Selecting a Subset of Queries for Acquisition of Further Relevance Judgements

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

[1] University College London
{m.hosseini,ingemar}@cs.ucl.ac.uk, {trevor}@stats.ucl.ac.uk
[2] Microsoft Research Cambridge
{natasamf,vvinay}@microsoft.com

Abstract. Assessing the relative performance of search systems requires the use of a test collection with a pre-defined set of queries and corresponding relevance assessments. The state-of-the-art process of constructing test collections involves using a large number of queries and selecting a set of documents, submitted by a group of participating systems, to be judged per query. However, the initial set of judgments may be insufficient to reliably evaluate the performance of future as yet unseen systems. In this paper, we propose a method that expands the set of relevance judgments as new systems are being evaluated. We assume that there is a limited budget to build additional relevance judgements. From the documents retrieved by the new systems we create a pool of unjudged documents. Rather than uniformly distributing the budget across all queries, we first select a subset of queries that are effective in evaluating systems and then uniformly allocate the budget only across these queries. Experimental results on TREC 2004 Robust track test collection demonstrate the superiority of this budget allocation strategy.

1 Introduction

In information retrieval (IR), a test collection is used to evaluate the performance of search systems. A test collection consists of (*i*) a document corpus, (*ii*) a set of topics with corresponding search queries, and (*iii*) a set of relevance judgments for each query. Relevance judgments indicate which documents in the corpus are relevant to a particular query. When the corpus and the number of queries are small, it is feasible to acquire relevance judgments by employing a number of human assessors and, possibly, judge the relevance of each document in the collection to all the queries. However, when the corpus and the number of test queries are large, this is no longer the case due to both the economic cost and the time involved.

In order to address this issue, the IR community has adopted a method of *pooling* candidate documents from the retrieval results of the participating systems [11]. Each participating system contributes a set of documents it retrieves for a query, e.g. the top-100 documents, and the set of unique documents is then judged for relevance by the human assessors. The total number of pooled documents is typically much smaller than the number of documents in the corpus. However, since documents are provided by systems that are being compared, the resulting document pool is expected to be effective in assessing their relative performance [14].

Gathering relevance assessments has an associated cost which, in its simplest form, depends on the number of queries and the number of documents per query that need to be assessed. However, the cost is not the only consideration when creating effective test collections. The accuracy and reusability of the test collections are also very important. A test collection is accurate if the participating systems' performance are precisely evaluated. In addition, a test collection is reusable if has no inherent bias that might affect evaluation of new as yet unseen systems.

In particular, a test collection may not be reusable if a new system, in response to queries in the test set, retrieves many documents that are not in the document pool. In this situation, (*i*) the previously unjudged documents must either be judged non-relevant [12], (*ii*) the new documents are assigned a probability of relevance and new systems' performance are measured by using metrics designed for incomplete relevance judgments, e.g. MTC [3], or (*iii*) additional user relevance judgments must be obtained for these documents. Assuming the documents are non-relevant potentially biases the test collection - only future systems that behave like the original participating systems will be evaluated accurately [5]. Assigning a probability of relevance may cause a high uncertainty in evaluation when there are a large number of unjudged documents for new systems [4], and acquiring additional user judgments can be expensive.

We assume a limited budget is available to build additional relevance judgements for previously unjudged documents retrieved by new systems. In this paper, we examine whether it is better to uniformly allocate the budget across all queries, or select a subset of queries and allocate the budget only to the selected queries to get deeper judgments per query at the same cost.

Selection of the subset is strongly related to the query selection problem. The query selection problem is motivated by empirical evidence that the ranking of systems based on many queries can be reproduced with a much reduced set of queries [8]. Thus, ideally, we would identify a minimal subset of queries that still enables a reliable evaluation of the existing and new systems. Furthermore, the gain from reducing the number of queries can be re-directed to increase the number of documents judged per query. Our hypothesis is that, given a fixed budget, a smaller but representative set of queries with a greater number of judged documents per query will increase the accuracy of ranking new systems.

In this paper, we introduce a query selection approach and present results of its application to obtain an accurate ranking of new systems' performance. In Section 3, we formalize the query selection problem and propose two query selection algorithms, a *greedy* algorithm and an optimization based on a *convex* objective. Section 4 then describes our experimental results, based on the Robust track of TREC 2004. Two different experiments are described. The first experiment is concerned with how a subset of queries performs when evaluating new systems that did not participate in the original pooling (generalization). The second experiment examines the problem of allocating new relevance judgments to queries, and compares a uniform allocation across all queries to a (deeper) uniform allocation across a subset of queries. Finally, Section 5 concludes by summarizing the results of our research and outlining future directions. However, before proceeding we first discuss related work.

2 Related Work

Research presented in this paper starts with [13], which suggests that reliable IR evaluation requires at least 50 queries and that including a larger number of queries makes for a better test collection. Given these results, it is not surprising that recent studies have concentrated on IR evaluations with large query sets [6], and methods for reducing the number of relevance judgments per query in order to make relevance assessment feasible [3], as well as introducing evaluation metrics for partially judged result sets [2].

Following the belief that a larger query set is desirable, the Million Query track of TREC 2007 [1] was the first to include thousands of queries. The Million Query track used two document selection algorithms, proposed by [3] and [2], to acquire relevance judgments for more than 1,800 queries. The experiments on this test collection showed that a large number of queries with a few judgements (*i*) results in an accurate evaluation of participating systems, and (*ii*) is more cost-effective than evaluation conducted by fewer queries with more judgements. However, due to the small number of documents assessed per query, the reusability of such a test collection still remains questionable. Indeed, Carterette et al. [5] demonstrated that the Million Query track of TREC 2009 is not usable for assessing the performance of systems that did not participate in pooling documents.

Guiver et al. [8] showed that some queries or query subsets are better in predicting systems' overall performance than others. Finding a representative subset of queries is a combinatorial problem with NP-hard complexity. Little work is available on practical approaches that could be used to select a subset. Mizzaro and Robertson [9] suggested prioritizing queries based on a per-query measure, hubness, that indicates how well a query contributes in system evaluation. Guiver et al. [8] showed that a representative subset consists of not only queries that individually predict systems' performance with high precision but also queries that are weak in prediction on their own. They also suggested a greedy algorithm for query subset selection. However, Robertson [10] showed that a subset that is selected by the greedy algorithm suffers from overfitting and is not able to generalize to new systems. In this paper, we propose a convex method for query selection and show that its generalization is superior to the greedy algorithm.

3 Expanding Relevance Judgements

A test collection consists of a document corpus, a set of N queries $Q_N = \{q_1, q_2, ..., q_N\}$, and the associated relevance judgements that are gathered based on documents returned by a set of L participating systems, S_L. More precisely, each of the systems returns a number of results for each of the N queries. A pooling technique or a recently proposed document selection method, e.g. [3], is used to select a subset of documents returned by each of the L systems to build relevance judgements. The aim of the test collection is not only to accurately evaluate the performance of the participating systems but also to reliably estimate the performance of new systems that did not participate in pooling.

We begin with the assumption that a system can be reliably evaluated and compared with other systems if we manually assess a significant portion of the document corpus or, at least, a large number of documents retrieved by each individual system. Therefore, if new systems return many new (unjudged) documents, the current relevance judgements are insufficient to reliably assess their performance. In this situation, we assume that there is a limited budget to build relevance judgements for a subset of the new documents. How should we spend the limited budget to acquire additional relevance judgments? We could consider all queries and use a heuristic method to pool a few documents per query. Alternatively, we could select a representative subset of queries that closely approximates systems' overall performance, and allocate the budget only to the selected queries. The final solution is likely to include elements of both these approaches. In this paper, we assume the pooling method [11] is used to select documents at the query level and restrict our attention to the task of choosing queries. In the following, we formalize the query selection problem and describe three solutions.

3.1 The Query Selection Problem

Evaluating the L systems on the N queries forms a $L \times N$ performance matrix X. Each row represents a system, and each column a query. An entry, $x_{i,j}$, in X denotes the performance of the i^{th} system on the j^{th} query. We also consider the column vector M, as the average performance vector. The values of M indicate the average performance of individual systems over all queries. Thus, if the individual elements, $x_{i,j}$, measure average precision (AP), then the elements of M represent mean average precision (MAP) scores.

Now let $\Phi = \{j_1, ..., j_m\}$ be a subset of $\{1, 2, ..., N\}$ with $1 \leq m \leq N$ and consider the subset of queries $\{q_j : j \in \Phi\}$. We define M_Φ as the vector of systems' average performance measured on the subset of queries indexed in Φ. The aim of a query selection method is to find a subset of queries of a particular size, m, such that the corresponding vector M_Φ closely approximates the vector M. There are several measures to evaluate how well M_Φ approximates M. The IR community usually uses Kendall-τ rank correlation coefficient to measure the closeness between two vectors of real values. In this paper, we assume that the objective of a query selection method is to maximize the Kendall-τ coefficient between the systems' rankings induced by M_{Φ} and M.

Finding the subset of queries of size m that maximizes this objective is NP-hard, and a brute force search is only practical for small N [8]. In the following, we introduce three computationally practical selection algorithms that approximate the optimal solution.

Random Sampling: One may use uniform random sampling to select a subset of queries. In this method, all queries are given the same chance to be selected. We use uniform random sampling as the baseline in our experiments. That is, for a given subset size, m, we randomly select a subset of m queries.

Greedy Algorithm: A forward selection scheme can be used to approximate the optimal subset of queries. That is, when m=1, the optimal solution is the query whose column score in matrix X leads to a systems' ranking that has the highest Kendall-τ rank correlation with the systems' ranking induced by M. For every $m > 1$ we use the best subset of size m-1 and select the m^{th} query from the queries indexed in Φ^c (the complement set of Φ) that maximizes the Kendall-τ between the two ranks induced by M and corresponding M_{Φ}. This greedy algorithm is fast and tractable but it is not guaranteed to find the best subset since the best subset of size m does not necessarily contain all the queries selected for the best subset of size m-1 [8]. In addition, applying a greedy algorithm may result in a subset that highly depends on the participating systems and does not accurately rank new systems [10].

Convex optimization can be used to find a globally optimum solution to an approximation of this problem, as outlined below.

Convex Optimization: For the i^{th} system, the average performance of this system, μ_i, based on queries in Q_N is

$$\mu_i = N^{-1} \sum_{j=1}^{N} x_{ij} = N^{-1} x_i e$$

where $e \in \{1\}^{N \times 1}$, is a column vector of N ones and $x_i \in R^{1 \times N}$ is the i^{th} row of matrix X. In addition, consider an activation vector $d \in \{0, 1\}^{N \times 1}$ such that $d_j = 1$ if $j \in \Phi$ and $d_j = 0$ otherwise. The average performance of the i^{th} system on the subset Φ of size m is then

$$\mu_{i\Phi} = m^{-1} \sum_{j \in \Phi} x_{ij} = m^{-1} x_i d$$

where m is the number of selected queries. Also μ_i and $\mu_{i\Phi}$ are the i^{th} elements of vectors M and M_{Φ} respectively. While the greedy algorithm optimizes for Kendall-τ, which we also use as the evaluation measure in our experiments, we cannot use this measure in convex optimization. Instead, we use the *residual sum of squares* to minimize the sum of differences between pairs of μ_i and $\mu_{i\Phi}$.

$$\min_{d} \sum_{i=1}^{L} \left(N^{-1} x_i e - m^{-1} x_i d \right)^2 ; \qquad \text{subject to: } \parallel d \parallel_0 \leq m \tag{1}$$

where $\| \, . \, \|_0$ is the L_0 norm that simply counts the number of non-zero elements in d and controls the size of the subset, m.

To minimize Equation 1, we use *convex relaxation* that replaces the above minimization function with a convex function that admits tractable solutions. Note that the optimization function in Equation 1 is not convex due to the L_0 norm constraint. We alter this constraint and convert Equation 1 to a convex form by removing the restriction for having only binary values, i.e., we replace d by $\beta \in [0,1]^{N \times 1}$ which contains real values bounded between 0 and 1 such that if j^{th} query is selected, $\beta_j > 0$, otherwise $\beta_j = 0$. In addition, we control the number of selected queries, m, based on the budget available to build additional relevance judgements. We denote Ω as the budget needed to build relevance judgements for all previously unjudged documents that are returned by new systems. Also B denotes the limited budget ($0 \leq B \leq \Omega$) that is available to build additional relevance judgements. We replace the L_0 constraint with the linear constraint $\sum_{j=1}^{N} \beta_j \leq \frac{B}{\Omega}$. Therefore, choosing a subset is now based on solving the following convex optimization function:

$$\min_{\beta} \sum_{i=1}^{L} \left(N^{-1} x_i e - x_i \beta \right)^2; \quad \text{subject to:} \sum_{j=1}^{N} \beta_j \leq \frac{B}{\Omega} \tag{2}$$

To solve this convex optimization we use Least Angle Regression (LARS) [7] in our experiments. LARS generates the optimal subsets of size $1, 2, ..., N$ as the budget B varies from 0 to Ω. Hence, we select all queries for which β_j is non-zero, and by varying the budget B, we can control the number, m, of queries in the subset.

4 Experimental Evaluation

Our experimental investigations were performed using the Robust track of TREC 2004 consisting of 249 topics (queries), and 14 sites with a total of 110 runs. Normally, organizations participating in a TREC experiment register as *sites* and submit a number of experimental *runs* for evaluation. These runs often represent variations on a retrieval model's settings. For our purposes we considered runs as search systems, taking special care when considering runs from the same site.

In order to evaluate properties of our query selection method, we partitioned the set of all experimental runs into *participating* and *new* systems. In addition, in order to ensure that new systems were truly different from the participating ones, we held out as new systems not only individual runs but also the entire set of runs from the same site. Furthermore, during computation of performance metrics, we removed the documents that were uniquely retrieved by the new (held-out) systems from the pool.

We describe the results of two experiments. In the first experiment, we select subsets of varying size m, using each of the three query selection methods. We then compare how well these subsets rank new systems based on the relevance judgements provided by participating systems. This experiment is intended to determine which of the three query selection methods selects the best subsets for ranking new systems, i.e. which algorithm provides subsets that generalize to new systems. The second experiment compares the performance of two budget allocation methods, uniform allocation across all queries versus uniform allocation across a subset of queries, that are used to expand relevance judgements based on documents retrieved by new systems.

4.1 Generalization

We assess generalization of query subsets selected by the three algorithms, namely *random*, *greedy*, and *convex*. For the random query sampling method we report the average of 1000 random trials.

Experimental Setup: We first partitioned systems into participating and new systems. Using the participating systems we constructed initial pools for the full set of queries. We then constructed the performance matrix X, comprising effectiveness scores (AP) of the participating system-query pairs. We selected a subset of queries of size m using one of the three query selection algorithms. Next, for all queries in the subset we used corresponding relevance judgements collected from the initial pool to measure the performance of new systems and construct the corresponding vector M_Φ. Note that in the convex method the vector M_Φ weights each query equally, i.e. the value of β coefficients, calculated as part of the solution to Equation 2 are ignored. We do this because the sample mean of AP scores of selected queries is an unbiased estimator of MAP calculated over the full set of queries, and amongst all the unbiased estimators, it has the smallest variance. We also computed the vector M of new systems based on all the queries and their original set of relevance judgements collected in the Robust track. The generalization was expressed through the Kendall-τ coefficient for the new systems' rankings induced by M_Φ and M. If the two rankings were similar, the reduced set of queries was deemed useful for evaluating new systems.

We used a repeated random sub-sampling technique to split systems into participating and new systems. At each trial, we randomly selected 40% of sites, and labeled their runs as new systems. The remaining runs were treated as participating systems and used to build documents pools. For each of the query selection methods, we selected a subset of queries for a given size m. We then measured their generalization as explained above.

We repeated the sampling process over 10 trials and averaged their results to produce single estimates. We note that 10 trials of sampling ensured that the runs of each site were at least assigned once to the participating set and once to the new set. The advantage of this technique over k-fold cross validation is that the proportion of the training/test split is not dependent on the number of iterations (folds). Consequently, a considerable subset of runs, about 45 out of 110 on average, were held out at each trial as new systems.

Experimental Results: The generalization of the selected subsets by each of the query selection methods is shown in Figure 1 for subset sizes between 1 and 70. As seen, the convex optimization outperforms the greedy method

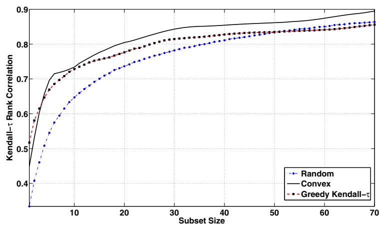

Fig. 1. The Kendall-τ of the three query selection methods as a function of query subset size. The total number of queries is 249.

and random sampling for almost all subset sizes. The difference between the Kendall-τ scores for convex and greedy is 0.04 on average, which is equivalent to correctly ordering 19 additional pairs of systems. The performance of the greedy method degrades toward random sampling for subset sizes bigger than 30. This suggests that as the size of the subset increases, the greedy method over-learns and consequently lacks generalizability. We also note that the differences between the three methods become negligible as the size of subset approaches the full set of queries. This happened in our experiment after selecting 174 queries from a total of 249.

4.2 Comparing Two Relevance Judgment Allocation Methods

In this section, we assume a fixed budget is available to collect new relevance judgments. We examine two methods for allocating the budget across queries. In the first method, the resources are equally spread across all queries. For example, if the budget can cover only 200 new judgments and there are 100 queries, we judge two new documents per query. In the second method, we select a subset of queries and then allocate the budget equally across them.

Experimental Setup: We first randomly selected a subset of sites and used their experimental runs as participating (held-in) systems. We then analyzed the held-out sites and distinguished between those sites that performed similarly to the held-in sites, i.e. there was considerable overlap in the documents retrieved by these sites and the held-in sites, and those sites that were very different from the held-in sites. To do this we applied the reusability measure proposed in [4] to measure the extent to which the corresponding pooled documents covered the documents retrieved by the held-out systems. For each held-out system and each query we considered the ranked list of documents and computed the average reuse (AR),

$$AR(q) = \frac{1}{judged(q)} \sum_{i} \frac{judged@i(q)}{i}$$

where $judged@i(q)$ was the number of judged documents in the top-i results of the held-out system for query q, and $judged(q)$ was the total number of documents judged for query q. In addition, we defined the mean average reuse (MAR) as the average of AR values for a system over the full set of queries.

We separated held-out sites into two groups based on the average of the MAR scores of their runs: those with high MAR across runs that could be evaluated using the existing relevance judgments, and the second group with runs that had low MAR and thus required additional relevance judgments in order to be evaluated. The first group of runs formed the *auxiliary set* of systems, and the others were considered as *new systems.* We used the new systems to evaluate the different resource allocation methods. The full experiment is as below:

1. Pick s_1 sites at random. The runs of these sites are treated as participating systems.
2. For each query, construct the *initial* pool of top-k_0 documents retrieved by participating systems and build associated relevance judgments. Compute the performance matrix X.
3. Compute the MAR for the runs that did not participate in the pooling. Average the MAR scores across the runs from the same site and produce average reuse score for each site.
4. Pick s_2 sites with the smallest scores and treat their runs as *new systems.* The remaining runs are *auxiliary systems* that can be evaluated with the existing relevance judgments. Their performance values are added to the performance matrix X. Note, however, that the auxiliary systems do not contribute to the document pool.
5. Given a budget B, select a subset of m queries using the convex optimization method.
6. Acquire additional relevance judgments in one of two ways:
   (a) **Subset:** For each of the m selected queries assess an additional k_1 documents contributed by the new systems where k_1 is adjusted based on B.
   (b) **Uniform:** For each of the N queries assess an additional k_2 documents contributed by the new systems where $m \times k_1 = N \times k_2$.
7. Add the newly judged documents to the initial pool and compute the effectiveness scores for the new systems.

Experimental Results: We applied the above steps across 10 trials. In each trial, we randomly chose a different set of participating sites, s_1=1, 3 or 5. The runs of the remaining sites were partitioned into auxiliary and new systems based on their reusability scores. We considered the s_2 lowest scoring sites and chose their runs to be new systems, where s_2 = 3, 6 or 8. The auxiliary sets comprised 5, 6 or 7 sites. To construct the initial pools we considered the top-k_0 documents from each participating system, where $k_0 = 10$ or $k_0 = 30$. Assuming a fixed budget, $B = \{1, 3 \text{ or } 5\} \times 10^4$ and a performance matrix X composed of participating and auxiliary systems, we used the convex optimization method to select a subset of queries. As B increased, the number of selected queries also increased. In our experiments, the size of the subsets varied between 14 to 237 with a median of 69.

Table 1 compares the performance statistics for the Robust 2004 track test collection before and after acquiring new relevance judgments in 12 different experimental configurations. The values given in the table are Kendall-τ scores

– averaged over 10 trials – between the ranking of new systems induced by the initial pool (containing top-k_0 documents returned by participating systems) or one of the two resource allocation methods (”uniform” and ”subset”) and the ranking induced by MAP scores that are measured over the full set of queries and by using the original pools (TREC *qrels*). Also, p^+ counts additional pairs of systems that are correctly ordered by the subset method when compared to the number of pairs correctly ordered by the uniform method. In addition, Ω is the number of judgements needed to build relevance judgements for all previously unjudged documents that are returned by new systems in a pool of depth 100.

We note that if the difference in average performance scores of two systems is not statistically significant, it is completely reasonable that they may be ordered differently when evaluated over a subset of queries. Having such tied systems in a test set increases the probability of a swap and consequently decreases Kendall-τ. This is because the Kendall-τ is not able to distinguish between pairs of systems with and without significant differences. This is the case in Robust track test collection in which about 30% of pairs are ties, when measured by a paired t-test at significance level 0.05. In Table 1, the Kendall-τ scores in parentheses are calculated by only considering the pairs of systems with a statistically significant difference in MAP.

The positive effect of increasing the number of sites s_1 that contribute to the document pool, can be observed from the experiments 1, 7 and 10 for which s_1 is varying from 1 to 5, with $B = 1 \times 10^4$. In addition, increasing s_1 or k_0 increases the average reuse scores of held-out sites and, consequently, reduces the number of new systems and the amount of Ω. This can be seen from the experiments 1-9. Diversifying the set of participating systems by increasing s_1 while keeping k_0 constant causes a bigger improvement in Kendall-τ than the opposite, i.e., increasing k_0 and keeping s_1 constant. This is demonstrated by

Table 1. Results for TREC 2004 Robust runs evaluated by MAP. The first six columns report experimental parameters. The next three columns report the Kendall-τ between the rankings induced by M and M_Φ of new systems for the initial pool and each of the two budget allocation methods. The last column (p^+) counts additional pairs of systems that are correctly ordered by the subset method against the uniform method. The values in parentheses are measured by only considering pairs of new systems with a statistically significant difference.

exp.#	s_1	s_2	k_0	Ω	B	$\frac{B}{\Omega}$	Kendall-τ			p^+
							initial pool	uniform	subset	
1					10,000	0.06		0.54 (0.63)	0.6 (0.66)	95 (50)
2	1	8	10	163,842	30,000	0.18	0.42 (0.49)	0.57 (0.66)	0.64 (0.70)	110 (63)
3					50,000	0.31		0.61 (0.69)	0.68 (0.74)	111 (79)
4					10,000	0.09		0.66 (0.74)	0.73 (0.79)	53 (44)
5	1	6	30	107,817	30,000	0.28	0.54 (0.61)	0.70 (0.77)	0.77 (0.81)	62 (42)
6					50,000	0.46		0.75 (0.80)	0.82 (0.85)	62 (46)
7					10,000	0.1		0.70 (0.76)	0.76 (0.83)	53 (51)
8	3	6	10	104,580	30,000	0.29	0.60 (0.65)	0.75 (0.81)	0.82 (0.87)	62 (53)
9					50,000	0.48		0.82 (0.80)	0.89 (0.89)	71 (79)
10					10,000	0.19		0.87 (0.91)	0.90 (0.95)	7 (11)
11	5	3	10	53,535	30,000	0.56	0.70 (0.77)	0.92 (0.94)	0.96 (0.98)	11 (8)
12					50,000	0.93		0.99 (1.0)	0.98 (1.0)	-2 (0)

the experiments 4 and 7 where $s_2 = 6$ and $B = 1 \times 10^4$. This result is consistent with observations by Carterette et al. [4] that a higher diversity of participating systems results in a better ranking of new systems. In experiments 1-9 where the total cost, Ω, is considerably bigger than the available budget, B, the subset method significantly outperforms the uniform method. As B approaches Ω, the amount of improvement decreases such that in the last experiment ($s_1 = 5$ and $\frac{B}{\Omega} = 0.93$) the Kendall-τ obtained by the uniform method is bigger than the Kendall-τ for subset. We note that, as B approaches Ω, the number of selected queries gets closer to the total number of queries in the test collection. Therefore, when $\Omega \approxeq B$, the difference between the performance of subset and uniform method is negligible.

5 Discussion and Conclusion

In this paper, we considered the problem of expanding the relevance judgements of a test collections in order to better evaluate the performance of new systems. Given a fixed budget, we investigated whether it is better to uniformly allocate the budget across all the queries in the test collection, or only to a subset of queries. Our hypothesis was that a smaller but representative set of queries with a greater number of judged documents per query increases the accuracy of ranking new systems.

The hypothesis was tested using the Robust Track of TREC 2004. Three methods for determining a subset of queries were considered, referred to as random, greedy and convex. Experimental results demonstrated that query selection based on a convex optimization provided better generalization. The difference between the Kendall-τ scores for convex and greedy was 0.04 on average, which is equivalent to correctly ordering 19 additional pairs of systems. For a fixed budget, we then compared how well new systems were ranked, based on a uniform allocation across (*i*) all queries and (*ii*) a subset of queries chosen using the convex method. A variety of different experimental configurations were tested, which (*i*) varied the number of participating sites (1, 3 or 5), (*ii*) the number of new sites (3, 6 or 8), (*iii*) the size of the top-k_0 documents contributing to the initial pool, and (*iv*) the budget available ($B = \{1, 3 \text{ or } 5\} \times 10^4$ additional relevance judgments). When B was much smaller than the required budget, Ω, to build complete relevance judgements, allocating the budget uniformly across a subset of queries performed better than uniform allocation across all queries. As B approached Ω the difference between two methods became negligible.

There are a variety of avenues for future work. First, while the convex optimization was shown to exhibit better generalizability, its objective function does not explicitly consider generalization. However, in practice we could, perhaps, indirectly measure generalizability based on predicting the number of unseen relevant document for each query, based on work in [14], and a corresponding cost term could be incorporated into our objective function.

Ultimately, a budget allocation strategy should be prioritizing not just queries, but individual query-document pairs. Considerable work has been done to minimize the number of documents pooled for a given query [3,2]. These techniques are largely complementary to the experiments in this paper, and future work is needed to combine these different approaches.

References

1. Allan, J., Carterette, B., Aslam, J.A., Pavlu, V., Dachev, B., Kanoulas, E.: TREC 2007 million query track. Notebook Proceedings of TREC 2007. TREC (2007)
2. Aslam, J.A., Pavlu, V., Yilmaz, E.: A statistical method for system evaluation using incomplete judgments. In: SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 541–548. ACM, New York (2006)
3. Carterette, B., Allan, J., Sitaraman, R.: Minimal test collections for retrieval evaluation. In: SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 268–275. ACM, New York (2006)
4. Carterette, B., Gabrilovich, E., Josifovski, V., Metzler, D.: Measuring the reusability of test collections. In: WSDM 2010: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 231–240. ACM, New York (2010)
5. Carterette, B., Kanoulas, E., Pavlu, V., Fang, H.: Reusable test collections through experimental design. In: SIGIR 2010: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 547–554. ACM, New York (2010)
6. Carterette, B., Pavlu, V., Kanoulas, E., Aslam, J.A., Allan, J.: Evaluation over thousands of queries. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 651–658. ACM, New York (2008)
7. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Annals of Statistics 32, 407–499 (2004)
8. Guiver, J., Mizzaro, S., Robertson, S.: A few good topics: Experiments in topic set reduction for retrieval evaluation. ACM Trans. Inf. Syst. 27(4) (2009)
9. Mizzaro, S., Robertson, S.: 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 2007, pp. 479–486. ACM, New York (2007)
10. Robertson, S.: On the contributions of topics to system evaluation. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 129–140. Springer, Heidelberg (2011)
11. Sparck Jones, K., van Rijsbergen, K.: Information retrieval test collections. Journal of Documentation 32(1), 59–75 (1976)
12. Voorhees, E.M.: The philosophy of information retrieval evaluation. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (eds.) CLEF 2001. LNCS, vol. 2406, pp. 355–370. Springer, Heidelberg (2002)
13. Voorhees, E.M., Buckley, C.: The effect of topic set size on retrieval experiment error. In: SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 316–323. ACM, New York (2002)
14. Zobel, J.: How reliable are the results of large-scale information retrieval experiments? In: SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 307–314. ACM, New York (1998)