Robust Test Collections for Retrieval Evaluation

Ben Carterette
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts Amherst
Amherst, MA 01003
carteret@cs.umass.edu

ABSTRACT

Low-cost methods for acquiring relevance judgments can be a boon to researchers who need to evaluate new retrieval tasks or topics but do not have the resources to make thousands of judgments. While these judgments are very useful for a one-time evaluation, it is not clear that they can be trusted when re-used to evaluate new systems. In this work, we formally define what it means for judgments to be reusable: the confidence in an evaluation of new systems can be accurately assessed from an existing set of relevance judgments. We then present a method for augmenting a set of relevance judgments with relevance estimates that requires no additional assessor effort. Using this method practically guarantees reusability: with as few as five judgments per topic taken from only two systems, we can reliably evaluate a larger set of ten systems. This makes even the smallest sets of judgments useful for evaluation of new systems.

Categories and Subject Descriptors: H.3 Information Storage and Retrieval; H.3.4 Systems and Software: Performance Evaluation

General Terms: Experimentation, Measurement

Keywords: information retrieval, evaluation, test collections, reusability

1. INTRODUCTION

Consider an information retrieval researcher who has invented a new retrieval task. She has built a system to perform the task and wants to evaluate it. Since the task is new, it is unlikely that there are any extant relevance judgments. She does not have the time or resources to judge every document, or even every retrieved document. She can only judge the documents that seem to be the most informative and stop when she has a reasonable degree of confidence in her conclusions. But what happens when she develops a new system and needs to evaluate it? Or another research group decides to implement a system to perform the task? Can they reliably reuse the judgments she originally made?

Can they evaluate without more relevance judgments?

Evaluation is an important aspect of information retrieval research, but it is only a semi-solved problem: for most retrieval tasks, it is impossible to judge the relevance of every document; there are simply too many of them. The solution used by NIST at TREC (Text REtrieval Conference) is the pooling method [19, 20]: all competing systems contribute N documents to a pool, and every document in that pool is judged. This method creates large sets of judgments that are reusable for training or evaluating new systems that did not contribute to the pool [21].

This solution is not adequate for our hypothetical researcher. The pooling method gives thousands of relevance judgments, but it requires many hours of (paid) annotator time. As a result, there have been a slew of recent papers on reducing annotator effort in producing test collections: Cormack et al. [11], Zobel [21], Sanderson and Joho [17], Carterette et al. [8], and Aslam et al. [4], among others. As we will see, the judgments these methods produce can significantly bias the evaluation of a new set of systems.

So can she reuse her relevance judgments? First we must formally define what it means to be "reusable". In previous work, reusability has been tested by simply assessing the accuracy of a set of relevance judgments at evaluating unseen systems. While we can say that it was right 75% of the time, or that it had a rank correlation of 0.8, these numbers do not have any predictive power: they do not tell us *which* systems are likely to be wrong or how confident we should be in any one. We need a more careful definition of reusability.

Specifically, the question of reusability is not how accurately we can evaluate new systems. A "malicious adversary" can always produce a new ranked list that has not retrieved any of the judged documents. The real question is how much *confidence* we have in our evaluations, and, more importantly, whether we can *trust* our estimates of confidence. Even if confidence is not high, as long as we can trust it, we can identify which systems need more judgments in order to increase confidence. *Any* set of judgments, no matter how small, becomes reusable to some degree.

Small, reusable test collections could have a huge impact on information retrieval research. Research groups would be able to share the relevance judgments they have done "in-house" for pilot studies, new tasks, or new topics. The amount of data available to researchers would grow exponentially over time.

2. ROBUST EVALUATION

Above we gave an intuitive definition of reusability: a collection is reusable if we can "trust" our estimates of confidence in an evaluation. By that we mean that if we have done some relevance judgments and have, for example, 75% confidence that system A is better than system B, we would like there to be no more than 25% chance that our assessment of the relative quality of the systems will change as we continue to judge documents. Our evaluation should be *robust* to missing judgments.

In our previous work, we defined confidence as the probability that the difference in an evaluation measure calculated for two systems is less than zero [8]. This notion of confidence is defined in the context of a particular evaluation task that we call *comparative evaluation*: determining the sign of the difference in an evaluation measure. Other tasks having different notions of confidence could be defined; estimating the magnitude of the difference or the values of the measures themselves are examples.

Confidence, therefore, is a probability estimate. One of the questions we must ask about a probability estimate is what it means. What does it mean to have 75% confidence that system A is better than system B? As described above, we want it to mean that if we continue to judge documents, there will only be a 25% chance that our assessment will change. If this is what it means, we can "trust" the confidence estimates. But do we know it has that meaning?

Our calculation of confidence rests on an assumption about the probability of relevance of unjudged documents, specifically that each unjudged document was equally likely to be relevant or nonrelevant. This assumption is almost certainly not realistic in most IR applications. As it turns out, it is this assumption that determines whether we can trust the confidence estimates. Before elaborating on this, we will stop to define confidence.

2.1 Estimating Confidence

Average precision (AP) is a standard evaluation metric that captures both the ability of a system to rank relevant documents highly (precision) and its ability to retrieve relevant documents (recall). It is typically written as the mean precision at the ranks of relevant documents:

$$AP = \frac{1}{|R|} \sum_{i \in R} prec@r(i)$$

where R is the set of relevant documents, and $r(i)$ is the rank of document i. Let X_i be a random variable indicating the relevance of document i. If documents are ordered by rank, we can express precision as $prec@i = 1/i \sum_{j=1}^{i} X_j$.

Average precision becomes the quadratic equation

$$AP = \frac{1}{\sum X_i} \sum_{i=1}^{n} X_i/i \sum_{j=1}^{i} X_j$$

$$= \frac{1}{\sum X_i} \sum_{i=1}^{n} \sum_{j \geq i} a_{ij} X_i X_j$$

where $a_{ij} = 1/\max\{r(i), r(j)\}$. Using a_{ij} instead of $1/i$ allows us to number the documents arbitrarily. To see why this is true, consider a toy example: a list of 3 documents with relevant documents B, C at ranks 1 and 3 and nonrelevant document A at rank 2. Average precision will be $\frac{1}{2}(\frac{1}{1}x_B^2 + \frac{1}{2}x_B x_A + \frac{1}{3}x_B x_C + \frac{1}{2}x_A^2 + \frac{1}{3}x_A x_C + \frac{1}{3}x_C^2) = \frac{1}{2}\left(1 + \frac{2}{3}\right)$

because $x_A = 0, x_B = 1, x_C = 1$. Though the ordering B, A, C is different from the labeling A, B, C, it does not affect the computation.

We can now see average precision itself is a random variable with a distribution over all possible assignments of relevance to all documents. This random variable has an expectation, a variance, confidence intervals, and a certain probability of being less than or equal to a given value. All of these are dependent on the probability that document i is relevant: $p_i = p(X_i = 1)$.

Suppose in our previous example we do not know the relevance judgments, but we believe $p_A = 0.4, p_B = 0.8, p_C = 0.7$. We can then compute e.g. $P(AP = 0) = 0.2 \cdot 0.6 \cdot 0.3 = 0.036$, or $P(AP = \frac{1}{2}) = 0.2 \cdot 0.4 \cdot 0.7 = 0.056$.

The expectation and variance of AP are:

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

$$Var[AP] \approx \frac{1}{(\sum p_i)^2} \left(\sum_{i}^{n} a_{ii}^2 p_i q_i + \sum_{j>i} a_{ij}^2 p_i p_j (1 - p_i p_j) \right.$$
$$\left. + \sum_{i \neq j} 2 a_{ii} a_{ij} p_i p_j (1 - p_i) + \sum_{k>j \neq i} 2 a_{ij} a_{ik} p_i p_j p_k (1 - p_i) \right)$$

AP asymptotically converges to a normal distribution with expectation and variance as defined above.[1]

For our comparative evaluation task we are interested in the sign of the difference in two average precisions: $\Delta AP = AP_1 - AP_2$. As we showed in our previous work, ΔAP has a closed form when documents are ordered arbitrarily:

$$\Delta AP = \frac{1}{\sum X_i} \sum_{i=1}^{n} \sum_{j \geq i} c_{ij} X_i X_j$$

$$c_{ij} = a_{ij} - b_{ij}$$

where b_{ij} is defined analogously to a_{ij} for the second ranking. Since AP is normal, ΔAP is normal as well, meaning we can use the normal cumulative density function to determine the confidence that a difference in AP is less than zero.

Since topics are independent, we can easily extend this to mean average precision (MAP). MAP is also normally distributed with expectation and variance:

$$\mathcal{E}MAP = \frac{1}{T} \sum_{t \in T} E[AP_t] \qquad (1)$$

$$\mathcal{V}MAP = \frac{1}{T^2} \sum_{t \in T} Var[AP_t]$$

$$\Delta MAP = MAP_1 - MAP_2$$

Confidence can then be estimated by calculating the expectation and variance and using the normal density function to find $P(\Delta MAP < 0)$.

As we discussed earlier, whether the confidence estimates are trustworthy depends on our estimates of p_i.

2.2 Confidence and Robustness

Having defined confidence, we turn back to the issue of trust in confidence estimates, and show how it ties into the robustness of the collection to missing judgments.

---

[1] These are actually approximations to the true expectation and variance, but the error is a negligible $\mathcal{O}(n2^{-n})$.

Let $\mathcal{Z}$ be the set of all pairs of ranked lists. Suppose we have a set of m relevance judgments $x^m = \{x_1, x_2, ..., x_m\}$ (using small x rather than capital X to distinguish between judged and unjudged documents); these are the judgments against which we compute confidence. Let $\mathcal{Z}_\alpha$ be the subset of pairs in $\mathcal{Z}$ for which we predict that $\Delta MAP = -1$ with confidence α given the judgments x^m. For the confidence estimates to be accurate, we need at least $\alpha \cdot |\mathcal{Z}_\alpha|$ of these pairs to actually have $\Delta MAP = -1$ after we have judged *every* document. If they do, we can trust the confidence estimates; our evaluation will be robust to missing judgments.

If our confidence estimates are based on unrealistic assumptions, we cannot expect them to be accurate. The assumptions they are based on are the probabilities of relevance p_i. We need these to be "realistic".

We argue that the best possible distribution of relevance $p(X_i)$ is the one that explains all of the data (all of the observations made about the retrieval systems) while at the same time making no unwarranted assumptions. This is known as the *principle of maximum entropy* [13].

The entropy of a random variable X with distribution $p(X)$ is defined as $H(p) = -\sum_i p(X = i) \log p(X = i)$. This has found a wide spectrum of uses in computer science and information retrieval. The maximum entropy distribution is the one that maximizes H. This distribution is unique and has an exponential form. The following theorem shows the utility of a maximum entropy distribution for relevance when estimating confidence.

THEOREM 1. *If $p(X^n | I, x^m) = argmax_p H(p)$, confidence estimates will be accurate.*

where x^m is the set of relevance judgments defined above, X^n is the full set of documents that we wish to estimate the relevance of, and I is some information about the documents (unspecified as of now). We forgo the proof for the time being, but it is quite simple.

This says that *the better the estimates of relevance, the more accurate the evaluation.* The task of creating a reusable test collection thus becomes the task of estimating the relevance of unjudged documents.

The theorem and its proof say nothing whatsoever about the evaluation metric. The probability estimates are entirely indepedent of the measure we are interested in. This means the same probability estimates can tell us about average precision as well as precision, recall, bpref, etc.

Furthermore, we could assume that the relevance of documents i and j is independent and achieve the same result, which we state as a corollary:

COROLLARY 1. *If $p(X_i | I, x^m) = argmax_p H(p)$, confidence estimates will be accurate.*

The task therefore becomes the imputation of the missing values of relevance. The theorem implies that the closer we get to the maximum entropy distribution of relevance, the closer we get to robustness.

3. PREDICTING RELEVANCE

In our statement of Theorem 1, we left the nature of the information I unspecified. One of the advantages of our confidence estimates is that they admit information from a wide variety of sources; essentially anything that can be modeled can be used as information for predicting relevance. A natural source of information is the retrieval systems themselves: how they ranked the judged documents, how often they failed to rank relevant documents, how they perform across topics, and so on. If we treat each system as an information retrieval "expert" providing an opinion about the relevance of each document, the problem becomes one of expert opinion aggregation.

This is similar to the metasearch or data fusion problem in which the task is to take k input systems and merge them into a single ranking. Aslam et al. [3] previously identified a connection between evaluation and metasearch. Our problem has two key differences:

1. We explicitly need probabilities of relevance that we can plug into Eq. 1; metasearch algorithms have no such requirement.

2. We are accumulating relevance judgments as we proceed with the evaluation and are able to re-estimate relevance given each new judgment.

In light of (1) above, we introduce a probabilistic model for expert combination.

3.1 A Model for Expert Opinion Aggregation

Suppose that each expert j provides a probability of relevance $q_{ij} = p_j(X_i = 1)$. The information about the relevance of document i will then be the set of k expert opinions $I = \mathbf{q_i} = (q_{i1}, q_{i2}, \cdots, q_{ik})$. The probability distribution we wish to find is the one that maximizes the entropy of $p_i = p(X_i = 1 | \mathbf{q_i})$.

As it turns out, finding the maximum entropy model is equivalent to finding the parameters that maximize the likelihood [5]. Blower [6] explicitly shows that finding the maximum entropy model for a binary variable is equivalent to solving a logistic regression. Then

$$ p_i = p(X_i = 1 | \mathbf{q_i}) = \frac{\exp\left(\sum_{j=1}^{k} \lambda_j q_{ij}\right)}{1 + \exp\left(\sum_{j=1}^{k} \lambda_j q_{ij}\right)} \qquad (2) $$

where $\lambda_1, \cdots, \lambda_k$ are the regression parameters. We include a beta prior for $p(\lambda_j)$ with parameters α, β. This can be seen as a type of smoothing to account for the fact that the training data is highly biased.

This model has the advantage of including the statistical dependence between the experts. A model of the same form was shown by Clemen & Winkler to be the best for aggregating expert probabilities [10]. A similar maximum-entropy-motivated approach has been used for expert aggregation [15]. Aslam & Montague [1] used a similar model for metasearch, but assumed independence among experts.

Where do the q_{ij}s come from? Using raw, uncalibrated scores as predictors will not work because score distributions vary too much between topics. A language modeling ranker, for instance, will typically give a much higher score to the top retrieved document for a short query than to the top retrieved document for a long query.

We could learn a separate predicting model for each topic, but that does not take advantage of all of the information we have: we may only have a handful of judgments for a topic, not enough to train a model to any confidence. Furthermore, it seems reasonable to assume that if an expert makes good predictions for one topic, it will make good predictions for other topics as well. We could use a hierarchical model [12], but that will not generalize to unseen topics. Instead, we

will calibrate the scores of each expert individually so that scores can be compared both within topic and between topic. Thus our model takes into account not only the dependence between experts, but also the dependence between experts' performances on different tasks (topics).

3.2 Calibrating Experts

Each expert gives us a score and a rank for each document. We need to convert these to probabilities. A method such as the one used by Manmatha et al. [14] could be used to convert scores into probabilities of relevance. The "pairwise preference" method of Carterette & Petkova [9] could also be used, interpeting the ranking of one document over another as an expression of preference.

Let q_{ij}^* be expert j's self-reported probability that document i is relevant. Intuitively it seems clear that q_{ij}^* should decrease with rank, and it should be zero if document i is unranked (the expert did not believe it to be relevant). The pairwise preference model can handle these two requirements easily, so we will use it. Let $\theta_{r_j(i)}$ be the "relevance coefficient" of the document at rank $r_j(i)$. We want to find the θs that maximize the likelihood function:

$$L_{jt}(\Theta) = \prod_{r_j(i) < r_j(k)} \frac{\exp(\theta_{r_j(i)} - \theta_{r_j(k)})}{1 + \exp(\theta_{r_j(i)} - \theta_{r_j(k)})}$$

We again include a beta prior on $p(\theta_{r_j(i)})$ with parameters $|R_t| + 1$ and $|N_t| + 1$, the size of the sets of judged relevant and nonrelevant documents respectively. Using these as prior parameters ensures that the resulting probabilities will be concentrated around the ratio of relevant documents that have been discovered for topic t. This means that the probability estimates decrease by rank and are higher for topics that have more relevant documents.

After finding the Θ that maximizes the likelihood, we have $q_{ij}^* = \frac{\exp(\theta_{r_j(i)})}{1+\exp(\theta_{r_j(i)})}$. We define $\theta_\infty = -\infty$, so that the probability that an unranked document is relevant is 0.

Since q_{ij}^* is based on the rank at which a document is retrieved rather than the identity of the document itself, the probabilities are identical from expert to expert, e.g. if expert E put document A at rank 1, and expert D put document B at rank 1, we will have $q_{AE}^* = q_{BD}^*$. Therefore we only have to solve this once for each topic.

The above model gives topic-independent probabilities for each document. But suppose an expert who reports 90% probability is only right 50% of the time. Its opinion should be discounted based on its observed performance. Specifically, we want to learn a calibration function $q_{ij} = C_j(q_{ij}^*)$ that will ensure that the predicted probabilities are tuned to the expert's ability to retrieve relevant documents given the judgments that have been made to this point.

Platt's SVM calibration method [16] fits a sigmoid function between q_{ij}^* and the relevance judgments to obtain $q_{ij} = C_j(q_{ij}^*) = \frac{\exp(A_j + B_j q_{ij}^*)}{1+\exp(A_j + B_j q_{ij}^*)}$. Since q_{ij}^* is topic-independent, we only need to learn one calibration function for each expert.

Once we have the calibration function, it is applied to adjust the experts' predictions to their actual performance. The calibrated probabilities are plugged into model (2) to find the document probabilities.
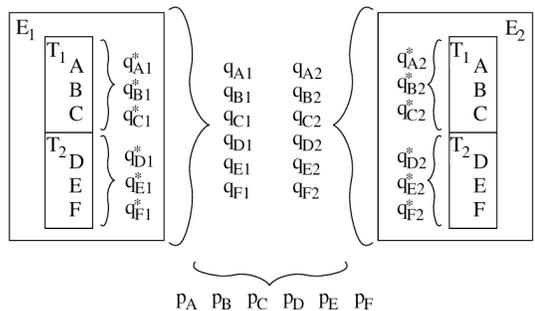

Figure 1: Conceptual diagram of our aggregation model. Experts E_1, E_2 have ranked documents A, B, C for topic T_1 and documents D, E, F for topic T_2. The first step is to obtain q_{ij}^*. Next is calibration to true performance to find q_{ij}. Finally we obtain $p_i = p(X_i = 1|q_{i1}, q_{i2}), \cdots$.

3.3 Model Summary

Our model has three components that differ by the data they take as input and what they produce as output. A conceptual diagram is shown in Figure 1.

1. ranks → probabilities (per system per topic). This gives us q_{ij}^*, expert j's self-reported probability of the relevance of document i. This is unsupervised; it requires no labeled data (though if we have some, we use it to set prior parameters).

2. probabilities → calibrated probabilities (per system). This gives us $q_{ij} = C_j(q_{ij}^*)$, expert j's calibrated probability of the relevance of document i. This is semi-supervised; we have relevance judgments at some ranks which we use to impute the probability of relevance at other ranks.

3. calibrated probabilities → document probabilities. This gives us $p_i = p(X_i = 1|\mathbf{q_i})$, the probability of relevance of document i given calibrated expert probabilities q_{ij}. This is supervised; we learn coefficients from a set of judged documents and use those to estimate the relevance of the unjudged documents.

Although the model appears rather complex, it is really just three successive applications of logistic regression. As such, it can be implemented in a statistical programming language such as R in a few lines of code. The use of beta (conjugate) priors ensures that no expensive computational methods such as MCMC are necessary [12], so the model is trained and applied fast enough to be used on-line. Our code is available at http://ciir.cs.umass.edu/~carteret/.

4. EXPERIMENTS

Three hypotheses are under consideration. The first, and most important, is that using our expert aggregation model to predict relevance produces test collections that are robust enough to be reusable; that is, we can trust the estimates of confidence when we evaluate systems that did not contribute any judgments to the pool.

The other two hypotheses relate to the improvement we see by using better estimates of relevance than we did in our previous work [8]. These are that (a) it takes fewer relevance

track	no. topics	no. runs	no. judged	no. rel
ad hoc 94	50	40	97,319	9,805
ad hoc 95	49	33	87,069	6,503
ad hoc 96	50	61	133,681	5,524
ad hoc 97	50	74	72,270	4,611
ad hoc 98	50	103	80,345	4,674
ad hoc 99	50	129	86,830	4,728
web 04	225	74	88,566	1,763
robust 05	50	74	37,798	6,561
terabyte 05	50	58	45,291	10,407

Table 1: Number of topics, number of runs, number of documents judged, and number found relevant for each of our data sets.

judgments to reach 95% confidence and (b) the accuracy of the predictions is higher than if we were to simply assume $p_i = 0.5$ for all unjudged documents.

4.1 Data

We obtained full ad hoc runs submitted to TRECs 3 through 8. Each run ranks at most 1000 documents for 50 topics (49 topics for TREC 4). Additionally, we obtained all runs from the Web track of TREC 13, the Robust[2] track of TREC 14, and the Terabyte (ad hoc) track of TREC 14. These are the tracks that have "replaced" the ad hoc track since its end in 1999. Statistics are shown in Table 1.

We set aside the TREC 4 (ad hoc 95) set for training, TRECs 3 and 5–8 (ad hoc 94 and 96–99) for primary testing, and the remaining sets for additional testing.

We use the *qrels* files assembled by NIST as "truth". The number of relevance judgments made and relevant documents found for each track are listed in Table 1.

For computational reasons, we truncate ranked lists at 100 documents. There is no reason that we could not go deeper, but calculating variance is $\mathcal{O}(n^3)$ and thus very time-consuming. Because of the reciprocal rank nature of AP, we do not lose much information by truncating at rank 100.

4.2 Algorithms

We will compare three algorithms for acquiring relevance judgments. The baseline is a variation of TREC pooling that we will call *incremental pooling* (IP). This algorithm takes a number k as input and presents the first k documents in rank order (without regard to topic) to be judged. It does not estimate the relevance of unjudged documents; it simply assumes any unjudged document is nonrelevant.

The second algorithm is that presented in Carterette et al. [8] (Algorithm 1). Documents are selected based on how "interesting" they are in determining whether a difference in mean average precision exists. For this approach $p_i = 0.5$ for all i; there is no estimation of probabilities. We will call this MTC for *minimal test collection*.

The third algorithm augments MTC with updated estimates of probabilities of relevance. We will call this RTC for *robust test collection*. It is identical to Algorithm 1, except that every 10th iteration we estimate p_i for all unjudged documents i using the expert aggregation model of Section 3.

RTC has smoothing (prior distribution) parameters that must be set. We trained using the ad hoc 95 set. We limited

---

[2] "Robust" here means robust retrieval; this is different from our goal of robust evaluation.

Algorithm 1 (MTC) Given two ranked lists and confidence level α, predict the sign of ΔMAP.

1: $p_i \leftarrow 0.5$ for all documents i
2: **while** $P(\Delta MAP < 0) < \alpha$ **do**
3:     calculate weight w_i for all unjudged documents i
        (see Carterette et al. [8] for details)
4:     $j \leftarrow \operatorname{argmax}_i w_i$
5:     $x_j \leftarrow 1$ if document j is relevant, 0 otherwise
6:     $p_j \leftarrow x_j$
7: **end while**

---

the search to uniform priors with relatively high variance. For expert aggregation, the prior parameters are $\alpha = \beta = 1$.

4.3 Experimental Design

First, we want to know whether we can augment a set of relevance judgments with a set of relevance probabilities in order to reuse the judgments to evaluate a new set of systems. For each experimental trial:

1. Pick a random subset of k runs.
2. From those k, pick an initial $c < k$ to evaluate.
3. Run RTC to 95% confidence on the initial c.
4. Using the model from Section 3, estimate the probabilities of relevance for all documents retrieved by all k runs.
5. Calculate $\mathcal{E}MAP$ for all k runs, and $P(\Delta MAP < 0)$ for all pairs of runs.

We do the same for MTC, but omit step 4. Note that after evaluating the first c systems, we make no additional relevance judgments.

To put our method to the test, we selected $c = 2$: we will build a set of judgments from evaluating only two initial systems. We will then generalize to a set of $k = 10$ (of which those two are a subset).

As we run more trials, we obtain the data we need to test all three of our hypotheses.

4.4 Experimental Evaluation

Recall that a set of judgments is *robust* if the accuracy of the predictions it makes is at least its estimated confidence. One way to evaluate robustness is to bin pairs by their confidence, then calculate the accuracy over all the pairs in each bin. We would like the accuracy to be no less than the lowest confidence score in the bin, but preferably higher.

Since summary statistics are useful, we devised the following metric. Suppose we are a bookmaker taking bets on whether $\Delta MAP < 0$. We use RTC or MTC to set the odds $O = \frac{P(\Delta MAP < 0)}{1 - P(\Delta MAP < 0)}$. Suppose a bettor wagers \$1 on $\Delta MAP \geq 0$. If it turns out that $\Delta MAP < 0$, we win the dollar. Otherwise, we pay out O. If our confidence estimates are perfectly accurate, we break even. If confidence is greater than accuracy, we lose money; we win if accuracy is greater than confidence.

Counterintuitively, the most desirable outcome is breaking even: if we lose money, we cannot trust the confidence estimates, but if we win money, we have either underestimated confidence or judged more documents than necessary. However, the cost of not being able to trust the confidence estimates is much higher than the cost of extra relevance judgments, so we will treat positive outcomes as "good".

The amount we win on each pairwise comparison i is:

$$W_i = y_i - (1 - y_i)\frac{P_i}{1 - P_i} = \frac{y_i - P_i}{1 - P_i}$$

$y_i = 1$ if $\Delta MAP < 0$ and 0 otherwise, and $P_i = P(\Delta MAP < 0)$. The summary statistic is $\overline{W}$, the mean of W_i.

Note that as P_i increases, the more we lose for being wrong. This is as it should be: the penalty should be great for missing the high probability predictions. However, since our losses grow without bound as probabilities approach 1, we cap $-W_i$ at 100.

For our hypothesis that RTC requires fewer judgments than MTC, we are interested in the number of judgments needed to reach 95% confidence on the first pair of systems. The median is more interesting than the mean: most pairs require a few hundred judgments, but a few pairs require several thousand. The distribution is therefore highly skewed, and the mean strongly affected by those outliers.

Finally, for our hypothesis that RTC is more accurate than MTC, we will look at Kendall's τ correlation between a ranking of k systems by a small set of judgments and the true ranking using the full set of judgments. Kendall's τ, a nonparametric statistic based on pairwise swaps between two lists, is a standard evaluation for this type of study. It ranges from -1 (perfectly anti-correlated) to 1 (rankings identical), with 0 meaning that half of the pairs are swapped. As we touched on in the introduction, though, an accuracy measure like rank correlation is not a good evaluation of reusability. We include it for completeness.

4.4.1 Hypothesis Testing

Running multiple trials allows the use of statistical hypothesis testing to compare algorithms. Using the same sets of systems allows the use of paired tests.

As we stated above, we are more interested in the median number of judgments than the mean. A test for difference in median is the Wilcoxon sign rank test. We can also use a paired t-test to test for a difference in mean.

For rank correlation, we can use a paired t-test to test for a difference in τ.

5. RESULTS AND ANALYSIS

The comparison between MTC and RTC is shown in Table 2. With MTC and uniform probabilities of relevance, the results are far from robust. We cannot reuse the relevance judgments with much confidence. But with RTC, the results are very robust. There is a slight dip in accuracy when confidence gets above 0.95; nonetheless, the confidence predictions are trustworthy. Mean W_i shows that RTC is much closer to 0 than MTC. The distribution of confidence scores shows that at least 80% confidence is achieved more than 35% of the time, indicating that neither algorithm is being too conservative in its confidence estimates. The confidence estimates are rather low overall; that is because we have built a test collection from only two initial systems. Recall from Section 1 that we cannot require (or even expect) a minimum level of confidence when we generalize to new systems.

More detailed results for both algorithms are shown in Figure 2. The solid line is the ideal result that would give $\overline{W} = 0$. RTC is on or above this line at all points until confidence reaches about 0.97. After that there is a slight dip in accuracy which we discuss below. Note that both

	MTC		RTC	
confidence	% in bin	accuracy	% in bin	accuracy
$0.5 - 0.6$	33.7%	61.7%	28.6%	**61.9%**
$0.6 - 0.7$	18.1%	73.1%	20.1%	**76.3%**
$0.7 - 0.8$	10.4%	70.1%	15.5%	**78.0%**
$0.8 - 0.9$	9.4%	69.0%	12.1%	**84.9%**
$0.9 - 0.95$	7.3%	78.0%	6.6%	**93.1%**
$0.95 - 0.99$	17.9%	70.4%	12.4%	**93.4%**
1.0	3.3%	68.3%	4.7%	**98.9%**
$\overline{W}$	-5.34		-0.39	
median judged	251		**235**	
mean τ	0.393		**0.555**	

Table 2: Confidence that $P(\Delta MAP < 0)$ and accuracy of prediction when generalizing a set of relevance judgments acquired using MTC and RTC. Each bin contains over 1,000 trials from the adhoc 3, 5–8 sets. RTC is much more robust than MTC. $\overline{W}$ is defined in Section 4.4; closer to 0 is better. Median judged is the number of judgments to reach 95% confidence on the first two systems. Mean τ is the average rank correlation for all 10 systems.
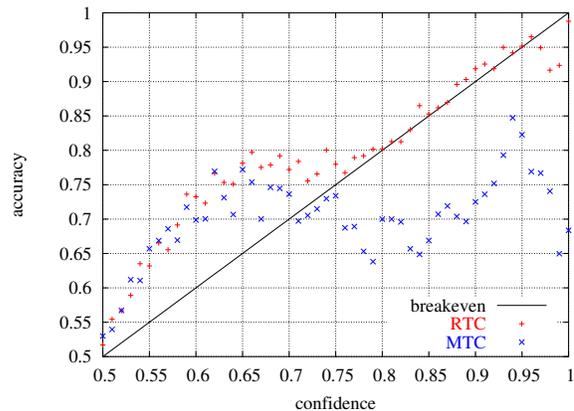

Figure 2: Confidence vs. accuracy of MTC and RTC. The solid line is the perfect result that would give $\overline{W} = 0$; performance should be on or above this line. Each point represents at least 500 pairwise comparisons.

algorithms are well above the line up to around confidence 0.7. This is because the baseline performance on these data sets is high; it is quite easy to achieve 75% accuracy doing very little work [7].

Number of Judgments: The median number of judgments required by MTC to reach 95% confidence on the first two systems is 251, an average of 5 per topic. The median required by RTC is 235, about 4.7 per topic. Although the numbers are close, RTC's median is significantly lower by a paired Wilcoxon test ($p < 0.0001$). For comparison, a pool of depth 100 would result in a minimum of 5,000 judgments for each pair.

The difference in means is much greater. MTC required a mean of 823 judgments, 16 per topic, while RTC required a mean of 502, 10 per topic. (Recall that means are strongly skewed by a few pairs that take thousands of judgments.) This difference is significant by a paired t-test ($p < 0.0001$).

Ten percent of the sets resulted in 100 or fewer judgments (less than two per topic). Performance on these is very high: $\overline{W} = 0.41$, and 99.7% accuracy when confidence is at least 0.9. This shows that even tiny collections can be reusable. For the 50% of sets with more than 235 judgments, accuracy is 93% when confidence is at least 0.9.

Rank Correlation: MTC and RTC both rank the 10 systems by $\mathcal{E}$MAP (Eq. (1)) calculated using their respective probability estimates. The mean τ rank correlation between true MAP and $\mathcal{E}$MAP is 0.393 for MTC and 0.555 for RTC. This difference is significant by a paired t-test ($p < 0.0001$). Note that we do not expect the τ correlations to be high, since we are ranking the systems with so few relevance judgments. It is more important that we estimate confidence in each pairwise comparison correctly.

We ran IP for the same number of judgments that MTC took for each pair, then ranked the systems by MAP using only those judgments (all unjudged documents assumed nonrelevant). We calculated the τ correlation to the true ranking. The mean τ correlation is 0.398, which is not significantly different from MTC, but is significantly lower than RTC. Using uniform estimates of probability is indistinguishable from the baseline, whereas estimating relevance by expert aggregation boosts performance a great deal: nearly 40% over both MTC and IP.

Overfitting: It is possible to "overfit": if too many judgments come from the first two systems, the variance is reduced too much, and the confidence estimates become unreliable. We saw this in Table 2 and Figure 2 where RTC exhibits a dip in accuracy when confidence is around 97%. In fact, the number of judgments made prior to a wrong prediction is over 50% greater than the number made prior to a correct prediction.

Overfitting is difficult to quantify exactly, because making more relevance judgments does not always cause it: at higher confidence levels, more relevance judgments are made, and as Table 2 shows, accuracy is greater at those higher confidences. Obviously having more relevance judgments should increase both confidence and accuracy; the difference seems to be when one system has a great deal more judgments than the other.

Pairwise Comparisons: Our pairwise comparisons fall into one of three groups:

1. the two original runs from which relevance judgments are acquired;
2. one of the original runs vs. one of the new runs;
3. two new runs.

Table 3 shows confidence vs. accuracy results for each of these three groups. Interestingly, performance is worst when comparing one of the original runs to one of the additional runs. This is most likely due to a large difference in the number of judgments affecting the variance of ΔMAP. Nevertheless, performance is quite good on all three subsets.

Worst Case: The case intuitively most likely to produce an error is when the two systems being compared have retrieved very few documents in common. If we want the judgments to be reusable, we should to be able to generalize even to runs that are very different from the ones used to acquire the relevance judgments.

A simple measure of similarity of two runs is the average percentage of documents they retrieved in common for each topic [2]. We calculated this for all pairs, then looked at performance on pairs with low similarity. Results are shown in

	accuracy		
confidence	two original	one original	no original
$0.5 - 0.6$	–	48.1%	62.8%
$0.6 - 0.7$	–	57.1%	79.2%
$0.7 - 0.8$	–	67.9%	81.7%
$0.8 - 0.9$	–	82.2%	86.3%
$0.9 - 0.95$	95.9%	93.7%	92.6%
$0.95 - 0.99$	96.2%	92.5%	93.1%
1.0	100%	98.0%	99.1%
$\overline{W}$	-1.11	-0.87	-0.27

Table 3: Confidence vs. accuracy of RTC when comparing the two original runs, one original run and one new run, and two new runs. RTC is robust in all three cases.

	accuracy when similar		
confidence	$0 - 0.1$	$0.1 - 0.2$	$0.2 - 0.3$
$0.5 - 0.6$	68.4%	63.1%	61.4%
$0.6 - 0.7$	84.2%	78.6%	76.6%
$0.7 - 0.8$	82.0%	79.8%	78.9%
$0.8 - 0.9$	93.6%	83.3%	82.1%
$0.9 - 0.95$	99.3%	92.7%	92.4%
$0.95 - 0.99$	98.7%	93.4%	93.3%
1.0	99.9%	97.9%	98.1%
$\overline{W}$	0.44	-0.45	-0.49

Table 4: Confidence vs. accuracy of RTC when a pair of systems retrieved 0–30% documents in common (broken out into 0%–10%, 10%–20%, and 20%–30%). RTC is robust in all three cases.

Table 4. Performance is in fact very robust even when similarity is low. When the two runs share very few documents in common, $\overline{W}$ is actually positive.

MTC and IP both performed quite poorly in these cases. When the similarity was between 0 and 10%, both MTC and IP correctly predicted ΔMAP only 60% of the time, compared to an 87.6% success rate for RTC.

By Data Set: All the previous results have only been on the ad hoc collections. We did the same experiments on our additional data sets, and broke out the results by data set to see how performance varies. The results in Table 5 show everything about each set, including binned accuracy, $\overline{W}$, mean τ, and median number of judgments to reach 95% confidence on the first two systems. The results are highly consistent from collection to collection, suggesting that our method is not overfitting to any particular data set.

6. CONCLUSIONS AND FUTURE WORK

In this work we have offered the first formal definition of the common idea of "reusability" of a test collection and presented a model that is able to achieve reusability with very small sets of relevance judgments. Table 2 and Figure 2 together show how biased a small set of judgments can be: MTC is dramatically overestimating confidence and is much less accurate than RTC, which is able to remove the bias to give a robust evaluation.

The confidence estimates of RTC, in addition to being accurate, provide a guide for obtaining additional judgments: focus on judging documents from the lowest-confidence comparisons. In the long run, we see small sets of relevance judg-

confidence	accuracy							
	ad hoc 94	ad hoc 96	ad hoc 97	ad hoc 98	ad hoc 99	web 04	robust 05	terabyte 05
$0.5 - 0.6$	64.1%	61.8%	62.2%	62.0%	59.4%	64.3%	61.5%	61.6%
$0.6 - 0.7$	76.1%	77.8%	74.5%	78.2%	74.3%	78.1%	75.9%	75.9%
$0.7 - 0.8$	75.2%	78.9%	77.6%	80.0%	78.6%	82.6%	77.5%	80.4%
$0.8 - 0.9$	83.2%	85.5%	84.6%	84.9%	86.8%	84.5%	86.7%	87.7%
$0.9 - 0.95$	93.0%	93.6%	92.8%	93.7%	92.6%	94.2%	93.9%	94.2%
$0.95 - 0.99$	93.1%	94.3%	93.1%	93.7%	92.8%	95.0%	93.9%	91.6%
1.0	99.2%	96.8%	98.7%	99.5%	99.6%	100%	99.2%	98.3%
$\overline{W}$	-0.34	-0.34	-0.48	-0.35	-0.44	-0.07	-0.41	-0.67
median judged	235	276	243	213	179	448	310	320
mean τ	0.538	0.573	0.556	0.579	0.532	0.596	0.565	0.574

Table 5: Accuracy, $\overline{W}$, mean τ, and median number of judgments for all 8 testing sets. The results are highly consistent across data sets.

ments being shared by researchers, each group contributing a few more judgments to gain more confidence about their particular systems. As time goes on, the number of judgments grows until there is 100% confidence in every evaluation—and there is a full test collection for the task.

We see further use for this method in scenarios such as web retrieval in which the corpus is frequently changing. It could be applied to evaluation on a dynamic test collection as defined by Soboroff [18].

The model we presented in Section 4 is by no means the only possibility for creating a robust test collection. A simpler expert aggregation model might perform as well or better (though all our efforts to simplify failed). In addition to expert aggregation, we could estimate probabilities by looking at similarities between documents. This is an obvious area for future exploration.

Additionally, it will be worthwhile to investigate the issue of overfitting: the circumstances it occurs under and what can be done to prevent it. In the meantime, capping confidence estimates at 95% is a "hack" that solves the problem.

We have many more experimental results that we unfortunately did not have space for but that reinforce the notion that RTC is highly robust: with just a few judgments per topic, we can accurately assess the confidence in any pairwise comparison of systems.

Acknowledgments

7. REFERENCES

[1] J. Aslam and M. Montague. Models for Metasearch. In *Proceedings of SIGIR*, pages 275–285, 2001.

[2] J. Aslam and R. Savell. On the effectiveness of evaluating retrieval systems in the absence of relevance judgments. In *Proceedings of SIGIR*, pages 361–362, 2003.

[3] J. A. Aslam, V. Pavlu, and R. Savell. A unified model for metasearch, pooling, and system evaluation. In *Proceedings of CIKM*, pages 484–491, 2003.

[4] J. A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *Proceedings of SIGIR*, pages 541–548, 2006.

[5] A. L. Berger, S. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[6] D. J. Blower. An easy derivation of logistic regression from the bayesian and maximum entropy perspective. In *Proceedings of the 23rd International Workship on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, pages 30–43, 2004.

[7] B. Carterette and J. Allan. Research methodology in studies of assessor effort for retrieval evaluation. In *Proceedings of RIAO*, 2007.

[8] B. Carterette, J. Allan, and R. K. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of SIGIR*, pages 268–275, 2006.

[9] B. Carterette and D. I. Petkova. Learning a ranking from pairwise preferences. In *Proceedings of SIGIR*, 2006.

[10] R. T. Clemen and R. L. Winkler. Unanimity and compromise among probability forecasters. *Management Science*, 36(7):767–779, July 1990.

[11] G. V. Cormack, C. R. Palmer, and C. L. Clarke. Efficient Construction of Large Test Collections. In *Proceedings of SIGIR*, pages 282–289, 1998.

[12] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2004.

[13] E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[14] R. Manmatha and H. Sever. A Formal Approach to Score Normalization for Metasearch. In *Proceedings of HLT*, pages 88–93, 2002.

[15] I. J. Myung, S. Ramamoorti, and J. Andrew D. Baily. Maximum entropy aggregation of expert predictions. *Management Science*, 42(10):1420–1436, October 1996.

[16] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. pages 61–74, 2000.

[17] M. Sanderson and H. Joho. Forming test collections with no system pooling. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, 2004.

[18] I. Soboroff. Dynamic test collections: measuring search effectiveness on the live web. In *Proceedings of SIGIR*, pages 276–283, 2006.

[19] K. Sparck Jones and C. J. van Rijsbergen. Information Retrieval Test Collections. *Journal of Documentation*, 32(1):59–75, 1976.

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

[21] J. Zobel. How Reliable are the Results of Large-Scale Information Retrieval Experiments? In *Proceedings of SIGIR*, pages 307–314, 1998.