Measuring the Reusability of Test Collections

Ben Carterette[†], Evgeniy Gabrilovich[‡], Vanja Josifovski[‡], Donald Metzler[‡]

[†] Department of Computer & Information Sciences, University of Delaware, Newark, DE
[‡] Yahoo! Research, 2821 Mission College Blvd, Santa Clara, CA
carteret@cis.udel.edu | {gabr | vanjaj | metzler }@yahoo-inc.com

ABSTRACT

While test collection construction is a time-consuming and expensive process, the true cost is amortized by reusing the collection over hundreds or thousands of experiments. Some of these experiments may involve systems that retrieve documents not judged during the initial construction phase, and some of these systems may be "hard" to evaluate: depending on which judgments are missing and which judged documents were retrieved, the experimenter's confidence in an evaluation could potentially be very low. We propose two methods for quantifying the reusability of a test collection for evaluating new systems. The proposed methods provide simple yet highly effective tests for determining whether an existing set of judgments is useful for evaluating a new system. Empirical evaluations using TREC datasets confirm the usefulness of our proposed reusability measures. In particular, we show that our methods can reliably estimate confidence intervals that are indicative of collection reusability.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

evaluation, test collections, reusability

1. INTRODUCTION

Test collections lie at the heart of IR evaluation; they foster reproducible results and allow principled comparison of multiple retrieval systems. Test collections typically consist of a set of queries, a set of documents, and a set of relevance judgments. In some test collections, the judgments cover all possible query-document pairs. For instance, text categorization test collections normally provide an exhaustive list of categories each document belongs to. However, this is not the case in most retrieval tasks, notably for those where the set of documents is so large that it is simply not feasible to judge every document for every query in the set.

The *pooling method* provides a way to focus judging effort on those documents least likely to be non-relevant [12]. Given a set of systems to be evaluated over the queries in the test collection, the top-scoring documents retrieved by the systems are pooled and judged for relevance to the queries that retrieved them. In test collections of realistic size, it is unlikely that pooling will find all the relevant documents in the corpus, but identifying and judging all such documents would be prohibitively expensive. When new systems are subsequently evaluated using the same test collection, practitioners are faced with one of the following two choices. One option is to collect judgments for the documents retrieved that were not previously judged. This can be costly and time consuming, especially when many new systems must be tested over a large test collection, as is the case for Web search. The other option is to only use existing judgments and effectively ignore newly retrieved documents that have not been previously judged. Evaluation can then be done either using compressed ranked lists [10], or by using evaluation metrics that can handle missing judgments [1, 3]. Depending on the number of queries and the number of unjudged documents retrieved, this approach may lead to a highly inaccurate measure of the system's true performance.

We propose methods for quantifying the suitability of an existing set of judgments for evaluating new systems. Specifically, we show how estimates of *confidence intervals* for evaluation metrics such as precision or mean average precision (MAP) over the space of *possible* judgments for unjudged documents. The widths of these intervals provide clues as to the suitability of existing judgments to evaluate the new system. We also propose point estimates of reusability based on standard evaluation metrics and show that despite being less informative than the full confidence interval, they can provide quick and easy estimates of the interval width.

The main contributions of this paper are threefold. First, we introduce the concept of *reusability estimation*, which aims to quantify how useful a set of existing relevance judgments is for evaluating a new system. To the best of our knowledge, principled estimates of reusability of test collections have not been previously studied. Second, we propose two novel methods for quantifying reusability. The first method constructs confidence intervals for evaluation metrics using logistic regression, while the second converts standard information retrieval metrics into reuse metrics. Fi-

WSDM'10, February 4–6, 2010, New York City, New York, USA.

nally, we report the results of experimental evaluation using several TREC collections, which confirm that our methodology provides a reliable way for quantifying reusability and predicting system performance.

The remainder of the paper is organized as follows. Section 2 describes previous work related to evaluation and reusability. Section 3 presents our methodology for measuring the reusability of a test collection with respect to a new system. In Section 4 we evaluate our proposed methodology using TREC data. Section 5 concludes the paper and discusses possible directions for future work.

2. RELATED WORK

Reusability has long been a central concern about test collections. The original IR test collection, the Cranfield set of aerodynamic engineering abstracts, research questions, and relevance judgments, came to be reused to study retrieval tasks far beyond what it was designed for simply because it was easier to obtain it than to build a new collection. When TREC (the Text REtrieval Conference) began in 1992, part of its mission was to provide the wider research community with large test collections devoted to particular tasks that would be reusable (or "portable") across research groups and over time [14]. There can be no doubt that it succeeded at that. However, in recent years, test collections have begun to become so large that their reusability is unclear. Pools tend to contain documents that exhibit certain properties such as containing a high proportion of title query terms, and many relevant documents that do not have those properties are left unjudged [2]. Reusing such collections will tend to favor systems that are like those that contributed the original documents.

Zobel was among the first to question the completeness of the pooling method and the effect of missing relevant documents on system evaluations [16]. He found that pooling could miss up to 50% of the relevant documents in the corpus, but relative orderings of systems would not be seriously affected. However, more recent collections seem to have serious bias problems [2, 4].

There are two general approaches to cope with the bias due to missing judgments. One is to modify existing or introduce new evaluation metrics. The *bpref* metric was introduced to deal with missing judgments by counting the number of non-relevant documents ranked above relevant documents [3]. Inferred average precision (infAP), as its name suggests, uses inferred precision values when judgments are missing [15]. Sakai introduced a suite of metrics based on compressing the ranked list to eliminate unjudged documents [10].

An alternative approach involves attempting to predict the relevance of unjudged documents for use in standard or new evaluation metrics. Jensen et al. addressed experimental repeatability, which is related to reusability) using judgments inferred from manually-built taxonomies [9]. Carterette addressed reusability in terms of whether two new systems could be reliably ranked relative to each other using relevance predictions based on very small sets of judgments. The predictions are used to calculate a probability that two systems are likely to swap after additional judgments [5]. Büttcher et al. used an SVM to predict the relevance of unjudged documents to find likely new relevant documents [4]. Although such estimates will have errors, it is unclear that they are any noisier than the judgments themselves.

This work is about assessing whether a particular system that did not contribute to a pool can be accurately evaluated given the judgments in that pool. We use both of the approaches above: we estimate the relevance of unjudged documents and use those estimates to calculate expectations for standard retrieval metrics. On top of that we add a *confidence interval* calculated using the predictions of relevance (note the difference from Carterette's previous work, which only estimated swap probabilities [5]). When confidence intervals are wide and overlapping with confidence intervals for other systems, it is a red flag that more judgments are required before any reliable conclusion about the systems may be made. Because these confidence intervals are difficult to compute, we also introduce much simpler point estimates of their width, using a simple linear combination of features.

3. MEASURING REUSABILITY

The judgments from an existing test collection are often used to measure the *performance*, or effectiveness, of a new system. Classical information retrieval metrics include precision, recall, F1, mean average precision, R-precision, and DCG. These metrics assume that the relevance judgments are complete, that is, they require every document retrieved for every query to be judged, or else the metrics are undefined. That does not mean they are not useful, of course; depending on the experimental setting and assumptions made, metrics calculated without knowing all judgments can impart a great deal of information. Nevertheless, when using past judgments to evaluate new systems with classical metrics, problems can arise when unjudged documents are retrieved. In this case the judgments are said to be incomplete. In practice, nearly all collections are incomplete, so dealing with missing judgments is very important.

As discussed above, there are several ways to deal with unjudged documents. Such documents can be treated as non-relevant, based on the assumption that most documents are indeed non-relevant to any given query—a problematic assumption for recent large collections. Instead of assuming they are non-relevant, then, they can be ignored by forming condensed ranked lists [10]. However, recently it was shown that evaluations based on condensed ranked lists are biased when judgments are collected by pooling [11]. Several metrics have been proposed that overcome the problem of missing judgments by inferring the relevance of such items [1, 4, 7], but these approaches fail to quantify how accurate such evaluations actually are in the presence of missing data.

We propose a set of *reusability measures* that quantify the confidence that the existing test collection can be used to accurately evaluate the performance of a new system. Such measures are of theoretical and practical importance. The theory behind the measures can be used to develop more robust evaluation metrics. From a practical side, the measures can be used by IR practitioners to determine whether or not their existing test collection is sufficient to evaluate a new system, or if new judgments are needed.

We propose two types of reusability measures, each with their strengths and weaknesses, as we will describe shortly. Measures of the first type estimate a confidence interval for the metric of interest, such as mean average precision, by inferring the relevance of unjudged documents within a logistic regression framework. Measures of the second type compute a single scalar value that is distilled from classical and newly proposed evaluation metrics.

3.1 Interval Estimates of Reusability

Our first measure of reusability comes in the form of confidence intervals. More formally, suppose we have an existing set of judgments $\mathcal{J}$ over the set of queries $\mathcal{Q}$ and we wish to evaluate a new system on $\mathcal{Q}$ (or some subset of $\mathcal{Q}$) according to metric m. Our goal is to estimate a confidence interval for m given $\mathcal{J}$ and the ranked list of documents retrieved by the new system.

If the estimated confidence interval is wide, then we can say that $\mathcal{J}$ is non-reusable. However, if the confidence interval is within some acceptable tolerance, as dictated by the underlying task, then we say that $\mathcal{J}$ is reusable.

Confidence intervals are rather powerful in this situation, as they allow the practitioner to determine an acceptable level of uncertainty in their estimate. The uncertainty in the confidence intervals comes from unjudged documents being retrieved by the new system.

One can see, from a mathematical perspective, how such variance arises for common retrieval metrics. Carterette showed that the mean and variance for precision at k and average precision have analytical forms [6]. Given a query $Q \in \mathcal{Q}$, these analytical forms are:

$$\mathrm{E}[prec@k] = \frac{1}{k}\sum_i p_i I(\mathrm{A}_i \leq k)$$

$$\mathrm{Var}[prec@k] = \frac{1}{k^2}\sum_i p_i q_i I(\mathrm{A}_i \leq k)$$

$$\mathrm{E}[AP] \approx \frac{1}{\sum_i p_i}\left(\sum_i a_{ii}p_i + \sum_{i,j} a_{ij}p_i p_j\right)$$

$$\begin{aligned}\mathrm{Var}[AP] \approx\ & \frac{1}{(\sum_i p_i)^2}\Big(\sum_i a_{ii}p_i q_i + \sum_{i,j} a_{ij}p_i p_j(1-p_i p_j) \\ & + \sum_{i,j} 2a_{ii}a_{i,j}p_i p_j q_i + \sum_{i,j,k} 2a_{ij}a_{ik}p_i p_j p_k q_i\Big)\end{aligned}$$

where the indexes i, j, and k go over the set of documents retrieved for Q, p_i is the probability that document i is relevant, $q_i = 1 - p_i$ is the probability that document i is non-relevant, A_i is the rank of document i, $I(\mathrm{A}_i \leq q) = 1$ if the inequality is true and 0 otherwise, and $a_{ij} = 1/\max\{A_i, A_j\}$.

3.1.1 Modeling document relevance probability (p_i)

If document i is judged relevant, then $p_i = 1$; if it is judged non-relevant, then $p_i = 0$. There are several options for estimating p_i if document i is unjudged. The most naïve is to let $p_i = 0$ or $p_i = 0.5$ for unjudged documents. However, these estimates are unlikely to be accurate and may lead to poor estimates of the mean and variance. An alternative, which we adopt here, is to estimate p_i using a statistical model. We choose to model p_i using logistic regression, which is commonly used to model binary responses. Under this model, estimates for p_i have the form:

$$p_i = \frac{1}{1 + \exp[-\theta^T F(Q_i, D, \mathcal{J})]}$$

where θ is the model parameter vector and $F(Q_i, D, \mathcal{J})$ is a vector of features extracted for some query Q_i, document D, and set of relevance of judgments $\mathcal{J}$.

The model is trained as follows. Given an existing test collection, we first extract feature vectors for every (query, judged document) pair. The target, or response, associated with each pair is the judgment, with non-relevant and relevant judgments corresponding to targets of 0 and 1, respectively. Finally, the model parameter vector θ is estimated using maximum likelihood. The trained model can then be used to estimate p_i for unjudged documents retrieved by a new system, thereby allowing us to effectively estimate the mean and variance of the metric under consideration.

3.1.2 Features

We explore two types of features in this work. The first type are *document similarity features*, which were originally proposed by Carterette and Allan [7]. For a given document i, we compute the cosine similarity between i and all of the judged documents. The general motivation behind these features is that if a given unjudged document is similar to one or more of the relevant documents, then the document itself is likely to be relevant. This is related to the well-known cluster hypothesis [13]. A similar argument can be made for non-relevant documents, as well.

Features of the second type are so-called *system features*, which quantify the effectiveness of a system and how complete the existing judgments are for the system. For every (query, document) pair we compute the following system features with respect to the existing judgments: the rank of the document, precision for known relevant documents at that rank, expected precision at that rank, and mean average reuse, which is a measure we will describe in more detail in Section 3.2. Each unique document may be associated with multpile feature vectors by virtue of having been ranked by more than one system. In these cases, the final probability of relevance p_i is obtained by averaging the values predicted by its feature vectors.

In addition to features that depend on both the query and the document, we also extract the following query-level features: fraction of relevant results retrieved, fraction of non-relevant results retrieved, fraction of unjudged results retrieved, and the mean average reuse of the query. Finally, for each query-level feature, we produce a system-level feature that is the mean of the query-level features computed over the entire set of queries.

Although we only consider these two types of features here, it is easy to include additional features, such as domain- or task-specific features, within the model. Additional features may improve the quality of the model estimates.

3.1.3 Confidence Intervals

Given a set of queries, it is common to report the mean of some metric over the entire set (e.g., mean average precision). We denote the mean of metric m by $\overline{m}$. Under the assumption that metrics are independent across queries, we compute the mean and variance of $\overline{m}$ as follows:

$$\mathrm{E}[\overline{m}] = \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}[m(Q_i)], \mathrm{Var}[\overline{m}] = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}[m(Q_i)]$$

where $m(Q_i)$ is metric m evaluated on query Q_i.

It is straightforward to estimate confidence intervals for $\overline{m}$ now that we have estimates for its mean and variance. The $100(1-\alpha)\%$ confidence interval for $\overline{m}$ is computed as follows:

$$\left[\mathrm{E}[\overline{m}] - z_{\frac{\alpha}{2}}\sqrt{\frac{\mathrm{Var}[\overline{m}]}{n}}, \mathrm{E}[\overline{m}] + z_{\frac{\alpha}{2}}\sqrt{\frac{\mathrm{Var}[\overline{m}]}{n}}\right]$$

where n is the number of queries and $z_{\frac{\alpha}{2}}$ is the value of z that satisfies $P(Z \leq z) = 1 - \frac{\alpha}{2}$, where Z is distributed according to a standard normal distribution. We note that these confidence intervals are only valid if $\overline{m}$ is normally distributed, which is generally true by the Central Limit Theorem, assuming a large enough sample of queries. Studentized intervals can be used, if necessary (e.g., for $n < 30$).

Based on this formulation, it is easy to see that the two primary ways to tighten the bounds of the confidence interval are to lower the variance of the metric (i.e., obtain relevance judgments for unjudged documents) or increase the number of queries.

Although we primarily focus on precision at k and average precision in this paper, it should be noted that analytical forms for the means and variances of other retrieval metrics exist, including recall and NDCG [6]. Thus, our interval-based reusability measures can be easily applied to these metrics, as well. In general, our approach can be applied to any IR metric, including those that are not normally distributed and those that do not have analytical forms for their mean and variance. For such metrics, it may be possible to use bootstrap methods to estimate confidence intervals [8].

3.2 Point Estimates of Reusability

Confidence intervals are useful because they estimate the entire range of possible values of evaluation metrics for a new system based on the existing judgments. In general, interval estimates are more expressive and useful than point estimates. However, point estimates can be useful, not only because they provide a single number summary, but also because they are typically easier to compute. As we just showed, estimating confidence intervals can be somewhat involved, as it requires extracting features, estimating model parameters, computing means and variances, and so on. Therefore, we would like to develop point measures that can be used as proxies for confidence intervals. An ideal point measure for reusability would correlate strongly with the width of the estimated confidence intervals.

To compute point estimates of retrieval metrics, we define a set of novel features that directly quantify collection reusability. Specifically, we propose a methodology for converting standard precision-based evaluation metrics, such as precision at rank k and mean average precision, into reusability measures. Traditionally, the concept of precision has been used in information retrieval evaluation to determine the distribution of relevant documents in a set of retrieved documents. We can use a similar approach to define the reusability measure called *reuse* as the proportion of the *judged* documents that are retrieved by the new version of the system. Reuse at rank k for query Q is defined as:

$$reuse@k(Q) = \frac{judged@k(Q)}{k}$$

where $judged@k(Q)$ is the number of judged documents in the top k results for query Q using the new system.

While $reuse@k(Q)$ provides a measure that indicates the reusability of the judgments of the new version of the system, it suffers from many of the same problems as precision in standard performance evaluation, as it is not rank-aware. Various precision-based evaluation metrics are rank-aware, including average precision. We can easily convert average precision into a reusability measure, which we call *average reuse* (AR), as follows:

$$AR(Q) = \frac{1}{judged(Q)} \sum_i reuse@i(Q)$$

where $judged(Q)$ is the number of documents judged for Q and i ranges over the judged document positions. We define the *mean average reuse* (MAR) as the mean of the AR values computed over a set of queries.

It should now be clear that any precision-based metric that uses binary relevance judgments can be easily converted into a reusability measure by assuming that judged documents are "relevant" (positive) and unjudged documents are "non-relevant" (negative).

To wit, our proposed reuse and average reuse measures ignore whether or not a retrieved document is relevant or not. The measures simply account for whether or not the document is judged. However, knowing whether the documents retrieved are relevant or not may be indicative of reusability. For example, if a system fails to retrieve many judged relevant documents, then we can assert with high confidence that the new system is bad. The opposite is not always true, however. A system that returns many judged relevant documents is not necessarily good, because its unjudged documents may actually be non-relevant. It depends on how many of its retrieved documents remain unjudged.

This is somewhat counterintuitive, but it follows from the fact that the proportion of relevant documents is very low. Most of the documents systems retrieve are non-relevant, and thus when a system fails to retrieve the relevant documents we know about, it is very unlikely that it retrieved many that we do not know about. There are exceptions, of course, but on average this is true. A system that retrieves many known relevant documents but still has many unjudged documents could go either way. It has established itself as being good at finding relevant documents, so there is reason to believe many of its unjudged documents are relevant. On the other hand, we know *a priori* that most unjudged documents are non-relevant. These conflicting states of knowledge produce low confidence in the system's performance.

Therefore, we propose using traditional retrieval metrics calculated over judged relevant documents as point estimates of reusability as well. These include recall, precision at k, and MAP. We hypothesize that a combination of relevance-unaware measures like mean average reuse and relevance-aware measures like recall are good proxies for full-blown confidence intervals. We test this hypothesis in Section 4 by measuring the correlation between these two measures and the widths of confidence intervals estimated using the procedure described in Section 3.1.

4. EXPERIMENTAL EVALUATION

In this section we present experimental results demonstrating our ability to estimate and predict confidence intervals for different evaluation metrics and tasks. We show that a surprisingly small set of judgments is needed to rank new systems accurately *and* with high confidence, as long as those judgments came from a diverse set of systems.

Broadly speaking, our experimental procedure is to simulate three experiments. In the first experiment, a small set of runs contribute documents to a pool that is judged, and the pool is then used to evaluate those systems. In the second

track	sites	runs	topics	judgments	rel
Web 2004	18	74	225	88,566	1,763
Robust 2005	17	74	50	37,798	6,561

Table 1: Statistics of runs submitted to the TREC 2004 Web and 2005 Robust tracks.

experiment, a new set of systems needs to be evaluated; we use the pool from the first experiment to evaluate them by expected MAP or precision with confidence intervals, and use the width of the confidence intervals and their degree of overlap to determine which systems need more judgments. The third experiment is similar to the second, except instead of computing expectations and confidence intervals, we use simple point estimators to try to predict the width of the confidence intervals.

4.1 Data

Simulating these experiments can be done with TREC data: we obtained the runs submitted by various sites (universities and companies) to two different TREC tracks, and used the TREC *qrels* to simulate pooling and judging. To ensure that our results hold for different tasks, we used runs submitted to the Web track in 2004 and the Robust track in 2005. These two tracks have topics representing four different tasks: ad hoc retrieval, topic distillation, home page finding, and named page finding. They are recent enough to reflect current trends.

Table 1 shows some statistics of the data. Both tracks had 74 submissions from roughly the same number of sites. The Web track runs are over 225 topics, of which 75 are topic distillation, 75 are home page finding, and 75 are named page finding. The Robust track runs are over 50 of the hardest topics drawn from earlier ad hoc tracks. The topics have been judged fairly extensively, with about 393 judgments per Web topic and 756 per Robust topic. The Web topics have many fewer relevant documents, reflecting the types of tasks they cover.

4.2 Methodology

The goal is to determine whether a new system can be accurately evaluated given a pool of judgments constructed from other systems. Any system can be evaluated, of course; the question is the degree of confidence we have in that evaluation. As explained above, we will simulate three experiments. Here we describe the methodology in more detail.

To maximize the difference between systems in the first experiment (those that contribute to the pool) and systems in the second and third experiments (those that are evaluated using that pool), we partitioned runs by the site that submitted them. Previous work has shown that this is the best way to ensure that the second and third groups are maximally uncoupled from the first for most robust experimentation. In each experiment we select one or more sites at random and use the runs submitted by that site to either collect judgments or to evaluate. The full experiment is as follows:

1. Pick m_1 sites randomly. We will refer to the runs submitted by these sites as *training* runs.
2. Form a pool from the top k documents retrieved for each query by each of the training runs.
3. Pick m_2 sites randomly. We will refer to the runs submitted by these sites as *validation* runs.
4. Estimate probabilities of relevance of unjudged documents retrieved by both training and validation runs using features extracted from both groups of runs.
5. Calculate expectations and variances of MAP and precision for each validation run.
6. Learn a relationship between standard error and a subset of pointwise estimates of reusability over validation runs. Training runs are excluded from this step; their confidence intervals are biased to a smaller range due to the fact that they contributed all of the pooled judgments.
7. Choose remaining sites' runs to be *testing* runs.
8. Estimate confidence interval widths for these runs using the function learned in step 6.

The confidence in the evaluation is the width of the 95% confidence interval. If the interval is wide, there is a great deal of uncertainty in the evaluation; more judgments are needed to understand the quality of the system. If it is narrow, it is unlikely that more judgments would be necessary.

4.3 Evaluation

We will evaluate results primarily by our ability to rank validation and testing runs, the average width of the 95% confidence intervals for validation and testing runs (which is a linear function of standard error), and our ability to predict standard error of testing runs using point estimators in a function trained over validation runs.

Kendall's τ rank correlation is frequently used to evaluate the quality of rankings of systems. τ is proportional to the number of pairs that have swapped between two rankings. A perfect τ is 1, meaning no pairs have swapped; $\tau = 0$ means half the pairs are swapped. For IR, $\tau \geq 0.9$ is widely considered the best that can be expected in the presence of assessor disagreement.

We report the standard error of MAP or precision rather than the width of a particular confidence interval. The interval can be calculated using the equation in Section 3.1.3. To evaluate our predictions of confidence interval width, we calculate Pearson's linear correlation between "true" standard error (calculated with relevance probabilities estimated from a particular set of features) and predicted standard errors.

All numbers are evaluated over multiple trials, randomly choosing m_1 training sites and m_2 validation sites each time.

4.4 Example

In this section we run through a single experiment in detail as an example. The next section presents results averaged over multiple experiments.

First, we choose $m_1 = 1$ site whose runs will be used to form a pool. The (randomly selected) site submitted four runs, which did not do a particularly good job of finding relevant documents; they were ranked 54th, 65th, 67th, and 68th by MAP among all 74 submitted runs.

We judged the top $k = 10$ documents retrieved by each of these four runs for each of the 225 queries. This produced 8,183 relevance judgments, 36 per query on average, of which 2.6% were relevant. These are the only judgments we have for evaluating the remaining 70 runs from the other 17 sites.

Next, we choose $m_2 = 5$ sites whose runs will be evaluated by expected MAP. The five sites submitted 19 runs of varying quality: the lowest rank any appeared at was 66th;

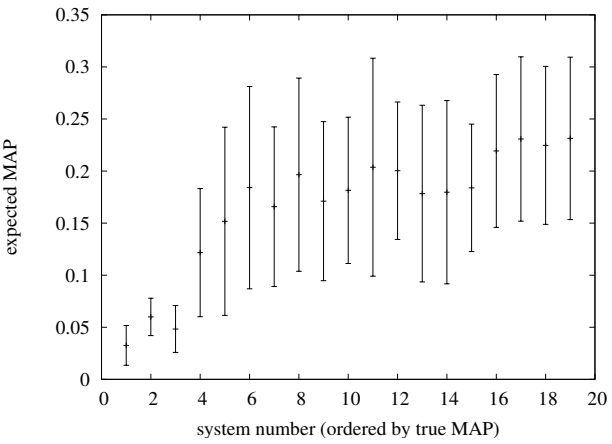

Figure 1: Example 1: one site submitted runs that contributed to a pool of depth 10. These judgments are used to rank 19 new systems from five sites by expected MAP, and to calculate confidence intervals.

the highest was 10th. The median rank of these 19 runs is 37. On average they retrieved 50% of the judged documents; over 90% of their documents were not judged. No run retrieved more than 70% of the judged documents, and no run retrieved less than 6%.

We then use the judgments and features extracted from the combined $m_1 + m_2 = 6$ sites' runs to estimate the relevance of the unjudged documents; these relevance estimates are used to calculate expected MAP and its variance. We used the logistic regression model with document similarity features described in Section 3.1.1. The resulting ranking of the 19 validation runs by EMAP has a Kendall's τ rank correlation of 0.743 with the "true" ranking by MAP, suggesting that the judgments are not good enough to evaluate systems accurately. But the real question concerns the confidence intervals: the average standard error is 0.0359, meaning MAP could be as much as ± 0.07 away from the estimated value on average. If the estimated MAP is 0.25, then, the true MAP has a 95% chance of being between 0.18 and 0.32.

Figure 1 shows the expected MAPs and 95% confidence intervals for the 19 validation systems ordered by true MAP. Note that the intervals are quite wide. More judgments would be needed for all but the worst-performing systems. However, the confidence intervals are accurate in that they contain each system's "true" rank by MAP. For instance, the top-ranked system and the 4th-from-last have little overlap in their confidence intervals; comparing these two we would probably not require more judgments. On the other hand, there is significant overlap among most of these systems; comparing them accurately will require more judgments. The three lowest-ranked systems have the greatest confidence, but actually retrieved the *most* unjudged documents. This supports our hypothesis above about confidence in systems that fail to retrieve known relevant documents.

Now we would like a simpler way to estimate the width of the confidence interval. Measuring the correlation between standard error and various point estimators (including recall, MAP, and MAR) over the validation runs, we find that recall has the best linear relationship to standard error: 0.916. This means that when recall of judged relevant documents is high, standard error is high, and the confidence interval is wide; when recall of judged relevant documents is low, standard error is low, and the confidence interval is narrow. This is more evidence in favor of our hypothesis about the relationship between retrieving judged relevant documents and confidence.

Though recall is a good predictor on its own in this case, using a linear combination of recall and MAR improves the predictions significantly in many cases. We fit a linear regression model[1] to the standard error as a function of recall of judged relevant documents (averaged over queries) and MAR. The data points are the 19 validation runs. The 4 training runs are excluded so as not to bias the function. The result of the linear regression is the function

$$\widehat{\sigma} = -0.0023 + 1.1154 \cdot rec + 0.1088 \cdot MAR$$

We use this function to estimate the standard error of the $m_3 = 12$ testing sites that submitted the remaining 51 runs.

The "gold standard" for standard error among the testing runs is that computed by the same procedure used to compute standard error for the validation runs. We compare the regression predictions to this gold standard, finding a correlation of 0.913. Figure 2 shows the "true" standard errors versus the predicted standard errors in two ways: on the top, the true vs. predicted; on the bottom, the resulting 95% confidence intervals for both types. Though the predictions aren't perfect, it is clear that a high predicted standard error indicates a high "true" standard error. The true and predicted confidence intervals for each system overlap substantially. Thus it is clear that a high predicted standard error indicates that more judgments are necessary to understand the quality of the system with confidence.

The only issue remaining is the lack of an absolute threshold of confidence interval width for determining whether to more judgments are necessary. To a large extent, the decision depends on how the confidence intervals compare to those of other systems. Even if a confidence interval is very wide, if it does not overlap with those of any other system it is not necessary to judge more documents to understand its relative performance.

A second example demonstrates the effect of taking more judgments from a more diverse set of runs. This time we took $m_1 = 3$ sites randomly to form a pool of the top $k = 10$ documents retrieved by each. The total number of judgments is only slightly higher—8,679 instead of 8,183—but twice as many were relevant, and the set of systems that supplied them more diverse. The result is that the τ correlation between expected MAP and true MAP on the $m_2 = 5$ validation systems is 0.971, and the confidence intervals for each system are much tighter (Figure 3).

4.5 Experimental Results

We performed all steps in Section 4.2, as described in detail in Section 4.4, multiple times, each time randomly choosing different training and validation runs, for increasing m_1, m_2, and k. The numbers we present here are averaged over 25 trials for each m_1, m_2, k.

Table 2 shows results for the Web track runs with probability estimates from the logistic regression model with document similarity features. In this table we can see the effect

[1]We chose linear regression primarily for its simplicity. We have no prior reason to suppose the relationship is linear.

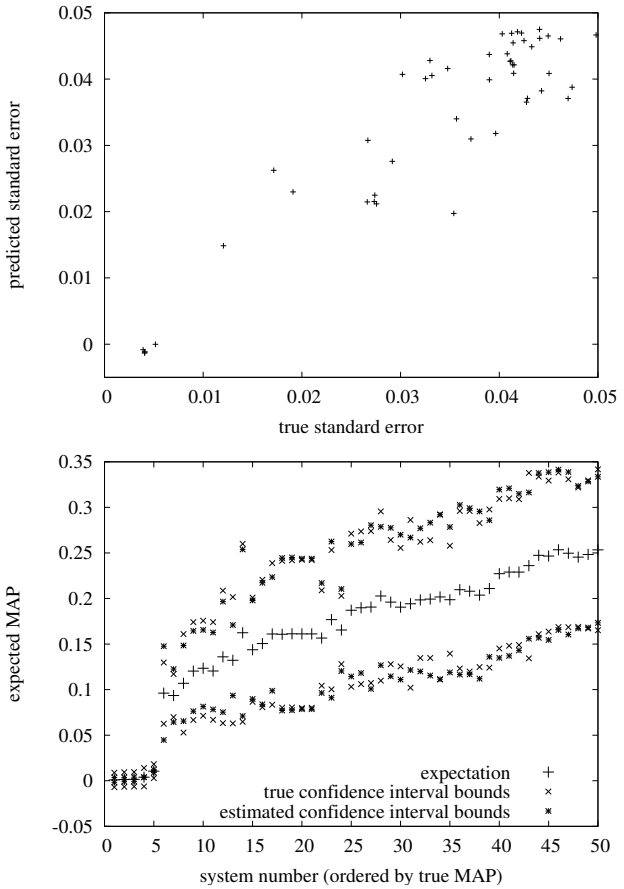

Figure 2: Standard error prediction (top) and expected MAP with both "true" confidence intervals and predicted confidence intervals (bottom) for testing systems.

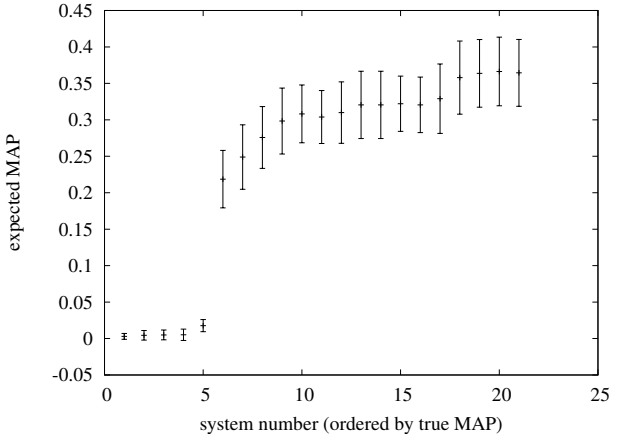

Figure 3: Example 2: three sites submitted runs that contributed to a pool of depth 10. The resulting confidence intervals on new runs are tighter than those in Figure 1.

of increasing the pool depth as well as increasing the number of sites that contribute to the pool: in both cases, as they increase, we are both better able to rank systems and have more confidence in the individual system results.

m_1	m_2	k	judged	rel	τ_{test}	σ_{test}	$\rho_{\sigma_{m_3},\widehat{\sigma_{m_3}}}$
1	5	1	644	103	0.536	0.010	0.899
1	5	5	2,247	224	0.855	0.026	0.953
1	5	10	3,885	257	0.870	0.025	0.915
1	5	20	7,367	392	0.901	0.022	0.928
1	10	5	2,197	199	0.829	0.028	0.904
1	10	10	4,175	269	0.851	0.025	0.888
1	10	20	7,200	371	0.875	0.023	0.865
3	5	1	1,028	206	0.812	0.025	0.891
3	5	5	5,267	361	0.926	0.026	0.915
3	5	10	10,372	520	0.960	0.017	0.921
3	5	20	20,573	668	0.968	0.012	0.869
5	5	1	1,724	252	0.872	0.031	0.905
5	5	5	7,287	478	0.960	0.018	0.892
5	5	10	15,941	611	0.970	0.013	0.871
5	5	20	27,000	777	0.975	0.009	0.855

Table 2: Results for Web track runs evaluated by MAP. The first three columns report experimental parameters. The next two report total number of judged documents and judged relevant documents for 225 queries averaged over 25 trials with the same parameters. Results are Kendall's τ correlation between rankings of m_2+m_3 validation and test systems by true and expected MAP, mean standard error of MAP for those systems, and linear correlation between standard error and estimated standard error for the m_3 test systems.

The effect of increasing the number of sites that contribute to the pool can be seen by looking at the experiments for which about 7,200 documents were judged (lines 4, 7, 13 in Table 2). Increasing the number of sites while keeping the total number of judgments constant both increases the rank correlation and decreases the variance in the individual system predictions. This suggests that more diversity in the input systems results in better reusability.

We are able to predict confidence interval width with ease: the correlation between true standard error and predicted standard error is always high. Interestingly, more judgments tend to make the predictions worse, possibly because variances are decreasing overall.

The only effect of increasing m_2 is to generate more training data for the linear regression standard error estimator and to reduce the amount testing data m_3. This did not seem to have any effect on our ability to predict standard error; in fact, the predictions obtained with $m_2 = 10$ are actually worse than those from $m_2 = 5$. We therefore focus on the effect of varying the first group of m_1 sites (for varying diversity among the initial set of judgments) and the pool depths (for varying total numbers of judgments).

One takeaway message from Table 2 is that very few judgments are needed to be able to accurately rank new systems. The τ correlations for the testing sets are very high, surpassing 0.9 with only 23 judgments per query (at $m_1 = 3, k = 5$). The more salient requirement is that there be a modicum of diversity among the runs submitting judgments, but even a small amount of diversity goes a long way.

eval	m_1	k	τ_{test}	σ_{test}	$\rho_{\sigma_{m_3},\widehat{\sigma_{m_3}}}$
p@5	1	10	0.762	0.007	0.758
p@10	1	10	0.721	0.005	0.639
p@5	3	5	0.863	0.007	0.740
p@10	3	5	0.823	0.005	0.772
p@5	3	10	0.903	0.005	0.718
p@10	3	10	0.873	0.004	0.656
p@5	5	20	0.935	0.003	0.667
p@10	5	20	0.925	0.002	0.665

Table 3: Results evaluated by precision at ranks 5 and 10 for a subset of the experiments in Table 2: mean Kendall's τ correlation between rankings of $m_2 + m_3$ validation and test systems by true precision and by estimated precision, mean standard error of precision for the same systems, and mean linear correlation between standard error and predicted standard error for the m_3 test systems.

4.5.1 Comparing Feature Sets

As described in Section 3.1.2, calculating expectations of MAP and precision requires estimates of the probability of relevance that are obtained by training a model with features. Table 2 used document similarity features; we can also use our point estimates as features of documents to predict relevance. Here we compare the two in terms of accuracy at predicting true MAP (RMSE between expected map and true map), width of resulting confidence interval, and ability to find the right ranking of systems.

Using system-based features in some pilot experiments tended to result in better predictions of MAP: the root mean square error between expected MAP and true MAP is slightly less. Despite the predictions being better, the ranking performance was much worse, even producing negative τ correlations in some cases. In addition, the confidence intervals on MAP are substantially wider. There is therefore no basis for using these features rather than the document similarity features, and we did not continue testing them beyond the small set of pilot experiments.

4.5.2 Results for Precision Measures

For Web-type tasks, precision at high ranks is often more important than a full-list recall-based metric like MAP. Table 3 shows a subset of results from Table 2, except evaluated by precision at ranks 5 and 10 rather than MAP.

It is a little harder to rank systems by precision; the correlation results are lower than the MAP correlation results with the same experimental parameters. The confidence intervals are much tighter, however, indicating that each system has fewer possible alternative rankings. The correlation numbers are therefore a bit misleading in that while they are lower, the tighter confidence intervals restrict the possible alternative rankings to a smaller set. It should be "easier" to find the true ranking by judging more documents.

Our point estimators do not do as good a job of predicting precision standard errors as they did of predicting MAP standard errors. This is likely because the point estimators were calculated over the entire list, like MAP.

4.5.3 Results by Task

The Web track runs used 225 topics from three different categories: topic distillation, named page finding, and home page finding. The results above are averaged over all three types. The named page and home page topics may cause the correlation results to seem better than they really are, since it is relatively "easy" to find the relevant documents for these types of topics.

task	m_1	k	judged	rel	τ_{test}	σ_{test}	$\rho_{\sigma_{m_3},\widehat{\sigma_{m_3}}}$
ah	1	10	1,162	392	0.779	0.011	0.778
td	1	10	1,580	166	0.683	0.032	0.766
hp	1	10	1,363	50	0.789	0.009	0.688
np	1	10	1,232	53	0.799	0.009	0.641
ah	3	5	1,393	520	0.817	0.007	0.832
td	3	5	1,955	223	0.776	0.029	0.832
hp	3	5	1,718	68	0.902	0.009	0.680
np	3	5	1,594	70	0.892	0.008	0.600
ah	3	10	2,663	904	0.843	0.008	0.821
td	3	10	3,830	368	0.833	0.022	0.857
hp	3	10	3,376	77	0.944	0.005	0.591
np	3	10	3,167	75	0.928	0.005	0.577
ah	5	20	7,095	1,867	0.846	0.006	0.783
td	5	20	9,889	617	0.888	0.014	0.812
hp	5	20	8,802	82	0.962	0.002	0.667
np	5	20	8,309	78	0.967	0.002	0.618

Table 4: Results broken out by task. Topic distillation (td; Web track runs) and ad hoc (ah; Robust track runs) are evaluated by MAP. Homepage (hp) and named page (np) are evaluated by precision@5.

Table 4 shows results for selected parameter values broken out by task. This table includes both the Web track and Robust track runs; the task identifies which set the results are for (ah=Robust; td,np,hp=Web). We evaluated different topic types with different metrics: topic distillation and ad hoc topics were evaluated by MAP while home page and named page finding were evaluated by precision at rank 5.

The topic distillation task is the "hardest" to evaluate, in that its rank correlations tend to be the worst compared to the other tasks with (roughly) the same number of judgments, and the confidence intervals significantly wider. One possible reason for this is that the estimates of probability of relevance are bad; it is unlikely that the cluster hypothesis holds for topic distillation.

5. CONCLUSIONS AND FUTURE WORK

This paper examined the problem of quantifying the reusability of a test collection with respect to a new system, which is important from both a theoretical and practical perspective. We argued that performance prediction approaches, which have been the focus of most previous studies, are not suitable for quantifying reusability because they do not measure the confidence of the prediction. We proposed quantifying reusability by estimating confidence intervals of new system performance. If the intervals are tight, then the existing judgments are suitable for evaluating the new system. However, if the intervals are wide, then it is likely that more relevance judgments are necessary to accurately evaluate the new system.

We evaluated two approaches for quantifying reusability. The first uses a logistic regression model to estimate the relevance of unjudged documents using document similarity and system-based features. Once the relevance of unjudged document are estimated, it is then possible to estimate confidence intervals for standard retrieval metrics, such as pre-

cision at k and mean average precision. The second method converts traditional information retrieval metrics into point measures of reusability.

Our experimental results, based on simulations of actual TREC relevance judgments and submissions, showed that the confidence intervals estimated were accurate, in that they always contained the actual mean average precision value of the system. The results also showed that ranking by expected system performance, with a very small number of judgments, was highly correlated ($\tau > 0.9$) with the actual ranking of systems with complete judgments. Finally, we showed there was a high correlation ($\rho > 0.9$) between a linear combination of our point estimates of reusability, namely recall and mean average reuse, and the width of the estimated confidence intervals. This suggests that these measures, which are very simple to compute, are good proxies for confidence interval widths, making them suitable reusability measures for many retrieval tasks.

There are several areas of future work, including extending the approaches proposed here to non-binary judgments, exploring richer feature sets for predicting the relevance of unjudged documents, and potentially using the proposed reusability measures within a learning to rank framework to produce more robust ranking functions.

6. REFERENCES

[1] J. A. Aslam and E. Yilmaz. Inferring document relevance from incomplete information. In *Proc. 16th Intl. Conf. on Information and Knowledge Management*, pages 633–642. ACM, 2007.

[2] C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees. Bias and the limits of pooling. In *Proc. 29th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 620–621, 2006.

[3] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 25–32. ACM, 2004.

[4] S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *Proc. 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 63–70. ACM, 2007.

[5] B. Carterette. Robust test collections for information retrieval evaluation. In *Proc. 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 55–62, 2007.

[6] B. Carterette. *Low-Cost and Robust Evaluation of Information Retrieval Systems*. PhD thesis, University of Massachuetts, 2008.

[7] B. Carterette and J. Allan. Semiautomatic evaluation of retrieval systems using document similarities. In *Proc. 16th Intl. Conf. on Information and Knowledge Management*, pages 873–876, New York, NY, USA, 2007. ACM.

[8] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1994.

[9] E. C. Jensen, S. M. Beitzel, A. Chowdhury, and O. Frieder. Repeatable evaluation of search services in dynamic environments. *Transactions on Information Systems*, 26(1):1, 2007.

[10] T. Sakai. Alternatives to bpref. In *Proc. 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 71–78. ACM, 2007.

[11] T. Sakai. Comparing metrics across TREC and NTCIR: the robustness to system bias. In *Proc. 17th Intl. Conf. on Information and Knowledge Management*, pages 581–590. ACM, 2008.

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

[13] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

[14] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge, MA, 2005.

[15] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proc. 15th Intl. Conf. on Information and Knowledge Management*, pages 102–111, 2006.

[16] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proc. 21st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 307–314, 1998.