If I Had a Million Queries

Ben Carterette*, Virgil Pavlu†, Evangelos Kanoulas†,
Javed A. Aslam†, and James Allan‡

* Dept. of Computer and Info. Sciences, University of Delaware, Newark, DE
† College of Computer and Info. Science, Northeastern University, Boston, MA
‡ Dept. of Computer Science, University of Massachusetts Amherst, Amherst, MA

Abstract. As document collections grow larger, the information needs and relevance judgments in a test collection must be well-chosen within a limited budget to give the most reliable and robust evaluation results. In this work we analyze a sample of queries categorized by length and corpus-appropriateness to determine the right proportion needed to distinguish between systems. We also analyze the appropriate division of labor between developing topics and making relevance judgments, and show that only a small, biased sample of queries with sparse judgments is needed to produce the same results as a much larger sample of queries.

1 Introduction

The sizes of collections used in Information Retrieval research has been growing at an astonishing pace, with every decade seeing at least an order-of-magnitude increase in research collection size. Batch evaluation of retrieval systems requires *test collections* that augment these collections with a set of information needs and judgments of the relevance of documents in the collection to those needs. While collection sizes keep increases, the budget for relevance judgments does not. Reliable batch evaluations will rely more and more on the relevance judgments being selected intelligently and the inferences about systems made robustly despite many missing judgments. In this line of work, Sakai [1] applied traditional measures to condensed lists of documents obtained by filtering out all unjudged documents. Carterette et al. [2] and Moffat et al. [3] selected a subset of documents to be judged based on the benefit they provide in ranking systems and in identifying the best systems, respectively. Aslam et al. [4] used random sampling to estimate the actual values of evaluation measures when relevance judgments are incomplete.

One of the stated goals of the TREC Million Query Track is to evaluate the reliability and robustness of different methods for selecting documents to judge. The 2007 track, which evaluated 24 systems from 10 sites, suggests that two such methods do indeed produce reliable evaluations when used to select documents from a very large corpus of 25 million documents, and further that only a relatively small number of queries and judgments are needed to draw robust conclusions.

In the 2008 track, queries were categorized by length and a proxy measure of their appropriateness to the corpus [5]. This allows a more detailed analysis of not only robustness of evaluation but also of what the query sample of a test collection should look like. In addition, queries were assigned different target numbers of judgments, which allows more detailed analysis of the proper level of budgeting for relevance judgments within that query sample.

In this work we analyze (theoretically and empirically) the query sample and relevance judgment allotment needed to draw reliable and robust conclusions about retrieval systems. We begin in Section 2 by briefly describing the methods used to select judgments and evaluate systems. Section 3 gives an overview of the results of evaluating systems with many queries judged sparsely. Section 4 presents our analysis of judgment targets and query samples, in which we show that a biased sample may be better at distinguishing between systems. We conclude in Section 5.

2 Methods

We implemented two recent methods for selecting documents to judge and estimating evaluation methods with incomplete judgments. One emphasizes the ranking of systems while the other emphasizes correct estimates of measures.

2.1 Minimal Test Collections

The Minimal Test Collections (MTC) method works by identifying documents that will be most informative for understanding performance differences between systems by some evaluation measure (in this case average precision). Details on the workings of MTC can be found elsewhere [6, 2, 7]. Here we focus on MTC estimates of evaluation measures.

First, we consider each document i to have a distribution of relevance $p(X_i)$. If the document has been judged, then $p(x_i) = 1$. Then we consider a measure to be a random variable expressible as a function of document relevance random variables X_i. For example, precision can be expressed as a random variable that is a mean of the random variables for those documents at ranks 1 through k.

If the measure is a random variable in terms of document random variables, then it has a distribution over possible assignments of relevance to unjudged documents. Note that this applies to measures averaged over queries as well as to measures for a single query. Abstracting even higher, this produces a distribution over possible rankings of systems: there is some probability that system 1 is better than system 2, which in turn is better than system 3; some probability that system 1 is better than system 3, which in turn is better than system 2, and so on. It can be shown that the maximum a posteriori ranking of systems is the one by the expected values of the evaluation measure of interest over the possible relevance assignments.

Calculating the expectation of an evaluation measure is fairly simple. Given the probability that document i is relevant $p_i = p(X_i = 1)$, we define $\mathbf{E}prec@k =$

$\frac{1}{k}\sum_{i=1}^{k} p_i$, $\mathbf{E}R\text{-}prec \approx \frac{1}{\mathbf{E}R}\sum_{i=1}^{\mathbf{E}R} p_i$, and $\mathbf{E}AP \approx \frac{1}{\mathbf{E}R}\sum_{i=1}^{n} p_i/i + \sum_{j>i} p_i p_j/j$, where $\mathbf{E}R = \sum_{i=1}^{n} p_i$. Though MTC is designed for ranking, in this work we largely present expectations of evaluation measures for individual systems as well as the probability of a relative ordering of two adjacent systems.

2.2 Statistical Average Precision (statAP)

In statistical terms, average precision can be thought of as the mean of a population: the elements of the population are the relevant documents in the document collection and the population value of each element is the precision at this document for the list being evaluated. This principle is the base for several recently proposed evaluation techniques [8, 4, 9, 7]. StatAP is a sample-and-estimate technique defined by the following two choices.

Stratified Sampling, as developed by Stevens [10, 11], is straightforward for our application. Briefly, it consists of bucketing the documents ordered by a chosen prior distribution and then sampling in two stages: first sample buckets with replacement according to cumulative weight, then sample documents inside each bucket without replacement according to selection at the previous stage.

Generalized ratio estimator. Given a sample S of judged documents along with inclusion probabilities, in order to estimate average precision, $statAP$ adapts the generalized ratio estimator for unequal probability designs [12].(very popular on polls, election strategies, market research etc). For our problem, the population values are precisions at relevant ranks; so for a given query and a particular system determined by ranking r(.) we have (x_d denotes the relevance judgment of document d) :

$$statAP = \frac{1}{\widehat{R}} \sum_{d \in S} \frac{x_d \cdot \widehat{prec@r}(d)}{\pi_d}$$

where $\widehat{R} = \sum_{d \in S} \frac{x_d}{\pi_d}$ and $\widehat{prec@k} = \frac{1}{k}\sum_{d \in S, r(d) \leq k} \frac{x_d}{\pi_d}$ are estimates the total number of relevant documents and precision at rank k, respectively, both using the Horwitz-Thompson unbiased estimator [12].

Confidence intervals. We can compute the inclusion probability for each document (π_d) and also for pairs of documents (π_{df}); therefore we can calculate an estimate of variance, $\widehat{var}(statAP)$, from the sample. Given tha the set of queries is chosen randomly and independently, and taking into account the weighting scheme we are using to compute the final MAP (see Results), we compute an estimator for the MAP variance. Assuming normally distributed $statMAP$ values, a 95% confidence interval is given by $\pm 2std$ or $\pm 2\sqrt{\widehat{var}(statMAP)}$.

3 Experiment and Results

As with the 2007 MQ track, the corpus is GOV2 and we started with a sample of 10,000 queries from the log of a commercial search engine. Assessors were allowed to select a query from a list of 10 to "backfit" into a topic definition

category	8	16	32	64	128	total
short-govslant	95 (7.87)	55 (15.58)	29 (29.93)	13 (58.85)	4 (117.50)	196 (18.92)
short-govheavy	118 (7.85)	40 (15.18)	26 (30.27)	10 (58.60)	3 (117.67)	197 (16.54)
long-govslant	98 (7.72)	52 (15.60)	26 (30.38)	13 (58.31)	8 (116.88)	197 (20.56)
long-govheavy	92 (7.79)	57 (15.32)	21 (29.95)	14 (59.29)	10 (114.40)	194 (21.61)
total	403 (7.81)	204 (15.43)	102 (30.14)	50 (58.78)	25 (116.08)	784 (19.40)

Table 1. Number of queries and number of judgments per query in parantheses.

and then judge. Eight participating sites submitted a total of 25 runs based on various retrieval strategies (BM25, metasearch, inference networks, etc).

3.1 Queries and Judgments

Queries are categorized by two features:
· **long/short:** queries with more than 5 words are considered "long".
· **govheavy/govslant:** all queries used have at least one known user click in the .gov domain; the ones with more than 3 clicks are considered "heavy".

To ensure a uniform distribution over categories, assessors picked from 10 queries with the same category. The category rotated round-robin. To ensure that there were queries with varying numbers of judgments, a target number was assigned after the assessor selected a query and completed topic definition. The targets increased by powers of 2 (8, 16, 32, 64, or 128), and were assigned to ensure a roughly equal amount of total judging effort for each target: the number of queries with 8 judgments would be twice the number with 16, which in turn would be twice the number with 32, and so on.

Selection methods alternated to pick documents to judge. Because of "collisions" (both methods selecting the same document) the total number of judgments could have been less than the target. In the end we obtained 15,211 judgments for 784 queries; Table 1 shows the distribution of queries and judgments by category and target. Note that the frequency of collisions is quite low.

Of the 15,211 judgments, 2,932 (19%) were relevant. "Govheavy" queries had substantially more relevant documents than "govslant" queries (24% to 15%), indicating that it is a good proxy for appropriateness to corpus. "Short" queries had more relevant documents than "long" queries (21% to 18%), perhaps indicating that a topic definition based on a short query is more fluid than one based on a long query. There were 220 queries for which no relevant documents were found; 198 of these had a target of 8 or 16 judgments.

3.2 Evaluation Results

Both evaluation methods estimate average precision for each run and each query. We calculated a weighted mean of APs to account for the fact that we have 16 times as many queries with 8 judgments as with 128; we did not want queries with 8 judgments to dominate the evaluation as they would with a

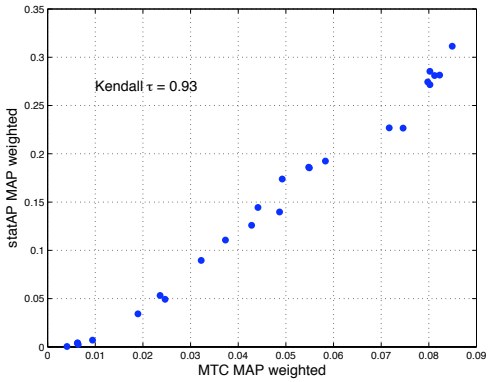

Fig. 1. MTC and statAP weighted-MAP correlation

straight average. Weighted MAP is calculated as $wMAP = \frac{1}{5}\sum_{j=1}^{5} MAP_j = \frac{1}{5}\sum_{j=1}^{5}\frac{1}{Q_j}\sum_{q\in j} AP_q$, where MAP_j is averaged over all queries at level j ($= 2^{j+2}$ target judgments) and Q_j is the number of queries at level j. Table 2 shows the weighted MAP estimates by both methods for each of the 25 runs.

In the absence of any traditional evaluation (in which many more documents are judged for each query), the best indication of being close to the "true" ranking is the correlation of the two evaluation methodologies. Their mechanisms for estimation are fundamentally different, so any correlation is much more likely due to correct estimation rather than other reasons. Figure 1 illustrates the weighted MAP scatterplot, with a Kendall's τ=.93. This tracks the results observed in the previous year's track [6].

Furthermore, the two methods continue to correlate well even when breaking queries and judgments into subsets by category, target, or method. Perhaps the best indicator that we have found the "true" ranking is that the two methods correlate well even when evaluated over only the documents they selected ($\tau = 0.87$), and even when evaluated over only the documents selected by *the other method* ($\tau = 0.91$)—despite very little overlap between the two methods.

Other Measures We concentrate our evaluation methods, results and analysis on MAP. However, to test whether the other measures (e.g., R-prec, p@10) do a fair job of ranking systems, we correlated system rankings by one measure over a set of queries/judgments to system rankings by another measure over the same set of queries/judgments (Table 2). They tend to correlate very well ($\tau \approx .9$), suggesting that even though our judgments were acquired to rank systems by MAP estimates, they can be used for other measures reliably.

4 Analysis

The above results are based on a large sample of 784 sparsely-judged queries distributed uniformly over four categories. The next step is to determine the

	MTC					statAP				
system	wMAP	conf	Rprec	prec@10	prec@30	wMAP	conf	Rprec	prec@10	prec@30
hedge0	0.0041	1.000	0.048	0.022	0.030	0.0004	±0.000	0.000	0.001	0.001
vsmstat07	0.0062	0.869	0.059	0.046	0.053	0.0039	±0.007	0.016	0.040	0.024
vsmstat	0.0063	0.911	0.059	0.048	0.054	0.0043	±0.007	0.017	0.045	0.027
lsi150stat	0.0064	1.000	0.060	0.050	0.055	0.0025	±0.002	0.009	0.086	0.041
sabmq08a1	0.0094	1.000	0.071	0.079	0.080	0.0069	±0.012	0.012	0.034	0.038
lsi150dyn	0.0190	1.000	0.091	0.138	0.127	0.0342	±0.017	0.067	0.135	0.136
000cos	0.0236	0.529	0.104	0.127	0.138	0.0533	±0.009	0.114	0.094	0.155
vsmdyn	0.0247	1.000	0.103	0.163	0.146	0.0493	±0.018	0.093	0.136	0.148
000tfidfLOG	0.0322	1.000	0.115	0.206	0.175	0.0896	±0.016	0.166	0.309	0.236
000tfidfBM25	0.0373	1.000	0.122	0.222	0.189	0.1106	±0.014	0.194	0.294	0.245
000klabs	0.0428	1.000	0.130	0.227	0.202	0.1259	±0.015	0.201	0.249	0.264
000okapi	0.0441	1.000	0.133	0.243	0.212	0.1443	±0.016	0.219	0.281	0.287
sabmq08b1	0.0487	0.994	0.133	0.282	0.228	0.1398	±0.020	0.202	0.391	0.314
LucDeflt	0.0492	1.000	0.143	0.255	0.221	0.1739	±0.015	0.244	0.331	0.278
indriLowMu08	0.0548	0.997	0.146	0.280	0.241	0.1861	±0.013	0.271	0.300	0.290
mpiimq0801	0.0549	1.000	0.139	0.316	0.248	0.1855	±0.020	0.245	0.384	0.355
neuMSRF	0.0583	1.000	0.146	0.328	0.263	0.1925	±0.019	0.261	0.440	0.398
neustbl	0.0717	1.000	0.161	0.359	0.289	0.2268	±0.020	0.302	0.417	0.367
neumsfilt	0.0746	1.000	0.161	0.385	0.303	0.2266	±0.023	0.316	0.444	0.402
dxrun	0.0798	0.547	0.170	0.373	0.309	0.2744	±0.019	0.366	0.399	0.398
txrun	0.0802	0.560	0.169	0.382	0.310	0.2854	±0.020	0.373	0.454	0.406
indriQLST08	0.0803	0.951	0.170	0.379	0.309	0.2716	±0.019	0.349	0.431	0.395
ind25QLnST08	0.0812	0.583	0.171	0.389	0.311	0.2810	±0.020	0.355	0.448	0.388
LucLpTfS	0.0823	1.000	0.172	0.407	0.322	0.2815	±0.022	0.359	0.506	0.408
indri25DM08	0.0849	NA	0.174	0.398	0.325	0.3114	±0.021	0.390	0.475	0.403

Table 2. MTC and statAP estimation of IR measures. The numbers are weighted averages over queries, ordered by MTC estimate of wMAP (second column). MTC confidence (3rd column): the confidence that the system performance is worse than the one of the next system; statAP confidence (8th column): the 95% confidence interval.

extent to which the number of queries and judgments can be reduced, and how to sample the categories, to achieve similar results with less overall effort.

Our aim is to answer two questions: (1) what is the number of queries needed for different levels of relevance incompleteness to guarantee that, when systems are run over this many queries, their MAP scores reflect their actual performance, and (2) given a fixed budget of total number of judgements (or total hours of labor), what is the ratio between number of queries and number of judgements per query that maximizes stability?

4.1 Analysis of Variance Studies

When systems are run over different sets of queries, MAP scores (and consequently the ranking of these systems) may vary. The amount of variability that occurs in MAP scores (as measured by variance) across all sets of queries and all systems can be decomposed into three components: (a) variance due to actual performance differences among systems — *system variance*, (b) variance due to the relative difficulty of a particular set of queries — *query set variance*, and (c) variance due to the fact that different systems consider different set of queries hard (or easy) — *system-query set interaction variance*. The variance due to

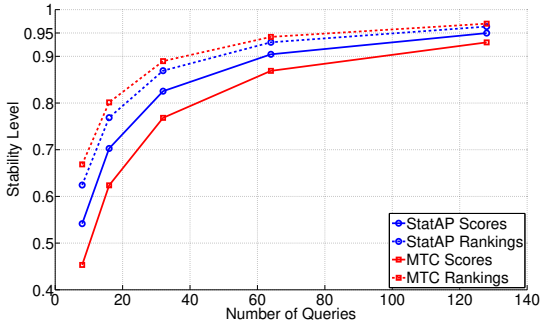

Fig. 2. Stability level of MAPs and induced ranking for statAP and MTC as a function of the number of queries.

other sources of variability, such as sampling of documents, is confounded with the later variance component, i.e. variance due to system-query set interaction.

Ideally, one would like the total variance in MAP scores to be due to the actual performance differences between systems, as opposed to the other two sources of variance. The percentage of variance attributed to the *system* effect is a function of the size of the query set.

Stability of MAP and induced rankings We ran two separate variance decomposition studies for the MAP estimates produced by each method. In both cases systems were run over the same set of 784 queries.[1]

The variance in wMAP is a function of the variance of the MAP within each query class and the covariance of the MAPs among query classes. Thus, instead of fitting a single ANOVA model in APs over all queries [13, 14], we used a Multivariate Analysis of Variance (MANOVA) [15]. The variance of MAP within each query class was decomposed into the aforementioned variance component, while the covariance of the MAPs among the query classes was solely attributed to *system* effects, since the query classes are disjoint.

For both studies, we report (a) the stability levels of the MAPs (system variance over total variance) and (b) the stability levels of the systems rankings (system variance over system and system/query interaction), both as a function of the total number of queries in the query set. Figure 2 shows the results: the solid lines correspond to stability levels of MAPs while the dashed lines correspond to stability levels of system rankings. As the figure indicates, statAP reaches a MAP stability level of 0.95 with a set of 129 queries, while MTC reaches the same level with 204 queries (not observed in the figure).[2] MTC reaches a ranking stability level of 0.95 with a set of 83 queries, while statAP reaches the same level with 102 queries.

[1] Note that statAP does not report scores for queries with no relevant document found; studies for statAP are on the 564 queries for which statAP returned scores.

[2] We have observed in our experiments that a stability of 0.95 leads to a Kendall's tau of approximately 0.9.

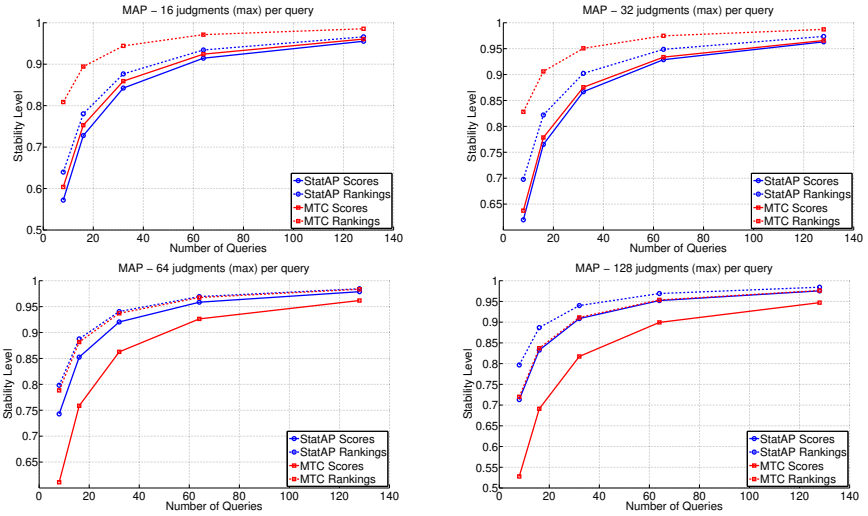

Fig. 3. Stability levels of MAP scores and induced ranking for statAP and MTC as a function of the number of queries for different levels of relevance incompleteness.

These results support the claims that the statAP method, by design, aims to estimate the actual MAP scores of the systems, while the MTC method, by design, aims to infer the proper ranking of systems.

Stability with incomplete judgments To illustrate how stable the MAPs returned by the two methods are with respect to different levels of relevance incompleteness, we ran ANOVA studies for the each one of the query classes separately. Figure 3 demonstrate the stability levels for both methods when 16, 32, 64, and 128 judgments are available, respectively. MTC leads to both more stable MAPs and induced rankings than statAP when 16 or 32 relevance judgments are available per query, while the opposite is true when 64 or 128 relevance judgments are available.

Note that, the stability of the MAP scores returned by statAP degrades with the relevance incompleteness, as expected. On the other hand, the opposite is true for MTC. For the estimation of MAP scores, MTC is employing a prior distribution of relevance which is calculated by combining information from all queries, which violates the query independence assumption ANOVA makes. The fewer the relevance judgments, the larger the weight on the prior distribution, and thus the more the assumption is violated. Consequently, MAP scores seem to be more stable than they should.

Stability for query categories Here we consider how sampling queries in different proportions can affect the stability of measures and induced rankings. For each pair of categories (short/long and govheavy/govslant), we fit a MANOVA model to the MAPs and calculated the optimal ratio of queries to be subsampled from each category. Representative results from these studies are illustrated in Figure 4. In both plots, we gradually change the ratio of "govheavy" to "govs-

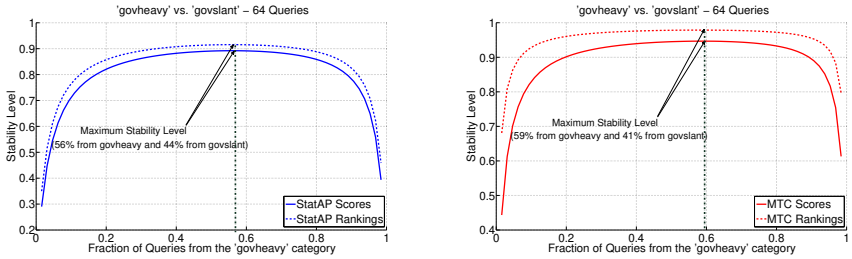

Fig. 4. Stability of MAPs and induced rankings returned by statAP and MTC as a function of different subsampling ratios for "govheavy" and "govslant", given a fixed sample of 64 queries.

lant" queries in a sample of 64 and plotted the stability level achieved for each ratio. Both statAP and MTC demonstrate a slight preference towards "govheavy" queries. In particular, selecting $55-60\%$ "govheavy" results in the optimal stability level for both scores and rankings. The results for "short" and "long" queries indicated no particular preference for one or the other.

4.2 Cost-Benefit Analysis

We measured the elapsed time assessors spent performing various interactions with the judging system. From these we can construct a cost function which we can then use to identify an optimal operating point: how many and what type of queries and judgments are needed to reach a point where the ranking would not be expected to change much with additional judgments or queries.

Assessor Time Assessor spent the majority of their time on three activities: (1) selecting a query; (2) backfitting the query to a topic definition; and (3) judging documents. Selecting a query can further be broken down into the following: the number of times the assessor refreshed the display (to see a new set of 10 queries), the time the assessor spent looking at each list of 10, and the time the assessor spent looking at the last list of 10 just before selecting one for topic definition. These numbers are shown in Table 3 along with the time spent on topic definition. Note that all numbers are median seconds—the mean is inappropriate because the distribution is heavily skewed by long breaks. Note also that time varied with query type: in particular, choosing queries and defining topics took quite a bit longer with long queries than with short.

Time to make judgments is also shown in Table 3. Here too time varied by query type, and by the target number of judgments. The fact that each judgment was made faster when more were requested suggests that assessors have some "ramp-up" time on a topic, after which they can make judgments faster.

Cost Given a query q of category c with target number of judgments j, the total time spent on that query is $((nr_c-1)tl_c+tf_c+tt_c+tj_{cj}j$ where nr_c is the number of refreshes for query type c, tl_c is the time spent looking at a list of 10 queries for query type c, tf_c is the time spent looking at the final list of 10 for

					judgment times					
category	refresh	view	lastview	topic	8	16	32	64	128	average
short	2.34	18.0	25.5	67.6	15.0	11.5	13.5	12.0	8.5	12.5
long	2.54	24.5	31.0	86.5	17.0	14.0	16.5	10.0	10.5	13.0
slant	2.22	22.5	29.0	76.0	13.0	12.5	13.0	9.5	10.5	12.0
heavy	2.65	20.0	27.5	78.0	19.0	13.0	17.0	12.5	8.5	13.5
average	2.41	22.0	29.0	76.0	15.0	13.0	15.0	11.0	9.0	13.0

Table 3. Average number of query lists viewed and median seconds spent viewing each list, viewing the final list, and defining the topic for the selected query.

query type c, tt_c is the time spent on topic definition for type c, and tj_{cj} is the time spent on a judgment for query type c with target j.

Then the total cost of Q queries of which q_{c_i} are from category i ($1 \leq i \leq k$) is $cost = \sum_{i=1}^{k} q_{c_i} \left((nr_{c_i} - 1)\, tl_{c_i} + tf_{c_i} + tt_{c_i} + t_{c_i j} j \right)$, assuming every query has the same target j. This cost function can accomodate arbitrary splits between query categories, and even take into account differing target numbers of judgments. When doing so one should take into account the variance in the time estimates. If that variance is high, it may be prudent to average categories together and not distinguish between them in the cost analysis.

Figure 5 shows the Kendall's τ correlation between the baseline ranking by weighted MAP and the ranking over a subset of queries (all with the same target number of judgments) versus the total cost of judging that set of queries. In this plot we used average times only; the cost is not based on query category. Note that τ increases logarithmically with total cost (as the fitted lines show clearly). The fit lines suggest that a little over 15 hours of assessor time are needed in order to reach an expected τ of 0.9. This corresponds to 100 queries with 32 judgments each. A mean τ of 0.9 is first reached after only 9 hours of assessor effort. However, the standard deviation of τs is fairly high; any given sample of topics could produce a τ anywhere between 0.84 and 0.96. To ensure a τ of 0.9 with 95% probability requires around 15 hours of assessor time.

The right plot in Figure 5 shows the minimum time required by each method to reach an expected Kendall's τ of 0.9 as the number of judgments per query increases. When there are few judgments per query, many queries are needed; a long time is required. As the number of judgments increases, the number of queries needed decreases, and less time is required. There is a tradeoff, though, as the time to make judgments begins to exceed the time spent on query selection, and the total time begins to rise again. The minimum time for both methods is achieved at 64 judgments.

Using the times in Table 3 and the full cost function, we may consider whether a τ of 0.9 could be reached with less effort when query types are sampled in different proportions. Figure 6 shows the tradeoff between cost and Kendall's τ when short queries are sampled 25%, 50%, and 75%, and the tradeoff when "govheavy" is sampled 25%, 50%, and 75%. Note that the empirical data to generate different splits is limited; since for example there are only 50 queries

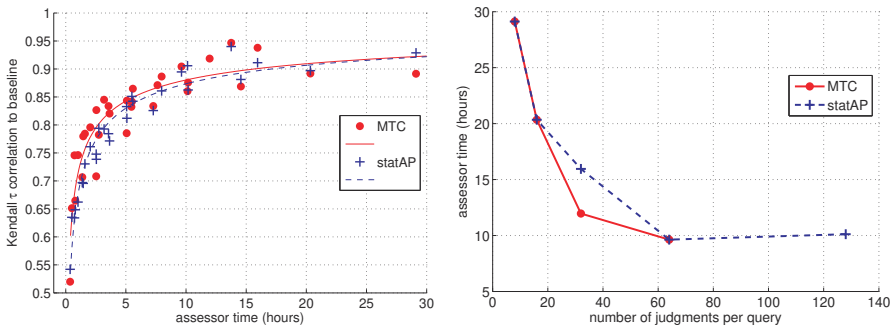

Fig. 5. Cost plots: on the left, total assessor time to reach a Kendall's τ rank correlation of 0.9 with the baseline; on the right, minimum time needed to reach a τ of 0.9 with increasing numbers of judgments per query.

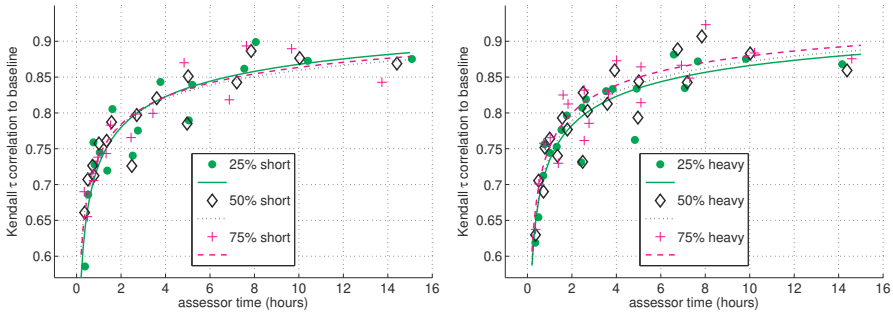

Fig. 6. Cost plots. On the left, sampling short versus long in different proportions. On the right, sampling heavy versus slant in different proportions.

with 64 judgments each, and they are roughly evenly split between categories, we cannot measure a τ with 75% of one category with 64 judgments.

The left plot suggests a slight (but not significant) preference for sampling long queries, while the right suggests a preference for heavy queries. Combining the two in such a way as to guarantee 75% long queries and 75% heavy queries (i.e. sampling long-heavy $0.75 \times 0.75 = 0.56$, long-slant 0.19, short-heavy 0.19, and short-slant 0.06), an expected Kendall's τ of 0.9 is reached after a little over 4 hours of assessor effort.

5 Conclusion

We put in practice two recently developed evaluation techniques that, unlike standard evaluation, scale easily and allow many more experiments and analyses. We experimented with 25 submitted systems over 10000 natural queries, evaluating 784 of them with only about 15000 judgments.

We investigated system performance over pre-assigned query categories. There is some evidence that over-sampling some types of queries may result in cheaper (if not substantially more efficient) evaluation: over-sampling long and govheavy queries resulted in a good ranking with just a little over four hours of simulated assessor time. More investigation to find the right tradeoffs is a clear direction for future work.

Cost analysis; optimal budget. Queries were randomly assigned 5 different target numbers of judgments such that the total number of judgments for each class is roughly constant. This split facilitated a derivation of optimal budgeting for IR evaluation via cost (in assessor hours) analysis. We concluded that 30-60 judgments per query with around 100 queries is optimal for assessing systems' performance ranking.

Evaluation stability. The setup also allowed an analysis of evaluation stability with fewer judgments or queries. Using ANOVA, we concluded that MTC needs about 83 queries with approximately 1700 total judgements for a reliable ranking, while statAP needs about 129 queries with approximately 2650 total judgements for a reliable estimate of MAP.

Acknolwedgments This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, in part by the National Science Foundation (NSF) under grant numbers IIS-0533625 and IIS-0534482, and in part by Microsoft Research. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

References

1. Sakai, T.: Alternatives to bpref. In: Proceedings of SIGIR, ACM (2007) 71–78
2. Carterette, B., Allan, J., Sitaraman, R.K.: Minimal test collections for retrieval evaluation. In: Proceedings of SIGIR. (2006) 268–275
3. Moffat, A., Webber, W., Zobel, J.: Strategic system comparisons via targeted relevance judgments. In: Proceedings of SIGIR, ACM (2007) 375–382
4. Aslam, J.A., Pavlu, V., Yilmaz, E.: A statistical method for system evaluation using incomplete judgments. In: Proceedings of SIGIR. (2006) 541–548
5. Allan, J., Aslam, J.A., Carterette, B., Pavlu, V., Kanoulas, E.: Overview of the trec 2008 million query track. In: Notebook Proceedings of TREC. (2008)
6. Carterette, B., Pavlu, V., Kanoulas, E., Allan, J., Aslam, J.A.: Evaluation over thousands of queries. In: Proceedings of SIGIR. (2008) 651–658
7. Allan, J., Carterette, B., Aslam, J.A., Pavlu, V., Dachev, B., Kanoulas, E.: Overview of the TREC 2007 Million Query Track. In: Proceedings of TREC. (2007)
8. Yilmaz, E., Aslam, J.A.: Estimating average precision with incomplete and imperfect judgments. In: Proceedings of CIKM. (2006) 102–111
9. Aslam, J.A., Pavlu, V.: A practical sampling strategy for efficient retrieval evaluation, technical report.
10. Brewer, K.R.W., Hanif, M.: Sampling With Unequal Probabilities. Springer, New York (1983)
11. Stevens, W.L.: Sampling without replacement with probability proportional to size. Journal of the Royal Statistical Society. Series B (Methodological), Vol. 20, No. 2. (1958), pp. 393-397
12. Thompson, S.K.: Sampling. Wiley Series in Probability and Mathematical Statistics (1992)
13. Banks, D., Over, P., Zhang, N.F.: Blind men and elephants: Six approaches to trec data. Inf. Retr. **1**(1-2) (1999) 7–34
14. Bodoff, D., Li, P.: Test theory for assessing ir test collection. In: Proceedings of SIGIR. (2007) 367–374
15. Brennan, R.L.: Generalizability Theory. Springer-Verlag, New York (2001)