the total set of *N*.
7. Given the budget B_2 acquire additional relevance judgments for documents pooled by participating systems in one of the four ways:

   (i) *Uniform* (*UN*): For each of the $\boldsymbol{N}$ queries, acquire relevance judgments for an additional k_1 documents, where k_1 is adjusted based on B_2.

   (ii) Random Allocation (*RA*): for a random sample of *n* queries acquire relevance judgments for additional k_2 documents per query, where $n \times k_2 = N \times k_1$.

   (iii) Score Adjustments (*SA*): for a random sample of *n* queries acquire relevance judgments for additional k_2 documents per query, where $n \times k_2 = N \times k_1$. Apply score adjustment.

   (iv) Query-Document Optimization (*QP*): order the query-document pairs and acquire relevance judgments for the $N \times k_1$ pairs with the highest priority scores.

6.3.1 Lagrange Multiplier

The *QP* formulation of the budget optimization in (6) requires the computation of the Lagrange multiplier λ. We determine λ empirically by systematic exploration of the range of values for λ, $0 \leq \lambda \leq 10$. This is performed after Stage 1 but before Stage 2. After Stage 1, we have allocated budget B_1 and acquired the same number of relevance judgments for all queries. We then simulate the steps 1 through 7 above, where we split the budget B_1 into two parts B_1' and B_2' in the same proportion as true budget allocation B_1 and B_2. Note that during this simulation the estimated number, r_j of un-judged relevant documents for query q_j is set to the number of relevant documents identified at Stage 1 using budget B_1 for query q_j. This ensures that at Stage 2 of the simulation to determine λ, no selected query requires more assessments than we have acquired during Stage 1. Thus, we have all the relevance judgments needed to evaluate the performance of the simulation.

For a particular value of λ within the range $0 \leq \lambda \leq 10$ we apply a 10-fold cross-validation technique. In each of the 10 iterations, 10% of participating systems are held out (these become our simulated new systems). Relevant documents that are in the initial document pool but solely retrieved by the held-out systems are removed from the pool. The *QP* method, using the reduced set of judgements, produces a set of query-document pairs. We evaluate this solution by computing the Kendall-τ of the held-out systems ranks with the corresponding systems ranks using all the

relevance judgments acquired using budget B_1 and Stage 1. We record the average Kendall-τ for the 10 trials. Finally, we choose the λ value with the highest average Kendall-τ.

6.4 Experimental Results

Our experimental results are divided into two parts, following the separate treatment of the homogenous and heterogeneous sets of IR systems. Thus, in Section 5.4.1 we consider the homogeneous collection of Robust TREC and use unbiased estimators for $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$ in our *QP* algorithm. For the Robust TREC collection we report experiments using a total budget that covers either 10,000 or 20,000 relevance judgments. This is less than 7% of the collection's assessor budget of 311,410 relevance judgments.

In Section 5.4.2 we present experiments with the heterogeneous collection of TREC-8 and use manual as new systems. For the TREC-8 collection we report results using a total budget that covers either 2,000 or 4,000 relevance judgments. This is less than 5% of the assessor budget that cover 86,830 relevance judgments for the collection. In the implementation of *QP* we use the biased estimators $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$ introduced in Section 4.2.

6.4.1 Homogeneous Systems

We applied the steps 1 through 7 in Section 5.3 across 10 trials and, in each trial we randomly choose s_1 percent of sites and associated runs as participating systems. The remaining runs are evaluated for MAR and the s_2 percent of sites with the lowest MAR scores are chosen to be *new* systems. Depending on the average MAR scores, s_2 varies between 50% and 40% of the total number of sites. We report averages over the 10 trials.

We repeated the experiment for 3 different values of s_1 and s_2, and 3 different budget allocations, B_1 and B_2. Table 1 summarizes the results.

We report the Kendall-τ statistic between the ranking of the systems induced by a resource allocation method, and the ranking scores of the systems over the full set of queries and the original document pools. We report separate Kendall-τ statistics for participating systems and for new systems, which is common in the literature and permits us to separately discuss the accuracy and generalization of the methods. We also report root mean square error (RMSE) between the MAP scores of the systems based on query-document selection and the *true* MAP scores measured over the fully assessed documents from the common pool. Once again, separate scores are provided for participating and new systems.

We observe that for all 9 experimental configurations, the Kendall-τ scores of the *QP* method outperform the other three resource allocation methods. Note that the uniform allocation strategy is comparable and often better than the random allocation strategy for both participating and new systems. The score adjustment (SA) method outperforms the uniform allocation when s_1=10% (rows 1 through 6). However, when for s_1=20 %, the *SA* method performs no better than a uniform allocation for new systems, but remains better for participating systems. In contrast, our *QP* method is superior in all cases, except for configuration #1, in which the initial budget B_1 is only 2000 relevance judgments. We believe this is due to the small value of B_1 which only covers 0.6% of the total assessor judgments.

It is important to note that the *QP* method has significantly better Kendall-τ scores than the random allocation method, for both participating and new systems, indication that the optimization achieved both accuracy and generalizability.

We note that, increasing the number of participating systems s_1 with the same budgets B_1 and B_2 leads to a larger improvement in Kendall-τ of *new* systems' ranking than increasing the budgets, i.e., relevance assessments and keeping the number of participating systems s_1 constant.

This can be seen by comparing experimental configurations 5 & 8 or 6 & 9. These results are probably related to observations by Carterette et al. [21] that a higher diversity of participating systems results in a better ranking of new systems.

Table 1 also reports RMSE values for the various methods. Similar observations hold true here.

Note that the *QP* method is not optimized for RMSE and it can be improved by simply replacing the correlation-based similarity measure with a mean square error measure. We have performed such experiments but space limitation precludes the inclusion and in-depth discussion of results. We just note that RMSE score were improved at the expense of slight degradations in Kendall-τ scores.

We note that in the experiments conducted in this section, the set of participating and new systems were randomly chosen. Hence, both partitions contained both good and poor performing systems. We therefore used unbiased estimators of the mean vector $\boldsymbol{\alpha}$ and covariance matrix $\boldsymbol{\Sigma}$, as explained in Section 4.1. In the next section we consider the scenario in which participating and new systems are not randomly chosen. Rather, we consider a set of highly performing systems as new systems and use the biased estimators discussed in Section 4.2.

6.4.2 Heterogeneous Systems

The TREC 8 test collection consists of 129 runs of which 13 runs are manually tuned and outperform the automatic runs. In this test collection 11 best performing runs are all manual and their performance measured by MAP is statistically significantly better than the remaining runs. We consider the 13 manual runs as new (unseen) systems and the rest as participating systems. We consider two variants of our *QP* resource allocation method. The first is our standard method *QP* for which unbiased estimators are used to approximate the mean vector $\boldsymbol{\alpha}$ and covariance matrix $\boldsymbol{\Sigma}$. The second method, *QP'* uses biased estimators, as explained in Section 4.2. Thus, when using the *QP'* method, participating systems contribute non-uniformly in prioritizing queries. The intuition is that, since new systems are likely to perform better than participating systems, we may achieve better accuracy and generalization of new systems, if we preferentially weigh highly performing participating systems.

We use a simple weighting function by which in Stage 1 all automatic systems equally contribute to the document pool that is initially judged. Next, we select the k participating systems with the highest mean *infAP* scores. If the i-th system is among the selected ones the corresponding weight is $p_i = \frac{1}{k}$, otherwise $p_i = 0$. Further, when collecting relevance judgements at Stage 2, only systems with $p>0$ participate in pooling documents[3].

[3] Excluding systems with $p=0$ from pooling documents at Stage 2 may result in underestimating their performance. However, this is not a concern here since our goal is to evaluate a set of new systems that neither participate in pooling at Stage 1 nor at Stage 2.

Table 2. Results for TREC-8 when the 13 manual runs are treated as new (unseen) systems and 116 automatic runs are treated as participating systems. The *QP'* is the extension of *QP* method in which the biased estimators are used to approximate mean vector α and covariance matrix Σ.

#	(B_1, B_2) $\times 10^3$	Kendall-τ										RMSE									
		participating systems					new systems					participating systems					new systems				
		UN	*RA*	*SA*	*QP*	*QP'*	*UN*	*RA*	*SA*	*QP*	*QP'*	*UN*	*RA*	*SA*	*QP*	*QP'*	*UN*	*RA*	*SA*	*QP*	*QP'*
1	$\left(\frac{1}{2},\frac{3}{2}\right)$		0.55	0.71	0.78	0.8		0.25	0.30	0.32	0.54		0.22	0.14	0.09	0.1		0.48	0.45	0.47	0.3
2	(1,1)	0.61	0.57	0.75	0.8	0.81	0.32	0.27	0.44	0.39	0.63	0.17	0.19	0.11	0.07	0.09	0.46	0.46	0.39	0.4	0.27
3	$\left(\frac{3}{2},\frac{1}{2}\right)$		0.6	0.76	0.82	0.83		0.28	0.39	0.39	0.67		0.18	0.12	0.07	0.08		0.42	0.41	0.4	0.24
4	(1,3)		0.65	0.84	0.9	0.92		0.48	0.47	0.50	0.78		0.17	0.1	0.06	0.08		0.38	0.3	0.27	0.2
5	(2,2)	0.86	0.69	0.83	0.89	0.91	0.69	0.49	0.62	0.68	0.87	0.14	0.14	0.09	0.05	0.09	0.34	0.35	0.26	0.25	0.16
6	(3,1)		0.75	0.84	0.9	0.92		0.51	0.66	0.69	0.91		0.16	.08	0.06	0.06		0.36	0.27	0.24	0.16

In our experiment, we arbitrarily set k=30 since (i) it was sufficiently large to approximate **α** vector and **Σ**, and (ii) retaining only 30 runs from 116 ensures that the retained runs have relatively good performance. We repeated the experiment for 6 different budget configurations. The results are shown in Table 3. For all budget configurations, the *QP'* method exhibits the best Kendall-τ scores for both participating and new systems.

Interestingly, even for participating systems, the biased *QP'* method is observed to perform best, although the difference between the biased and unbiased estimators is small. For new systems, there is a much larger difference in performance, with the biased estimator, *QP',* clearly performing much better. Nevertheless, both the *SA* and unbiased *QP* methods exhibit performance that is better than random allocation. Note, however, that for budget configurations 4-6 (total budget 4000 relevance judgments), a uniform allocation strategy performs better than score adjustment (*SA*).

Table 2 also shows that the biased *QP'* method has the lowest RMSE values for new systems throughout the configurations though for participating systems, the *QP* method outperforms *QP'*.

Note, however, that while the RMSE values for participating systems are comparable with Table 1, the values for new systems are considerably larger. We believe this is due to the fact that, when treating manual runs as new systems, many relevant documents are absent from the document pools. In fact, the manual runs retrieve 24% of the unique relevant documents that were judged in the original document pools. Hence, even after judging all documents returned by participating systems, we are unable to accurately measure the absolute performance scores of manual systems.

7. DISCUSSION AND FUTURE DIRECTIONS

The construction of a test collection requires acquisition of a set of user relevance judgments. The number of such judgments is limited by the budget available to construct the test collection. Typically, this budget is uniformly allocated across the queries in the test collection. In this paper we consider the problem of prioritizing query-document pairs for relevance assessment in order to (i) improve the accuracy of evaluating participating systems, and (ii) ensure that the test collection generalizes to new, previously unseen systems. As a result of the optimization method we arrive at

- Relevance assessment method that tailors the number of assessments per query as opposed to the standard approach of uniform allocation of relevance assessments across queries.

In the paper we illustrate a two-stage procedure. In the first stage, we allocate a budget B_1 uniformly across all queries, acquiring a corresponding set of relevance judgments. In the second stage, we use the information gained in the first stage to prioritize query-document pairs and allocate a budget B_2 accordingly. However, it is important to emphasize that

- Our method is iterative in nature and can be applied to support a growing set of new systems and the corresponding collection of relevance assessments.

As we have demonstrated, the method is successfully applied to prioritization of the relevance assessment in the common pool of documents to address the generalization to new systems. This is different from the scenario in [5] where the second round budget is spent to judged documents returned by new systems.

Through a systematic and careful treatment of the system sampling and constraint based formulation of the query selection, our work provides several unique contributions:

- Modeling query and document selection through explicit cost optimization
- Formulating the problem as a convex optimization for which computationally efficient algorithms exist to identify the optimum solution.
- Consideration of the biased sampling of systems and appropriate use of similarity measure for a biased estimator

Our experimental set up compared the *QP* algorithm with, uniform, random sampling and a variant of the score adjustment method presented in [1]. It provided strong evidence that

- Our method is (i) superior to the selected benchmark methods, (ii) exhibits good accuracy, i.e., predicts the performance of participating systems, (iii) exhibits good generalization, i.e., predicts the performance of new systems.

There are various avenues for future work. One of the main advantages of our method is its extensibility. Our work demonstrates how one might incorporate further criteria into our objective function. Furthermore, there have been many recent papers studying characteristics of queries that might make them *better* for use in an evaluation set. By encoding such desirable characteristics as components and constraints within our optimization framework, the method to identify a set of queries that embodies our requirements is a simple process. Thus, our future work will investigate a richer set of such heuristics towards the aim of producing a test collection construction method that is efficient (in terms of resources required for the collection to be compiled) and effective (in terms of accuracy of evaluations).

Furthermore, the experimental set up can be expanded to examine the sensitivity of the algorithm to errors in estimating the number of un-judged relevant documents and investigating the sensitivity to other errors such as errors in matrix $\boldsymbol{X}$ due to uncertainty of *infAP* scores.

Finally, the full potential of the method would be realized through an effective iterative model of relevance assessment. Thus, it would be interesting and beneficial to extend and evaluate the dynamic and real time application of the cost optimization in the context of the emerging practice of crowdsourcing relevance assessments.

8. REFERENCES

[1] W. Webber and L. A. F. Park, "Score Adjustment for Correction of Pooling Bias," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, Boston, 2009, pp. 444-451.

[2] K. Sparck-Jones and C. J. van Rijsbergen, "Information retrieval test collections," Journal of Documentation, vol. 32, no. 1, pp. 59-72, 1976.

[3] J. Zobel, "How reliable are the results of large-scale information retrieval experiments," in Proceeding of ACM SIGIR Special Interest Group on Information Retrieval, 1998, pp. 307-314.

[4] G. Cormak, C. Palmer, and C. Clarck, "Efficient Construction of large test collections," in Proceeding of ACM SIGIR Special Interest Group on Information Retrieval, 98, pp. 282-289.

[5] P. Welinder and P. Perona, "Online crowdsourcing: rating annotators and obtaining cost-effective labels," in CVPR'10: IEEE Conference on Computer Vision and Pattern , 2010, pp. 1526-1534.

[6] M. Sanderson and J. Zobel, "Information retrieval system evaluation: effort, sensitivity, and reliability," in Proceeding of ACM SIGIR Special Interest Group on Information Retrieval, 2005, pp. 162-169.

[7] B. Carterette and M. D. Smucker, "Hypothesis testing with incomplete relevance judgments," in the Sixteenth ACM Conference on Information and Knowledge Management, Lisbon, 2007, pp. 643-652.

[8] B. Carterette, J. Allan, and R. Sitaraman, "Minimal test collections for retrieval evaluation," in Proceeding of ACM SIGIR Special Interest Group on Information Retrieval, 2006, pp. 268-275.

[9] J. A. Aslam, V. Pavlu, and E. Yilmaz, "A Statistical Method for System Evaluation Using Incomplete Judgments," in Proceeding of ACM SIGIR Special Interest Group on Information Retrieval, 2006, pp. 541-548.

[10] J. Allan, J. A. Aslam, V. Pavlu, E. Kanoulas, and B. Carterette, "Overview of the TREC 2007 million query track," in Notebook Proceedings of TREC 2007.

[11] B. Carterette, E. Kanoulas, V. Pavlu, and H. Fang, "Reusable Test Collection Through Experimental Design," in Proceeding of ACM SIGIR Special Interest Group on Information Retrieval, Geneva, 2010, pp. 67-73.

[12] P. Welinder, B. Steve, and B. Serge, "The Multidimensional Wisdom of Crowds," in Proceeding of The Neural Information Processing Systems (NIPS), 2010, p. 2424–2432.

[13] N. Stefanie and S. Ruger, "How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation," in Proceedings of the international conference on Multimedia information retrieval, Philadelphia, Pennsylvania, USA, 2010, pp. 557-566.

[14] M. Winter and W. Duncan J, "Financial incentives and the "performance of crowd," in Proceedings of the ACM SIGKDD Workshop on Human Computation, 2009, pp. 77-85.

[15] R. Snow, B. O'Connor, D. urafsky, and A. Y. Ng, "Cheap and fast---but is it good?: evaluating non-expert annotations for natural language tasks," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 2008, pp. 254-263.

[16] E. Yilmaz, E. Kanoulas, and J. A. Aslam, "A Simple and Efficient Sampling Method for Estimating AP and NDCG," in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, 2008, pp. 603-610.

[17] J. Guiver, S. Mizzaro, and R. Stephen, "A few good topics: Experiments in topic set reduction for retrieval evauluation," ACM Transactions of Information Systems, vol. 27, no. 4, pp. 1-26, 2009.

[18] S. Robertson, "On the contributions of topics to system evaluation," in Advances in Information Retrieval, 33th European Conference on IR Research (to appear), 2011 .

[19] W. Dinkelbach, "On nonlinear fractional programming," Management Science, vol. 13, no. 7, pp. 492-498, Mar. 1967.

[20] R. W. Cottle, J.-S. Pang, and R. E. Stone, The linear complementarity problem. Boston, London: Academic Press Inc, 1992.

[21] B. Carterette, E. Gabrilovich, V. Josifovski, and D. Metzler, "Measuring the Reusbility of Test Collections," in Proceeding of ACM International conference on Web Search and Data Mining, New York, 2010, pp. 231-240.

[22] E. Yilmaz and J. Aslam, "Estimating average precision with incomplete and imperfect judgments," in Proceedings of the 15th ACM international conference on Information and knowledge management, 2006, pp. 102--111.