An Uncertainty-aware Query Selection Model for Evaluation of IR Systems

Mehdi Hosseini, Ingemar J.Cox
Computer Science Department
University College London, UK
{m.hosseini, ingemar}@cs.ucl.ac.uk

Nataša Milić-Frayling, Milad Shokouhi, Emine Yilmaz
Microsoft Research Cambridge, UK
{natasamf,milads,eminey}@microsoft.com

ABSTRACT

We propose a mathematical framework for query selection as a mechanism for reducing the cost of constructing information retrieval test collections. In particular, our mathematical formulation explicitly models the uncertainty in the retrieval effectiveness metrics that is introduced by the absence of relevance judgments. Since the optimization problem is computationally intractable, we devise an adaptive query selection algorithm, referred to as Adaptive, that provides an approximate solution. Adaptive selects queries iteratively and assumes that no relevance judgments are available for the query under consideration. Once a query is selected, the associated relevance assessments are acquired and then used to aid the selection of subsequent queries. We demonstrate the effectiveness of the algorithm on two TREC test collections as well as a test collection of an online search engine with 1000 queries. Our experimental results show that the queries chosen by Adaptive produce reliable performance ranking of systems. The ranking is better correlated with the actual systems ranking than the rankings produced by queries that were selected using the considered baseline methods.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.4[Systems and Software: Performance Evaluation]

General Terms

Measurements, Algorithms, Theory and Performance

Keywords

Information Retrieval, Test Collection, Query Selection

1. INTRODUCTION

Modern test collections are large, comprising millions of documents and thousands of queries that require relevance judgments in order to calculate metrics of retrieval effectiveness. Constructing a test collection incurs a cost which, in the simplest case, is proportional to the number of queries chosen to evaluate the retrieval systems and the number of documents that need to be judged per query. Hence, the cost can be reduced by: (*i*) query selection that limits the number of queries for which relevance judgments need to be collected, (*ii*) document selection that reduces the number of documents to be judged for each query, and (*iii*) a combination of the two.

Much recent work has been devoted to effective and efficient construction of test collections [1, 5], with the primary focus on document selection [4]. In this paper, we explore methods for query selection as a means of dealing with a budget constraint. As Guiver et al. [10] showed, it is possible to reproduce the results of an exhaustive evaluation of systems by using a much reduced set of queries. However, it is still unclear how to select such a subset when relevance judgments are not available for the queries under consideration.

In our approach, we first formulate query selection as an optimization problem, minimizing the error in systems evaluation based on a subset of queries. In contrast to the previous work which is mostly retrospective and assumes that some relevance judgments are available for each query [10, 15, 17], our model is designed to work in practice and does not require relevance judgments for a query that is not yet selected. The mathematical formulation shows that an optimal subset of queries satisfies the following properties: (*i*) selected queries have a low correlation with one another, thereby maximizing the information we gain from each, (*ii*) selected queries have strong correlation with the remaining queries, as without this correlation there is no predictive capability, and (*iii*) the total uncertainty associated with the selected queries is small. Here the correlation between two queries refers to their similarity in evaluating systems.

Since selecting the optimal subset of queries is a computationally intractable problem, we approximate the solution by an iterative query selection process referred to as Adaptive. The Adaptive algorithm starts by selecting the first query at random, assuming no information about relevance judgments. However, once this query is selected, the relevance judgments are acquired and used to assist with the selection of subsequent queries.

Specifically, at each iteration we use previously selected queries and associated relevance judgments to train a classifier that estimates the relevance of documents pooled for each of the unselected queries. Using the output of the clas-

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

sifier we compute the probability of relevance for the pooled documents and used it to estimate the performance metric, e.g. the average precision, and the corresponding approximation variance which we refer to as *uncertainty*.

We evaluate our method by comparing the ranking of systems based on the subset of queries with the ranking over the full set of queries. We report the results in terms of Kendall-τ and Pearson correlation coefficients and show that the query subsets chosen by our models are significantly more effective than those selected by the considered baseline methods.

It is known that using query subsets may lead to poor performance when estimating the performance of previously unseen (new) systems [17]. We conduct experiments to investigate how our method generalizes to new systems. We show that the iterative algorithm can be modified to improve generalizability. Additionally, we consider the query selection problem for multiple metrics. In our experiment we show that a query subset selected based on a particular metric may not provide reliable systems evaluation when another metric is used to measure retrieval performance. Thus, we modify our query selection algorithm to select a query subset that leads to reliable evaluations across multiple metrics.

In summary, the contributions of this paper are threefold: (*i*) we provide a theoretical model for query selection that explicitly models uncertainty in retrieval effectiveness scores, (*ii*) we develop Adaptive, an iterative algorithm, that efficiently implements our theoretical model in practice, and (*iii*) we modify the iterative algorithm to investigate how the selected query subset generalizes to (a) new unseen systems and (b) changes to the evaluation metric.

2. RELATED WORK

The increased size of document corpora and query sets has made the cost of relevance assessments one of the main challenges in creating IR test collections. Spark-Jones and Van Rijsbergen [19] proposed a document pooling technique to select a subset of documents to be judged and the National Institute of Standard and Technology (NIST) adopted this method in most TREC experiments. For example, in the TREC AdHoc task, each participating system adds the top-100 ranked documents per query to the common pool to be judged by human assessors. Documents that are not included in this pool are assumed to be non-relevant. Several alternative approaches to the original pooling method have been suggested in order to judge more relevant documents at the same pool depth, e.g. Zobel [21] and Cormack et al. [7].

On average, NIST assessors judge 2,000 documents per query. This is often sufficient for a reliable evaluation of systems, even for recall-sensitive metrics such as average precision (AP), and increases the reusability of the TREC test collection for other tasks and new systems. Nevertheless, it still demands a considerable human effort and incurs a significant cost when the test collection contains a large number of queries. As a result, most of the existing TREC test collections contain as few as 50 queries.

Using small samples of queries to measure retrieval performance may cause considerable errors in the systems evaluation. Therefore, recent studies have concentrated on the IR evaluation with large set of queries [5]. In order to make relevance assessments feasible, various document selection methods are suggested [2, 4]. These methods reduce the number of documents that need to be judged per query and are often used in conjunction with evaluation metrics [2, 4, 20] designed for shallow sets of relevance judgments.

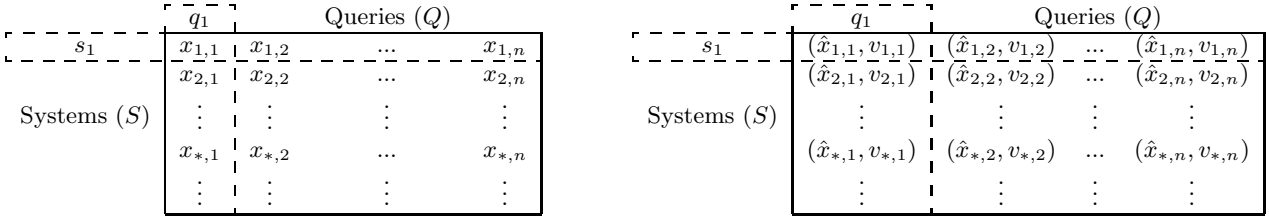
Following the trend of online search engines with large number of queries, NIST has recently launched the Million Query track [1]. In 2007, the Million Query track used 1800 queries for evaluation. The track mostly focused on (*i*) analyzing the efficiency of document selection methods, and (*ii*) the impact that incomplete relevance judgments have on the measurement accuracy.

Query selection is a complementary approach to document selection and is used to reduce the cost of creating test collections. Guiver et al. [10] have shown that query subsets vary in their accuracy of predicting the average performance of systems. Their results indicate that some subsets of queries, known as representative subsets, are particularly good predictors of the systems average performance as measured over the full set of queries. Mizzaro and Robertson [15] explored the characteristics of individual queries that were beneficial for systems evaluation. They defined the notion of hubness for queries where a higher hubness score indicates that a query is better than others at distinguishing the systems retrieval effectiveness. Robertson [17] later showed that the query selection based on the hubness scores alone does not necessarily result in a reliable prediction of the systems rankings. The work by Guiver et al. [10] shows that, indeed, representative query sets that are effective in approximating the systems ranking, comprise of queries that range in their individual ability to predict the systems performance.

Most studies on query selection are based on retrospective analysis [10, 15, 17] and assume that relevance judgments are available at the time the subset of queries is selected. Typical scenarios involve selecting queries to increase the pool of existing relevance judgments to accommodate evaluation of new systems. For example, Hosseini et al. [14] proposed a query selection approach to enhance the reusability of a test collection. Their focus was on generalizability, i.e. a reliable ranking of systems that did not contribute to the judged pool of documents. They assumed that an initial set of relevance judgments is available for each candidate query, and proposed a budget constrained model to select an optimal subset of queries. The selected queries were then used to expand relevance judgments based on the documents retrieved by new systems. The authors later extended their work [13] by considering a query-document pair selection problem when relevance judgments were to be created iteratively. Yet again, they assumed that the available budget allows for gathering at least a few relevance judgments per each query in the test collection. The authors applied convex optimization with a budget constraint to collect relevance judgments optimally across queries.

3. QUERY SELECTION

Previous research provided valuable insights on framing theoretical models for query selection [13, 14]. It focused on cases where there were judged documents for each query in the pool. The objective was to select a subset of queries to obtain additional relevance judgments for a more robust evaluation. In contrast, we focus on a scenario when no relevance judgments are available until a query is selected. We assume that a large set of queries has been initially compiled to measure the performance of a set of systems. Our goal



(a) In Theory (b) In Practice

Figure 1: (a) The true performance matrix X for systems in S and queries in Q. Each entry indicates the system performance score based on the available relevance judgments. (b) The approximated performance matrix $\hat{X}$, for systems in S and queries in Q. Each pair indicates the estimated performance and associated uncertainty.

is to find a subset of queries that most closely approximates the system evaluation results that would be obtained if we judged documents for the full set of queries.

To that effect we first formulate the query selection problem as in [13], assuming that relevance judgements are available and, thus the performance scores are known. We then relax this assumption and introduce our query selection model in the next section.

Consider a set of n known queries, Q, together with a set of l known systems, S. When relevance judgments are available, we can represent the associated retrieval effectiveness scores of the l systems against the n queries using a $l \times n$ performance matrix $X \in R^{l \times n}$ (Figure 1a). In X, the rows represent systems, the columns represent queries, and the value $x_{s,q}$ shows the performance score, e.g. the average precision, of s^{th} system in S against q^{th} query in Q. We also consider a column vector $M \in R^{l \times 1}$, as the average performance vector. The elements of M represent the average performance scores of individual systems across the set of queries. Thus, when the metric is AP, M represents the mean average precision (MAP) scores for individual systems.

Now, let $\Phi = \{j_1, ..., j_m\}$ be a subset of $\{1, 2, ..., n\}$ with $1 \leq m \leq n$ and Q_Φ be the corresponding query subset. We define $M_\Phi \in R^{l \times 1}$ as the column vector comprising the average performance of systems for the subset of queries, Q_Φ. The aim of a query selection method is to find a subset of m queries such that the corresponding column vector M_Φ closely approximates the M vector.

The approximation is quantified using the mean squared errors between elements of M and M_Φ, if the similarity in terms of the absolute values of performance scores is of interest. Alternatively, we can use Kendall-τ correlation, if the similarity in the ranking of systems is of interest, or Pearson linear correlation, if the similarities in the relative performance scores are of interest. As in [13], we choose Pearson correlation as it is amenable to mathematical optimization. However, we use Kendall-τ as our final evaluation measure for comparing the rankings of systems produced by full set and a subset of queries.

The Pearson correlation between the elements of M and M_Φ is

$$\rho_\Phi = \frac{cov(M, M_\Phi)}{\{var(M) var(M_\Phi)\}^{\frac{1}{2}}} \quad (1)$$

where

$$\begin{aligned} var(M) &= n^{-2} e^T \Sigma e \\ var(M_\Phi) &= m^{-2} d^T \Sigma d \\ cov(M, M_\Phi) &= n^{-1} m^{-1} d^T \Sigma e \end{aligned}$$

where $e = \{1\}^{n \times 1}$ is the vector of n components, each equal to 1; $d \in \{0, 1\}^{n \times 1}$ is a binary vector such that $d_j = 1$ if $j \in \Phi$, and $d_j = 0$ otherwise. Note that this is slightly different from the formulation in [13] as the task is also different. In [13], d is defined as a vector of real values and its role is to establish the proportions of the budget allocated across queries. In this paper, since we are interested in the query selection problem, the vector d is defined, without loss of generality, as a binary vector and considered as an indicator to selected queries.

Following Equation 1, we have $\Sigma = cov(X)$ as the $n \times n$ covariance matrix of the system-query performance scores such that $(i, j)^{th}$ element of Σ is the covariance between i^{th} and j^{th} columns of X. In practice, the covariance matrix Σ calculated based on the l systems is:

$$\Sigma = (l-1)^{-1} \sum_{i=1}^{l} (x_i - \alpha)^T (x_i - \alpha)$$

where $\alpha = l^{-1} \sum_{i=1}^{l} x_i$ and x_i is the i^{th} row of X. The optimum subset maximizes the Pearson correlation ρ_Φ, where substituting for the variances and the covariance we have

$$\rho_\Phi = \frac{(d^T \Sigma e)}{\{(e^T \Sigma e)(d^T \Sigma d)\}^{\frac{1}{2}}}$$

This derivation assumes that the elements of the X matrix are true representatives of the systems performance as computed over the full set of relevance judgments. Of course, in practice this assumption does not hold because of the absence of relevance judgments during query selection. In the following section, we propose an extended model that uses performance predictors for approximating systems true performance. We then extend the model to incorporate explicitly the noise in the measurement of the systems performance.

4. UNCERTAINTY-AWARE SELECTION

We assume that instead of containing the true performance values, each element of X holds a predicted performance estimate with a variance from the true value that we refer to as *uncertainty*. Hence, the noisy $\hat{X}$ matrix can be represented as shown in Figure 1b where each of its ele-

ments represents a pair of values: $\hat{x}_{s,q}$ and $v_{s,q} = var(x_{s,q})$. In addition, let $\hat{M}_\Phi \in R^{l\times 1}$ be the vector of l average performance scores computed based on the query subset, Q_Φ, and the performance matrix $\hat{X}$. Thus, in practice we look for a subset that maximizes the Pearson correlation between $\hat{M}_\Phi$ and M. To compute the Pearson correlation we need to compute the variances and the covariance of $\hat{M}_\Phi$ and M.

The variance of $\hat{M}_\Phi$ is due to two sources, the variance across systems and the variance due to the measurement noise. The first variance is expressed by $var(M_\Phi)$ as calculated in Section 3. To compute the second variance we first note that each of the elements in $\hat{M}_\Phi$ has its own variance. If $\hat{\mu}^i_\Phi$ denotes the performance of i^{th} system in $\hat{M}_\Phi$, then the associated variance is

$$var(\hat{\mu}^i_\Phi) = m^{-2}\sum_{j\in\Phi} v_{i,j}$$

Following the law of total variance [3], the variance of $\hat{M}_\Phi$ is given by

$$var(\hat{M}_\Phi) = var(M_\Phi) + E_s(var(\hat{\mu}^s_\Phi)) = \quad (2)$$
$$m^{-2}d^T\Sigma d + m^{-2}\sum_{j\in\Phi} E(v_j) = m^{-2}d^T(\Sigma + U)d$$

where $1 \le s \le l$ and $U = diag\big(E(v_1), ..., E(v_n)\big)$ is a diagonal matrix, referred to as the uncertainty matrix. $E(v_q) = l^{-1}\sum_{i=1}^{l} v_{i,q}$ is the mean uncertainty for a query q.

To compute the covariance between $\hat{M}_\Phi$ and M, let us consider an unknown system that is randomly sampled from the l systems, and let x and $\hat{x}$ denote the associated performance row vectors in X and $\hat{X}$. The system's average performance computed based on X and the full set of queries is

$$\mu = n^{-1}xe$$

Also the systems average performance based on the subset of m queries, Q_Φ, and $\hat{X}$ is

$$\hat{\mu}_\Phi = m^{-1}\hat{x}d$$

where e and d are the column vectors as defined in Section 3. The covariance between $\hat{M}_\Phi$ and M is then

$$cov(\hat{M}_\Phi, M) \equiv cov(\hat{\mu}_\Phi, \mu) = m^{-1}n^{-1}cov(\hat{x}d, xe) =$$
$$m^{-1}n^{-1}d^T cov(\hat{x}^T, x)e = m^{-1}n^{-1}d^T\Sigma e \quad (3)$$

where $\hat{x}d = d^T\hat{x}^T$, and

$$cov(\hat{x}^T, x) = cov(x^T + \epsilon, x) =$$
$$E\big\{\big(x - E(x)\big)^T\big(x - E(x)\big)\big\} \equiv cov(X) = \Sigma$$

Note that, $\hat{x}^T = x^T + \epsilon$ where $\epsilon \in R^{1\times n}$ is the vector of estimator's noise.

Thus, the Pearson correlation between $\hat{M}_\Phi$ and M is given by

$$\hat{\rho}_\Phi = \frac{(d^T\Sigma e)}{\big\{\big(e^T\Sigma e\big)\big(d^T(\Sigma + U)d\big)\big\}^{\frac{1}{2}}} \quad (4)$$

Formally, we seek a subset Q_Φ that maximizes $\hat{\rho}_\Phi$. Reordering the correlation above we have

$$\gamma_\Phi \equiv (e^T\Sigma e)^{\frac{1}{2}}\hat{\rho}_\Phi = \frac{(e^T\Sigma d)}{\big(d^T(\Sigma + U)d\big)^{\frac{1}{2}}}$$

Selecting a subset of queries Φ that maximizes $\hat{\rho}_\Phi$ is equivalent to selecting a subset of queries that maximizes γ_Φ since $(e^T\Sigma e)^{\frac{1}{2}}$ is a constant. Let $\sigma_{i,j}$ be the $(i,j)^{th}$ element of Σ and $E(v_j)$ be the j^{th} diagonal element of the uncertainty matrix U. We can then rewrite γ_Φ as

$$\max_\Phi \gamma_\Phi = \frac{\sum_{1\le i\le n, j\in\Phi}(\sigma_{i,j})}{\big\{\sum_{i,j\in\Phi}(\sigma_{i,j}) + \sum_{j\in\Phi} E(v_j)\big\}^{\frac{1}{2}}} \quad (5)$$

Equation 5 provides valuable insights into the query selection problem. In order to maximize γ_Φ we seek a set of queries that minimizes the denominator and maximizes the numerator.

To minimize the denominator, we need to choose m queries for which the corresponding columns in X are least correlated with one another. This is equivalent to maximizing the information we derive from each query in the subset. Conversely, if the columns of X are perfectly correlated, then all the queries provide the same information and we may as well have a subset of size one. Additionally, the sum of the expected uncertainty, $\{E(v_j)|j \in \Phi\}$, of the selected queries should be a minimum.

The numerator is maximized if the columns in X, associated with the selected queries, have high correlation with the rest of columns in X. This is also intuitively clear. After all, if the selected subset is completely uncorrelated with the remaining queries, then it has no prediction power of the systems performance on the remaining queries. Thus, we seek a subset of queries that are highly correlated with the rest of queries.

5. ADAPTIVE QUERY SELECTION

Thus far, we introduced a theoretical model for query selection that extends the previous work by explicitly modeling uncertainty. The model allows for the elements of the performance matrix to represent predicted rather than actual performance values. Equation 5 shows how the predicted performance values can be incorporated into the optimization process, but does not indicate how they can be computed in practice. We describe in detail the Adaptive query selection algorithm that iteratively selects queries and refines the estimations in $\hat{X}$. This method exploits the supervised prediction and uses the relevance judgments of already selected queries, to train a model for selecting subsequent queries.

The Adaptive method iteratively selects a new query, that in combination with the previously selected queries maximizes Equation 5. The relevance judgments of the previously selected queries are used to predict the relevance judgments of yet non-selected queries. The adaptive method subsequently estimates the performance scores, and updates the $\hat{X}$ matrix by adding the systems performance scores for the selected query, and those predicted for the non-selected queries. This process is repeated until we reach the maximum number of queries to be selected.

In order to predict the relevance of documents for queries that have not been selected yet, we train a classifier using judged documents of previously selected queries as training data. Each query-document pair is represented to the classifier as a vector of $7+l$ generic features where l refers to the number of systems. These features are:

- The number of systems that retrieved the query-document pair (one feature).

- The average, minimum and maximum ranks given to the query-document pair by the systems (three features).
- For systems that retrieve the query-document pair, we calculate their corresponding past-performance scores based on the subset of queries for which we have relevance judgments. For example, if the metric is AP, we compute a system's MAP based on its AP scores obtained for previously selected queries. We then determine the minimum, maximum and average across systems (three features).
- The l relevance scores provided by l systems for the given query-document pair (l features). If a system does not retrieve the document, the corresponding score is set to the minimum score obtained for the documents retrieved by that system.

We train a linear support vector machine (SVM) as our classifier [8]. For each query-document pair we map the output of the classifier to a probability score using the calibration method proposed in [16]. Briefly, let $f \in [a, b]$ be the output of the SVM classifier. We use a sigmoid function to map f to a posterior probability on [0,1]:

$$p_i = P(r_i = 1|f_i) = \frac{1}{1 + exp(Af_i + B)}$$

where r_i refers to the true relevance value of document i, p_i is its probability of relevance, and A and B are the parameters of the sigmoid function, fitted using maximum likelihood estimation from a calibration set (r_i, f_i). The training data is the same as the training data used by the classifier. Thus, at each iteration we retrain the classifier and fit the sigmoid function to exploit the increase in training data from the new round of relevance judgments.

After each query-document pair is assigned a probability of relevance, we use these probabilities in the family of MTC estimators, proposed by Carterette et al. [4], to provide new estimates for the unknown performance scores in the $\hat{X}$ matrix. For example, when the metric of interest is $P@k$, the expectation and variance are calculated as:

$$E[P@k] = \frac{1}{k}\sum_{i=1}^{k} p_i$$
$$var[P@k] = \frac{1}{k^2}\sum_{i=1}^{k} p_i(1 - p_i)$$

where p_i is the calibrated relevance probability of the document retrieved at rank i. The formulations for other metrics, e.g. AP, can be found in [4].

6. EVALUATION SETTINGS

Query selection and document selection methods are often evaluated by considering the systems ranking they produce compared to the ground-truth systems ranking based on all the queries and the full set of relevance judgments. As in previous work [4, 10], we use Kendall-τ and Pearson coefficient as the correlation metrics. Kendall-τ penalizes disordering of high-performance and low-performance system pairs equally. However, in practice, distinguishing between best performing systems is often more important. Therefore, in many of our experiments, we also report separately the results for the subsets of best performing systems in terms of the average precision (AP) and the precision at position 100 ($P@100$).

We run our experiments on both the TREC 2004 Robust track, and the TREC-8 AdHoc track data sets. The TREC 2004 Robust track includes a collection of 249 queries, 14 participants with a total of 110 automatic runs, and 311,410 relevance judgments. The TREC-8 consists of 50 queries, 39 participants with 13 manual runs and 116 automatic runs, and 86,830 relevance judgments. In our experiments, we consider individual runs as different search systems, taking special care when considering runs from the same participant. We also create a Web test collection based on the query logs of a commercial search engine. This dataset comprises 1,000 queries, 50 runs of a learning to rank system trained with different feature sets, and 30,000 relevance judgments. We compare the performance of our query selection against query sets obtained by three baseline methods: random, oracle and iterative query prioritization.

Random: randomly selects a subset of queries. We report the results averaged over 1,000 random trials and consider 95% confidence interval of the sample average.

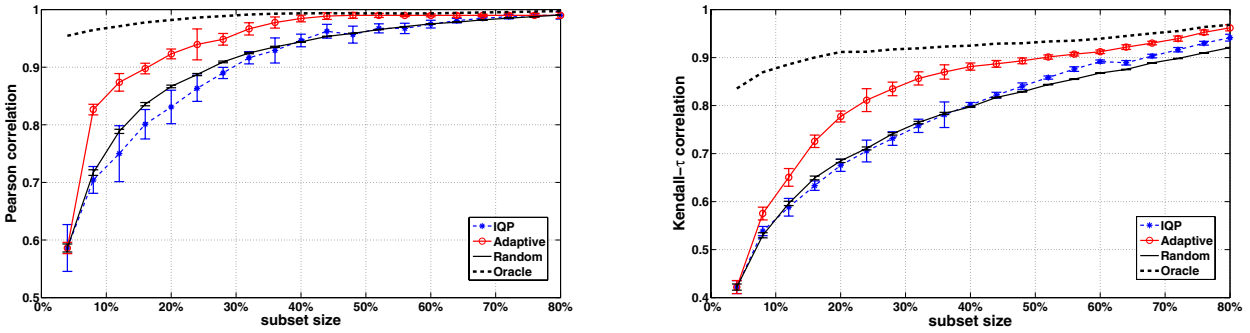
Oracle(Ideal): selects the best subset of queries by considering the full X matrix constructed from the full set of queries and all the relevance judgments in the test collection. For a given subset size $m < 10$ and $m > (n - 10)$, we perform an exhaustive search to find the best subset. Exhaustive search is computationally expensive for $10 < m < (n - 10)$. Therefore, we estimate the best subset of size $10 < m < (n - 10)$ by randomly generating 10,000 query subsets from which the best subset is selected.[1]

Iterative Query Prioritization (IQP): we consider a modified version of the budget constrained optimization method proposed in [13] as a query selection baseline. The original method, referred to as query prioritization (QP), cannot be used in our experiments because it is defined as a convex optimization that demands a set of initial judgments for all the queries. This assumption is not valid in our experiments. Thus, we consider a modified version of this method that does not rely upon such an assumption and is specialized for the query selection task. We replace the budget vector $\beta \in [0, 1]^{n\times 1}$ with a binary vector $d \in \{0, 1\}^{n\times 1}$ as an indicator to selected queries. Therefore, the optimization function is changed to

$$\max_d f(d) = \frac{d^T \Sigma e}{\left(d^T \Sigma d\right)^{\frac{1}{2}}} \quad \text{subject to } \sum_{j=1}^{n} d_j \leq m$$

This modified version, (IQP), starts from zero relevance judgments and iteratively selects queries and updates the vector d. Similar to the original method in [13] IQP does not consider the uncertainty in estimating the entries of the X matrix. That is, it uses the same classifier (as in our iterative adaptive method) but directly maps the output of the classifier to 0 or 1, when the relevance judgments are binary, and regards them as the predicted absolute relevance values. Therefore there is no calibration of relevance probabilities involved. As such, it does not use the MTC estimators discussed in Section 5 but, instead, relies upon the standard metrics, e.g. AP, to measure the systems performance. Considering IQP in our experiments helps us investigate the effect of incorporating measurement uncertainty into the query selection.

[1]We considered the oracle subset in our experiments to provide an upper bound for the performance of the query selection algorithms.



(a) (b)

Figure 2: Selecting queries using (i) Oracle, (ii) random, (iii) IQP, (iv) Adaptive query selection algorithm, for the Robust 2004 test collections with 249 queries. The first query is randomly selected. The results are averaged over 50 trials with AP metric.

7. EXPERIMENTAL RESULTS

In the experiments with TREC test collections, we consider all the official retrieval runs. Each system contributes 100 documents to the pool for each query. After selecting a query, the official TREC judgments are collected and revealed. The Adaptive and IQP methods, then add these recently judged documents to their training sets.

On each testbed, we report the results for three different groupings of systems: (i) all systems, (ii) 30 best performing systems, and (iii) only pairs of systems with a statistically significant performance difference, measured by the paired t-test with the significance level of 0.05.

Figure 2 shows the results on the Robust 2004 test collection with 249 queries. The retrieval evaluation metric is AP. Pearson and Kendall-τ correlation are used to measure the correlation of a query subset vector $\hat{M}_\Phi$, and corresponding vector M, calculated using the full set of 249 queries. At the initialization step of the Adaptive and IQP methods, the first query is randomly selected. To deal with the variation of random sampling, we consider 50 trials. We report the average of 50 trials as the average results for Adaptive and IQP. We also consider the 95% confidence interval of the average performance to detect significant differences between the query selection methods. For instance, in Figure 2a when the subset covers 28% of the full query set, the average of 50 Pearson correlation scores obtained by the Adaptive method is 0.94 and the associated standard deviation is 0.07. Thus the 95% confidence interval is: $\left[0.94 \pm 1.96 \times \frac{0.07}{\sqrt{50}}\right]$. The confidence intervals are shown as error bars. In general, the difference between two methods is statistically significant for a particular subset size, if the associated error bars do not overlap. As seen, in Figure 2a and 2b, for both Pearson correlation and Kendall-τ, the Adaptive method significantly outperforms the Random and IQP baselines across different subset sizes. From Figure 2b we see that the Adaptive method achieves a Kendall-τ correlation of 0.9 with a subset that covers 50% of the queries (125 out of 249 queries). However, the Random and IQP methods require at least 70% of queries to achieve the same Kendall-τ.

Table 1 summarizes the Kendall-τ and Pearson correlation for the four query selection methods when selecting {20, 40, 60}% of queries in the Robust 2004 and the TREC-8 test collections.

The columns labeled 'all' indicates the results for all the systems in a test collection. For both test collections and the three subset sizes, {20, 40, 60}%, the Adaptive method significantly outperforms IQP and Random in most cases. For instance, in the Robust 2004 test collection the Adaptive method obtains {15, 10, 5}% average improvements over Random and IPQ in Kendall-τ for subsets of {20, 40, 60}% respectively. Similar improvements are observed for the TREC-8 test collection.

The columns labeled 'top' indicates the results for the 30 best performing systems, i.e. those with the highest MAP scores. When calculating Pearson and Kendall-τ correlations, the vectors $\hat{M}_\Phi$ and M are constructed by considering only the top 30 systems. The remaining systems only contribute to the query selection process and are not used for evaluation. Once again, the Adaptive method significantly outperforms the IQP and Random methods in most of the cases. Interestingly, the improvements are even larger than the improvements of the full set of systems. For instance, for the Robust test collection, improvement in Kendall-τ is on average 10% for the full set of systems and it rises to 25% for the top 30 best performing systems. Similarly, the average improvement in Pearson correlation rises from 7% to 14% on average. Similar results are observed for the TREC-8 test collection.

The columns labeled 'sig' indicates the results for the pairs of systems whose performances difference is statistically significant. If a difference in average performance scores of two systems is not statistically significant, it is reasonable that they may be ordered differently when evaluated over a subset of queries. Such tied systems increase the probability of a swap and thus may considerably decrease Kendall-τ since the common formulation of Kendall-τ does not distinguish between pairs of systems with and without significant differences. This is, in fact, the case for the Robust and the TREC-8 test collection where about 30% of pairs of systems are tied as measured by the paired t-test with the significance level of 0.05. Thus, we also compute the Kendall-τ of systems with a significant difference in MAP. Again, the

Table 1: Comparisons of the four query selection methods for the two TREC test collections based on the AP metric. The statistically significant improvements of *Adaptive* over IQP and Random are marked by †.

Subset	Method	Robust2004					TREC-8				
		Kendall-τ			Pearson		Kendall-τ			Pearson	
		all	top	sig	all	top	all	top	sig	all	top
20%	**Random**	0.68	0.45	0.75	0.83	0.68	0.72	0.45	0.88	0.92	0.77
	IQP	0.67	0.47	0.78	0.86	0.70	0.74	0.53	0.92	0.93	0.81
	Adaptive	**0.77**†	**0.63**†	**0.85**†	**0.92**†	**0.79**†	**0.83**†	**0.69**†	**0.95**†	**0.95**†	**0.92**†
	Oracle	0.90	0.81	0.90	0.97	0.95	0.88	0.80	0.97	0.97	0.95
40%	**Random**	0.80	0.58	0.82	0.93	0.76	0.77	0.58	0.95	0.95	0.86
	IQP	0.80	0.56	0.85	0.94	0.78	0.81	0.66	0.96	0.95	0.89
	Adaptive	**0.87**†	**0.69**†	**0.89**†	**0.98**†	**0.89**†	**0.90**†	**0.81**†	**0.99**†	**0.97**†	**0.95**†
	Oracle	0.92	0.86	0.95	0.99	0.96	0.93	0.85	1.0	0.98	0.97
60%	**Random**	0.85	0.71	0.88	0.97	0.90	0.87	0.70	0.97	0.97	0.90
	IQP	0.88	0.73	0.90	0.96	0.91	0.88	0.80	0.99	0.98	0.92
	Adaptive	**0.91**†	**0.83**†	**0.95**†	**0.99**†	**0.96**†	**0.93**†	**0.85**†	1.0	0.98	**0.96**†
	Oracle	0.94	0.92	0.97	0.99	0.99	0.95	0.91	1.0	0.99	0.99

Table 2: Comparisons of Random and Adaptive using the Web test collection.

	Method	desired Kendall-τ		
		0.7	**0.8**	**0.9**
#queries	**Random**	167	368	739
	Adaptive	71	207	486
#relevance judgments	**Random**	5010	10235	28804
	Adaptive	2086	5803	15854

Adaptive method significantly outperforms IQP and Random in most cases.

We repeated the experiments in Table 1 for P@100 metric, and observed similar results for both the test collections.[2]

7.1 Results of the Web Data

We also investigate the performance of Adaptive on a test collection comprising the Web search results from a commercial search engine with 1,000 queries and 50 different search systems. Here, the systems are various rankers (runs) of a learning to rank system that were trained with different feature sets. To generate a ranker we randomly sample $g = \{5, 10, 20, 30 \text{ or } 40\}$ features from a given feature set and optimize the ranker on a common training set (the training details are out of the scope of this paper). For each query, the top 5 web pages returned by the rankers are pooled for relevance assessment. The performance of each ranker is measured according to the precision at the position rank 5 (P@5).

Table 2 reports the number of queries, and the number of relevance judgments required to reach Kendall-τ values of {0.7, 0.8, and 0.9} by the Adaptive, and the Random query selection method. We focus on the comparisons between Adaptive and Random because the random sampling is commonly used for selecting candidate queries. Thus, we can observe the cost reduction that can be achieved in practice by the Adaptive method.

The reported results for the Adaptive method are the average over 10 trials. In each trial, the initialization involves 20 queries instead of a randomly selected single query. This ensures a sufficiently large training set for the initial classifier without losing much in the query selection performance.

[2]The results on P@100 are not reported due to lack of space.

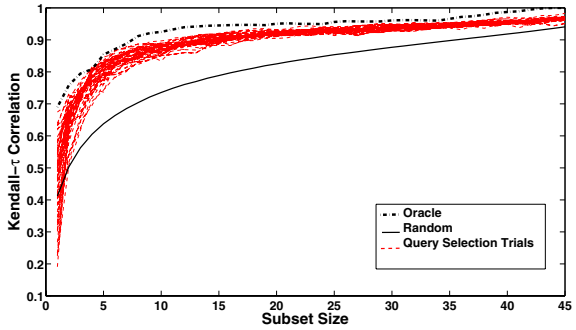


Figure 3: Sensitivity of the query selection to the first query using TREC-8 comprising 50 queries. The subset size varies between 1 and 45.

As seen in Table 2, the required subset sizes for τ={0.7, 0.8, 0.9} are statistically significantly smaller than those required for random sampling. For instance, the random method obtains $\tau = 0.9$ for a subset of size 739 whereas the Adaptive method only requires 486 queries to reach the same τ. This is equivalent to judging 12,950 fewer documents than the Random method. Similar results are observed for τ={0.7 and 0.8}.

7.2 Effects of Initialization

The initialization step of the Adaptive method involves randomly selecting the first query. We now consider the sensitivity of the Adaptive method to the selection of the first query. The choice of the first query could possibly affect the quality of both (i) the queries selected in the subsequent stages and (ii) the training data for the classifier. Our analysis focuses on (i) since the impact on the training data cannot be separated from the characteristics of the classification method and thus is out of the scope of this paper.

In order to isolate the effects of the initial query on the subsequent queries, we assume that the true X matrix is available, i.e. that the estimator has access to all relevance judgments. Using the TREC-8 data set, we randomly select the first query. Subsequent queries are iteratively selected based on the query selection model in Equation 5 but using

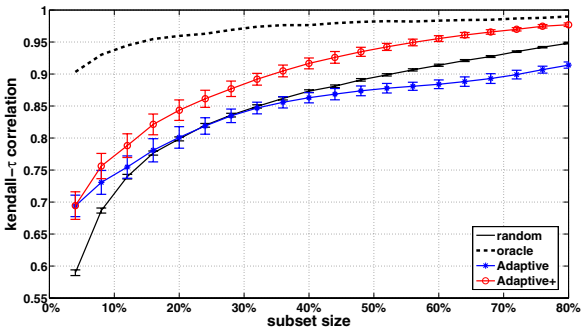


Figure 4: The generalizability test for the query subsets selected by Adaptive and Adaptive$^+$, the modified query selection method.

the true X matrix where the corresponding uncertainty U matrix is zero. Results are shown in Figure 3 for 50 trials. Each trail contains a distinct query for the initialization. As seen, the Kendall-τ scores obtained in all the trials converge to the Kendall-τ for the ideal query set more quickly than the Kendall-τ obtained for Random. The kendall-τ variation across trials decreases as more queries are selected. For the query subsets of size greater than 10 queries the performance is similar across all the trials. This suggests that the query selection model is robust to the selection of the first query.

8. GENERALIZATION

We now consider the generalization of the Adaptive query selection method. In Section 8.1 we discuss the effectiveness of the resulting query set in evaluating new systems that did not contribute to the query selection process. In Section 8.2 we consider the reliability of the query subset for evaluating systems performance using multiple metrics. This is particularly important when systems are compared using various performance metrics.

8.1 Evaluation of New Systems

Previous work [14, 17] has shown that queries selected based on the performance of a particular set of systems may not be effective in evaluating new, previously unseen systems. We observed a similar tendency by the Adaptive query selection. Thus, to avoid over-fitting to the systems used to select the queries we modify the Adaptive algorithm.

The modified algorithm is referred to as 'Adaptive$^+$' and comprises the following changes. When selecting a query we consider $c(c > 1)$ random subsets of the l systems of size $h(h < l)$. We allow overlaps between the subsets and ensure that each system appears in at least one of the subsets. For each subset of systems we choose a query that, in combination with already selected queries in Φ, maximizes γ_Φ (see Equation 5). Finally, we pick the query that is selected by most of the systems subsets, and consider it for relevance assessments.

We test the generalization of the Adaptive$^+$ approach using the two TREC test collections. We first randomly select 50% of systems and treat them as new systems. When selecting new systems, we hold out not only individual runs but the entire set of runs from the same participant. The remaining systems are considered as participating systems and used to select queries. When computing the performance metrics for the participating systems, we also remove documents that are uniquely retrieved by the new (held-out) systems.

Table 3: Comparing the generalization of a selected subset using two metrics: P@100 and AP.

Subset	Method	kendall-τ			
		Robust 2004		TREC-8	
		P@100	AP	P@100	AP
20%	Random	0.77	0.80	0.75	0.78
	Adaptive	0.76	0.82	0.76	0.80
	Adaptive$^+$	**0.84**†	**0.87**†	**0.81**†	**0.85**†
	Oracle	0.91	0.95	0.86	0.89
40%	Random	0.82	0.87	0.84	0.85
	Adaptive	0.80	0.85	0.84	0.86
	Adaptive$^+$	**0.89**†	**0.92**†	**0.90**†	**0.90**†
	Oracle	0.93	0.97	0.94	0.93
60%	Random	0.89	0.91	0.89	0.90
	Adaptive	0.84	0.88	0.87	0.91
	Adaptive$^+$	**0.93**†	**0.96**†	**0.95**†	**0.95**†
	Oracle	0.96	0.98	0.97	0.97

The results of the generalization test for the Robust 2004 test collection and Kendall-τ are shown in Figure 4. We created $c = 100$ random system subsets, each of size $h = 0.2 \times l$, where l refers to the number of participating systems. Figure 4 clearly shows that the Adaptive algorithm overfits the set of queries to the participating systems and performs no better than Random when evaluating new systems. In contrast, Adaptive$^+$ significantly outperforms Random across different sizes of the query subsets.

The detailed results of the generalization experiments are shown in Table 3 for both Robust and TREC-8 test collections, and two metrics, AP and P@100. In all cases the Kendall-τ obtained by Adaptive$^+$ is significantly larger than the Kendall-τ of the Adaptive and Random algorithm.[3]

8.2 Use of Alternative Performance Metrics

One of the goals of IR test collections is to enable evaluation of systems in terms of various metrics. Ideally, the subset of queries used for systems evaluation should provide reliable estimates of the systems performance regardless of the metrics used. In the following experiments we show that for some methods that may not be the case. Namely, when the metrics used to select queries differs from the metrics used to evaluate systems, the query subset may not provide reliable estimates of the systems performance.

For that reason, we modified the Adaptive algorithm to generalize across multiple metrics. The modified version is referred to as 'Adaptive*'. At each step of the query selection process, for each metrics, and for each non-selected query the Adaptive* computes the associated γ_Φ scores. It averages the scores across different metrics and then selects the candidate query with the maximum average of γ_Φ scores.

We consider four IR metrics: P@10, P@100, *Recall* and AP and measure the associated Kendall-τ scores (T_1) for query subsets of various sizes selected. Let T_2 be the set of Kendall-τ scores for various subset sizes calculated when the evaluation metric is different from the metric used for query selection – the selection metric. Ideally the Kendall-τ

[3]Similar results were also observed for Pearson correlation but not reported due to lack of space.

Table 4: The average Kendall-τ loss $\big(mean(T_2)$ - $mean(T_1)\big)$ for four metrics using the TREC 2004 Robust track. Given a metric α, T_1 denotes the set of Kendall-τ scores for various query subset sizes when the metric α is used for both query selection and system evaluation. T_2 denotes the set of Kendall-τ scores when the metric α is used to measure systems performance for a subset of queries selected using another metric.

Selection Metric	Evaluation Metric			
	P@10	**P@100**	**Recall**	**AP**
P@10	0.0	-0.082	-0.065	-0.068
P@100	-0.084	0.0	-0.042	-0.051
Recall	-0.076	-0.063	0.0	-0.073
AP	-0.089	-0.070	-0.062	0.0
Adaptive*	**-0.011**†	**-0.012**†	**-0.018**†	**-0.014**†
Random	-0.114	-0.086	-0.056	-0.078

scores in T_2 would comparable with those in T_1. To measure the distances between T_1 and T_2 set of scores we use $\big(mean(T_2) - mean(T_1)\big)$, i.e. the average loss Kendall-τ.

Table 4 represents the results of our experiment for the Robust 2004 test collection. Each of the four metrics are used as a selection metric for the Adaptive method and as a system evaluation metric. The Recall metric leads to the smallest average loss for P@10 and P@100 but not for AP. The best selection metric for AP and Recall is P@100. Thus, there is no single selection metric that leads to the minimum loss in Kendall-τ for all of the evaluation metrics.

We also select queries by using the Adaptive* method. As seen from Table 4, the average loss for all the metrics is considerably reduced. The last row of Table 4 represents the results of random sampling averaged over 1000 trials. In order to investigate whether the Adaptive* method significantly outperforms other methods, we measured the differences in average Kendall-τ loss using the paired t-test at the significance level of 0.05. Table 4 shows that Adaptive* leads to average Kendall-τ losses that are significantly smaller than for the Random method and any of the selection metrics.

9. DISCUSSION

Our experiments demonstrated the advantages of the Adaptive query selection method and its variations. Here we consider how they may be practically used to support the query selection and assessment scenarios encountered in the TREC type experiments. In TREC tasks, a set of queries is typically selected in advance and delivered to assessors to collect relevance judgments. This setting ensures that all the assessors are efficiently involved in the construction of relevance judgments, and the full set of judgments is constructed in a reasonable time.

We first investigated if we can use Adaptive to collect a subset of queries before constructing any relevance judgments. We considered the use of the query performance predictors, e.g. [9, 18], to construct the performance matrix $\hat{X}$ and iteratively select a subset of queries in the absence of any relevance judgments. Since no relevance judgments are collected, the elements of $\hat{X}$ are fixed throughout the iterations. Once a set of queries are selected, the associated relevance judgments are collected concurrently.

We used the pseudo-relevance judgments approach proposed by Soboroff et al. [18] to construct $\hat{X}$ since (i) it directly estimates the performance metrics and provides the corresponding variance that is required for our model, and (ii) it is reported to be among the best available performance predictors [11]. However, experiments with the Robust 2004 test collection showed that the accuracy of the performance prediction was not sufficiently high and, as a result, Adaptive failed to perform statistically better than the random sampling of queries.

We then took a different approach and modified Adaptive query selection to construct relevance judgments for multiple queries at a time. Instead of updating $\hat{X}$ at each iteration, we updated $\hat{X}$ after selecting a sequence of $k \geq 1$ queries. Once k queries were selected, the associated relevance judgments were constructed and $\hat{X}$ matrix was updated.

We evaluated this approach on the Robust 2004 test collection with 249 queries and investigated the effect of size k on the performance of the method. We considered three configurations $k = \{1, 10, \text{ and } 50\}$ where the first k queries were randomly sampled. To deal with the sampling variance we repeated the experiments 50 times, every time with a new set of initial queries, and averaged the results across the experiments. For each of the configurations, we calculated Kendall-τ after selecting 60% of the queries. As the value of k increased the associated Kendall-τ decreased. However, for $k = 10$ the Kendall-τ was still statistically larger than the Kendall-τ obtained for the random sampling of queries. As k rose to 50, the Adaptive method performed no better than Random. These results suggest that for some k we can use our iterative method to collect relevance judgments for multiple queries at a time. However, further experiments will be needed to find the optimal setting for k across the iterations, and its relationship with the total number of queries.

10. CONCLUSION

In this paper we considered the task of selecting a representative set of queries and corresponding relevance judgments to achieve an optimal approximation of the systems performance. We started with the premise that no relevance judgments are available for a query before it is selected for the relevance assessment and that only a limited assessment budget is available. Thus, we provided a mathematical model for query selection that explicitly formulated the uncertainty in the performance scores that was due to the absence of relevance judgments. The mathematical formulation showed that the optimal subset of queries should be the least correlated with each other and maximally correlated with the remaining queries. Furthermore, the total uncertainty associated with the selected queries should be a minimum.

Since the optimization problem was intractable, we proposed the Adaptive algorithm in which queries were iteratively selected and their relevance judgments were obtained after they were added to the query set. The relevance judgments of the selected queries were used to train a classifier to facilitate the selection of subsequent queries.

We demonstrated the effectiveness of the Adaptive algorithm using two TREC test collections and a Web test collection of a commercial search engine. For all three test collections, the Adaptive algorithm significantly outperformed the considered baseline methods.

Generally, the query selection methods have been criticized for their lack of generalization to previously unseen systems and multiple evaluation metrics. Our Adaptive al-

gorithm exhibited the same problems. However, we refined the Adaptive algorithm and showed that the selected query subset provides effective evaluations of new systems and can reliably be used with multiple metrics.

One of the main advantages of our query selection model is its extendibility to accommodate different sources of uncertainty in measuring systems performance. In this paper, we focused on the uncertainty of the performance estimator due to lack of relevance judgments for a query. However, other sources of uncertainty could be considered. Recent research is particularly concerned with measuring uncertainty of the systems performance due to (i) partial relevance judgments [2, 4, 20] and (ii) errors in the relevance judgments made by human assessors [6, 12]. Our future work will expand the theoretical model to incorporate additional sources of uncertainty and explore more general cost models for constructing test collections.

11. REFERENCES

[1] J. Allan, B. Carterette, J. A. Aslam, V. Pavlu, B. Dachev, and E. Kanoulas. TREC 2007 million query track. In *Notebook Proceedings of TREC 2007*. TREC, 2007.

[2] J. A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 541–548, New York, NY, USA, 2006. ACM.

[3] P. Billingsley. *Probability and Measure*. New York: Wiley, New York, NY, USA, 1995.

[4] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 268–275, New York, NY, USA, 2006. ACM.

[5] B. Carterette, V. Pavlu, E. Kanoulas, J. A. Aslam, and J. Allan. Evaluation over thousands of queries. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 651–658, New York, NY, USA, 2008. ACM.

[6] B. Carterette and I. Soboroff. The effect of assessor error on IR system evaluation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 539–546, New York, NY, USA, 2010. ACM.

[7] G. V. Cormack, C. R. Palmer, and C. L. A. Clarke. Efficient construction of large test collections. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 282–289, New York, NY, USA, 1998. ACM.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995.

[9] F. Diaz. Performance prediction using spatial autocorrelation. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 583–590, New York, NY, USA, 2007. ACM.

[10] J. Guiver, S. Mizzaro, and S. Robertson. A few good topics: Experiments in topic set reduction for retrieval evaluation. *ACM Trans. Inf. Syst.*, 27(4), 2009.

[11] C. Hauff, D. Hiemstra, L. Azzopardi, and F. de Jong. A case for automatic system evaluation. In *Proceedings of the 32nd European conference on Advances in Information Retrieval*, ECIR'2010, pages 153–165, Berlin, Heidelberg, 2010. Springer-Verlag.

[12] M. Hosseini, I. J. Cox, N. Milic-Frayling, G. Kazai, and V. Vinay. On aggregating labels from multiple crowd workers to infer relevance of documents. In *ECIR'12: Proceedings of the 34th European conference on Advances in information retrieval*, ECIR'12, pages 182–194, 2012.

[13] M. Hosseini, I. J. Cox, N. Milic-Frayling, T. Sweeting, and V. Vinay. Prioritizing relevance judgments to improve the construction of IR test collections. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 641–646, New York, NY, USA, 2011. ACM.

[14] M. Hosseini, I. J. Cox, N. Milic-Frayling, V. Vinay, and T. Sweeting. Selecting a subset of queries for acquisition of further relevance judgements. In *Proceedings of the Third international conference on Advances in information retrieval theory*, ICTIR'11, pages 113–124, Berlin, Heidelberg, 2011. Springer-Verlag.

[15] S. Mizzaro and S. Robertson. Hits hits TREC: exploring IR evaluation results with network analysis. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 479–486, New York, NY, USA, 2007. ACM.

[16] J. C. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 2000.

[17] S. Robertson. On the contributions of topics to system evaluation. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 129–140, Berlin, Heidelberg, 2011. Springer-Verlag.

[18] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 66–73, New York, NY, USA, 2001. ACM.

[19] K. Sparck Jones and K. van Rijsbergen. Information retrieval test collections. *Journal of Documentation*, 32(1):59–75, 1976.

[20] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 102–111, New York, NY, USA, 2006. ACM.

[21] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314, New York, NY, USA, 1998. ACM.