Prioritizing Relevance Judgments to Improve the Construction of IR Test Collections

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

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

{m.hosseini, ingemar}@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 to prioritize the queries and remaining pooled documents for further relevance assessments. The query prioritization is formulated as a convex optimization problem, thereby permitting efficient solution and providing a flexible framework to incorporate various constraints. Query-document pairs with the highest priority scores are evaluated using the remaining budget. We evaluate our resource optimization approach on the TREC 2004 Robust track collection. We demonstrate that our optimization techniques are cost efficient and yield a significant improvement in the reusability of the test collections.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *selection process*

General Terms: Performance

1. INTRODUCTION

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. Instead, the relevance assessments are typically obtained for a subset of documents retrieved by the set of participating IR systems. Each participating system contributes a fixed number of retrieved documents to the common pool which are then assessed for relevance by expert judges. While this approach increases the chance of identifying relevant documents, economic constraints may still prevent exhaustive judgments of all the documents in the pool.

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 unique contributions of this paper are (i) explicitly incorporating a cost constraint within the optimization, (ii) formulating the optimization problem as a convex optimization, leading to computationally efficient algorithms for finding a globally optimum solution, and (iii) incorporating a *generalization constraint* based on the estimated number of un-judged relevant documents for each query.

In Section 2 we discuss related work. Particular attention is given to the work of Webber and Park [1], to 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 the TREC 2004 Robust test collection. Finally, Section 6 provides a summary of our results and suggestions for future research directions.

2. RELATED WORK

Document pooling, originally proposed by Spärck-Jones and Van Rijsbergen [2] has been widely adopted for the construction of test collections. 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] and Cormack et al. [4].

Webber and Park [1] estimated 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. They also considered a more precise error estimation by considering a set of *common topics* and newly introduced systems for which they had full assessments. By removing the uncertainty of the un-judged documents they proposed 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.

Many 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. Sanderson and Zobel [5] 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

CIKM'11, October 24-28, 2011, Glasgow, Scotland, UK.

candidate documents, i.e., much smaller than 100 documents per query, then the assessors' effort would be greatly reduced without compromising the accuracy of evaluation. Carterette and Smucker [6] supported this suggestion by using statistical tests. Evaluation based on a large number of queries with shallow judgments motivated a variety of approaches for selecting documents for assessments and defining evaluation metrics for partially judged result sets, such as statAP [7] or MTC [8].

Following the belief that a larger query set is desirable, the TREC 2007 Million Query track [9] 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 [10]. 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.

3. PROBLEM FORMULATION

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. This is a simplifying assumption but a good starting point for developing the mathematical model.

We are given a document corpus $\mathscr{D}$ and a set of N test queries $\boldsymbol{Q_N} = \{q_1, q_2, \dots, q_N\}$. 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* $\mathscr{P}$, ($\mathscr{P} \subset \mathscr{D}$) of documents to be used for comparative evaluation of the systems. Let Ω denote the desired budget required to build relevance judgments over the pooled documents $\mathscr{P}$. 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* $\mathscr{P}$*.*

In the first stage we allocate a portion B_1 of the budget B to assess the relevance of some of the documents in the common pool $\mathscr{P}$. 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 fixed number of documents with the highest priority scores. In the standard pooling technique, the documents are ranked based on the query relevance. Thus one can choose a uniform pool depth across queries to select documents to fit the available budget B_1. Therefore, the priority score w_d is 1 for documents above the cut-off rank and 0 for those below.

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 from Stage 1. The allocation of B_2 is based on a convex optimization of a cost function that seeks to (i) achieve maximum agreement with the evaluation of $\boldsymbol{S_L}$ systems using the full set of common documents $\mathscr{P}$ and ideal budget Ω, and (ii) generalize to new, unseen systems.

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

3.1 Concepts and Notation

For the population of all IR systems $\boldsymbol{S}$, we observe the retrieval performance of each participating system over a finite set of N test queries. The performance measurements are represented in the form of a performance matrix $\boldsymbol{X}$. 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 of a randomly selected system in $\boldsymbol{S}$ across all the queries. If $\boldsymbol{x}$ is the system's row in the matrix $\boldsymbol{X}$, then

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

where $\boldsymbol{e} = \{1\}^{N\times 1}$ is an N-dimensional vector of 1's. By definition, the column vector $\boldsymbol{m}$ comprises system average performance μ.

We are interested in the expectation and variance of μ across all the systems. We therefore define $\boldsymbol{\alpha} \in R^{1\times N}$ to be the vector of average performance scores for an individual query across the systems. Further, let $\boldsymbol{\Sigma}$ denote the $N \times N$ covariance matrix of the N query-systems vectors. Then the expectation and the variance of μ across systems are given by [11],

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

More generally, the performance μ of a system can be expressed as a weighted combination of the scores $x_{i,j}$. Let $\boldsymbol{\beta} \in [0,1]^{N\times 1}$ denote the associated weight vector with real values in [0,1]. Then the weighted average is expressed as $\mu_\beta = \boldsymbol{x\beta}$ and the expectation and the variance of μ_β across systems are given by

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

We now determine $\boldsymbol{\beta}$ as priority scores of queries in order to expand relevance judgments from Stage 2 under specified conditions.

3.2 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] & [12]). Similarly, some queries are more effective than others [13]. 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 [8], for simplicity, we adopt the uniform selection of documents across queries and focus our attention on the query prioritization.

We consider a query j as representative of the query set if its performance across systems is similar to the average performances of systems across all the test queries, i.e., the *j-th* column of $\boldsymbol{X}$ is close to the vector $\boldsymbol{m}$.

Our objective is to determine the most representative subset of queries based on several criteria. We formalize this by defining the vector $\boldsymbol{m_\beta}$ to represent the weighted average performance of

[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 systems across queries with query weights $\boldsymbol{\beta}$ and $\boldsymbol{m}_\beta[\mathrm{i}] = \mu_{\beta i}$, $\{\forall\, i : s_i \in S\}$, the weighted average performance of the system *i* across queries. We introduce an objective function $f(\boldsymbol{\beta})$ to define 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., $\mu_{\beta i} \approx \mu_i$.

The closeness of two rankings is usually measured using Kendall-τ. Unfortunately, using such a measure leads to computationally inefficient solutions. Consequently, we did not consider this similarity measure further. However, we do use Kendall-τ as an evaluation measure in our experiments to assess the quality of the optimization method.

3.2.1 Performance Score Similarity

There are many ways to characterize similarity in values between $\boldsymbol{m}_\beta$ and $\boldsymbol{m}$ such as the mean squared error and correlation. In our experiments we measure and report on correlation. Experiments with mean squared error are reported in [14].

The linear correlation measure ρ_β, between μ_β and μ is given by

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

with the covariance between μ and μ_β computed as

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

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. The maximum value of Equation (2) can be approximated by the minimization problem that is expressed in a quadratic programming form[2]:

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

where the $\eta \geq 0$ is a regularization parameter to adjust the trade-off between the quadratic part ($\boldsymbol{\beta^T \Sigma \beta}$) and the linear part ($\boldsymbol{e^T \Sigma \beta}$) part. In our experiments the regularization term η is set to N^{-1}. We now add constraints to arrive to the final optimization function.

3.3 Constraints

We consider two constraints. The first establishes the maximum available budget that can be used for additional judgments while the second, referred to as the *generalizability constraint*, ensures effective evaluation of new, previously unseen systems. We note that other constraints could easily be incorporated into this framework.

3.3.1 Budget Constraint

During Stage 2, we assume that a fixed budget B_2 is available for relevance judgments. It is natural to assume that the budget is allocated in proportion to each query's priority. We can, without loss of generality, take the query weight coefficients $\boldsymbol{\beta}$ 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. Given the finite budget, we have that $\sum_{j=1}^{N} \beta_j = \frac{B_2}{\Omega - B_1}$. The number of 'active' queries, i.e., queries for which β_j>0, is then based on the optimization:

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

3.3.2 Generalizability Constraint

If all the relevant documents for each query in the test collection are identified, then the test collection generalizes to any system. While we cannot guarantee that all the relevant documents are detected, it is clear that the *fewer unidentified relevant documents there are, 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 $\beta_j \times (\Omega - B_1)$ of the B_2 budget to a query q_j then, at the end of the Stage 2, the number of newly judged relevant documents will be proportional to $\beta_j r_j$. Thus, the total number of relevant documents judged in Stage 2 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

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

where $\sum_{j=1}^{N} r_j$ is added as a normalization factor to keep the first and second term in the same scale. The above optimization function is convex and we solve it using a sequential quadratic programming algorithm [16]. Section 4.2 discusses how to estimate the expected number of relevant documents $r_{j.}$.

4. IMPLEMENTATION DETAILS

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

4.1 Random Sampling of Systems

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

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

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

In practice, new systems may not be considered as drawn from a random sample. For example, over time, new systems are likely to perform better than participating systems, as system performance improves. In this case, we can use unbiased estimators for *weighted* systems to approximate $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$. Unfortunately, space

[2] The optimization form in Equation 3 is in *convex-fractional form* [15] and is optimized by transferring it to quadratic programming form.

limitations precludes further discussion and the interested reader is directed to [17].

4.2 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.

Given a set of initially judged documents, Carterette et al. [18] applied logistic regression to calculate the probability of relevance of un-judged documents. We use the same method to partition un-judged documents into relevant and non-relevant. 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:

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

where $\boldsymbol{w}$ is the parameter vector of the model and $\boldsymbol{F}$ is a feature vector that uses the document similarity features as introduced in [18]. An un-judged document d_i, retrieved for a query q_j is labelled as relevant if the probability of relevance $R(d_i, q_j) > 0.5$; otherwise d_i is labelled as non-relevant. Hence, the total number of relevant unseen documents for a query j is estimated by the number of un-judged documents with $R(d_i, q_j) > 0.5$.

5. EXPERIMENTAL EVALUATION

In this Section we describe a set of experiments that we conducted to evaluate the proposed *QP* method (Equation 6). Evaluations are conducted by comparing the ranking of the systems based on the full set of queries and available relevance judgments, with the ranking based on our method. The comparison uses Kendall-τ.

In the evaluation, we are particularly interested in the generalization to unseen systems. Thus we define the criteria for identifying markedly different systems. We use the *mean average reuse* (MAR) to characterize individual systems (see Section 5.3) and select those with low MAR as 'new', yet unseen systems.

Note, in order to avoid bias against any individual system, all participating systems contribute equally to the pooling of documents in both phases.

5.1 Benchmarks

We consider three baseline methods for resource allocation in comparison with our resource optimization method (*QP*):

(i) ***Uniform Allocation (UN)***, in which the 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 scores and use the average bias as a correction term for both the queries and the systems, as proposed by Webber and Park [1].

Note, the original algorithm by Webber and Park [1] assumes that relevance judgments are rendered for documents retrieved by new systems. In our context, the score adjustment is applied to the relevance judgments of documents within the common pool $\mathscr{P}$, i.e., contributed by the participating systems. We use the *SA* method in 1000 trials of random query sampling and report the average of the corresponding results.

5.2 Data Sets and Parameter Settings

Our experiments were performed using TREC 2004 Robust track. 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 submissions from the same site. In particular, when experiments require that we exclude some of the systems in order to treat them as 'new' systems, we hold out 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.

The TREC 2004 Robust track consists of 249 queries, 14 sites with a total of 110 automatic runs, and 311,410 relevance judgments obtained over documents in TREC Disks 4 & 5 corpora, excluding the Congressional Record sub-collection.

Comparative evaluation of TREC runs is conducted based on the *average precision* (AP). However, since we use only a fraction of relevance judgments in Stage 1, many documents remain un-judged. Consequently, the AP scores measured for participating systems are uncertain and the performance matrix ***X*** is noisy. For that reason, in our experiments we use *infAP* rather than AP to measure systems effectiveness with respect to the initial judgments. The *infAP* scores provide a better approximation of the true AP scores [19] and, hence, a less noisy performance matrix ***X***.

5.3 Experimental Setup

In order to test the generalization and robustness of the four resource allocation methods relative to the evaluation of new systems, we first divide the TREC runs into participating systems and held-out systems. During Stage 1, we randomly select a few sites and use their corresponding runs as participating IR systems. Using the pooling technique we select the set of documents retrieved by these participating systems and compile the corresponding relevance judgments. The pool depth is adjusted to fit the budget allocated to Stage 1.

For each held-out system, and each query, we compute the *average reuse* (AR) [18] to measure the overlap between the documents retrieved by a held-out system and the judged documents. We then define the *mean average reuse* (MAR) 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. This auxiliary group is evaluated with the relevance judgments obtained at Stage 1 and their performance values are added to the matrix ***X***. Note, however, that the auxiliary group does not contribute to the document pool. The second group, consisting of runs with low MAR values, forms the set of new systems.

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 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, constituting the auxiliary group, are evaluated with the existing (Stage 1) relevance judgments and their performance values are added to the matrix $\boldsymbol{X}$.
5. Prioritize queries using the *QP* method, i.e., 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.

5.3.1 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≤λ≤10. This is performed after Stage 1 but before Stage 2.

During 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 a query q_j is set to the number of relevant documents identified during Stage 1, using the 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≤λ≤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 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.

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 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							
			participating systems				new systems			
			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
2		(5,5)		0.61	0.7	0.78		0.52	0.66	0.71
3		(8,2)		0.63	0.67	0.79		0.52	0.63	0.74
4	(10, 40)	(4,16)	0.72	0.66	0.76	0.9	0.68	0.62	0.7	0.76
5		(10,10)		0.68	0.79	0.89		0.65	0.77	0.81
6		(16,4)		0.74	0.81	0.91		0.67	0.74	0.83
7	(20, 40)	(4,16)	0.79	0.69	0.83	0.91	0.8	0.66	0.74	0.84
8		(10,10)		0.75	0.82	0.89		0.67	0.8	0.9
9		(16,4)		0.77	0.83	0.91		0.7	0.81	0.91

5.4 Experimental Results

For the TREC 2004 Robust 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.

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. These experiments are repeated 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 over the full set of queries and corresponding relevance judgments. 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.

5.4.1 Experimental Results

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 the participating and the 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 the uniform allocation for new systems but remains better for participating systems. In contrast, our *QP* method yields superior results in all cases, except for configuration #1, in which the initial budget B_1 provides only 2000 relevance judgments, i.e., only 0.6% of the total 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; an indication that the query prioritization 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. [18] that a higher diversity of participating systems results in a better ranking of new systems.

6. DISCUSSION AND FUTURE DIRECTIONS

In this paper we consider the problem of prioritizing query-document pairs for relevance assessment given a budget constraint, in order to (*i*) improve the accuracy of evaluating participating systems, and (*ii*) ensure that the test collection generalizes to new, previously unseen systems. We propose a two-stage procedure. In Stage 1, we allocate a budget B_1 uniformly across all queries, acquiring a corresponding set of relevance judgments. In Stage 2, we sue information from Stage 1 to prioritize query-document pairs and allocate a budget B_2 accordingly. While we presented a 2-stage process, our method is iterative and can be applied repeatedly to support a growing set of systems and the corresponding set of relevance assessments.

The novelty of our work is in (*i*) modeling query and document selection through explicit cost optimization, and (*ii*) formulating the problem as a convex optimization for which computationally efficient algorithms exist for identifying the optimum solution. Our experiments compare the *QP* method with, uniform, random sampling and a variant of the score adjustment method presented in [1]. They provided strong evidence that the *QP* method is (*i*) superior to the selected benchmark methods, (*ii*) exhibits good accuracy, i.e., predicts the performance of participating systems, and (*iii*) exhibits good generalization, i.e., predicts the performance of new systems.

One of the main advantages of the *QP* method is its extensibility. We can leverage research on identifying query characteristics that make queries *better* suited for use in system evaluation and formulate new components and constraints within the optimization framework. Our future work will investigate a richer set of such heuristics, aiming to produce methods for test collection construction that are efficient, in terms of required resources for relevance assessments, and effective, in terms of accuracy of systems evaluations.

Furthermore, our experiment set up can be expanded to examine the sensitivity of estimation errors such as estimating the number of un-judged relevant documents and errors in the matrix $\boldsymbol{X}$. Finally, the full potential of the method would be realized through an effective iterative model of relevance assessment. Thus, it would be beneficial to evaluate the dynamic and real time application of the cost optimization in the context of the emerging practice of crowdsourcing relevance assessments.

REFERENCES

[1] W. Webber and L. A. F. Park, "Score Adjustment for Correction of Pooling Bias," in *Proc 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. Cormack, 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] 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.

[6] 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.

[7] 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.

[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. 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*.

[10] 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.

[11] P. J. Huber, *Robust Statistics*, 2nd ed. Wiley, 2009.

[12] 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.

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

[14] M. Hosseini, I. Cox, N. Milic-Frayling, V. Vinay, and T. Sweeting, "Selecting a Subset of Queires for Acquisition of further Relevance Judgements," in *3rd international Conference on the Theory of Information Retrieval*, 2011, pp. 113-124.

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

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

[17] M. Hosseini, I. J. Cox, T. Sweeting, N. Milic-Frayling, and V. Vinay, "Generalizing IR Test Collections through Incremental and Cost Constrained Gathering of Relevance Judgements," University College London RN/11/16, 2011.

[18] 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.

[19] 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.