Research Methodology in Studies of Assessor Effort for Information Retrieval Evaluation

Ben Carterette & James Allan

Center for Intelligent Information Retrieval
Computer Science Department
140 Governors Drive
University of Massachusetts Amherst
Amherst, MA 01003
{carteret, allan}@cs.umass.edu

Abstract

As evaluation is an important but difficult part of information retrieval system design and experimentation, evaluation questions have been the subject of much research. An "evaluation study" is an investigation into some aspect of evaluation. These types of studies typically experiment on ranked results from actual retrieval systems, most often those that were submitted to TREC tracks. We argue that the standard of evidence in these types of studies should be increased to the level required of text retrieval studies, by testing on multiple data sets, multiple subsets of data, and comparison to baselines using hypothesis testing. We demonstrate that baseline performance on the standard data sets is quite high, necessitating strong evidence to support claims.

1 Introduction

Evaluation is a difficult but important problem in information retrieval: choices of retrieval task, test collection, evaluation measure, and definition of relevance can all have a significant effect on conclusions drawn from an evaluation. There has thus been a great deal of work studying these issues. We refer to these works as *evaluation studies*. Under this broad heading we include studies of evaluation measures, studies of hypothesis tests, and studies of low-cost retrieval evaluation. The typical experimental methodology in these studies uses retrieval results that were submitted to TREC (**T**ext **RE**trieval **C**onference). These are real retrieval systems used for research and commercial purposes, so it makes sense to test evaluation questions against them. However, they have not been studied well enough to really understand what is going on when testing on them. As a result, we believe that the results of evaluation studies should not be accepted out of context, but should be rigorously compared to other results on the same data.

To illustrate our point, we make an analogy to *tf-idf* term weighting in text retrieval. We know that tf-idf is a decent way to weight terms, even though modeling a document as a vector of tf-idf weights loses a lot information. But *how* do we know this? If we were retrieval novices and performed one experiment on one corpus and measured an average precision of 0.2, would we believe that tf-idf was any good? But if we did multiple experiments with multiple retrieval models on multiple corpora and saw that the performance of tf-idf was seldom far from the other models no matter what the corpus, we would be much more receptive to the idea.

Scientific results need strong support in order to be accepted. In the information retrieval research community, for a result to gain broad acceptance, it must have been tested on multiple corpora, compared to strong baselines, and shown to be statistically significant. We argue that evaluation studies should be held to the same standard of evidence.

This is especially true for the types of studies we are considering. The research corpora used in text retrieval studies typically comprise documents that users would actually search; research topics come from actual users of retrieval systems. They are representative of a certain sample space. It is less clear that the retrieval runs submitted to TREC are as representative. Many of the systems submitted to TREC are experimental; a handful have serious bugs. Some of them involve human adjustments. It is therefore important to show that performance is not simply an artifact of the data.

Furthermore, the data sets typically used to evaluate meta-evaluation studies are "easy": baseline performance is quite high. A small number of relevance judgments can achieve better results than we have any right to expect. This is similar to the situation in text retrieval before TREC, when the available corpora (such as CACM) had high performance baselines.

This paper is structured as follows: we first present the previous work that has led us to this study and the data typically used in studies like this one. We then present an algorithm based on pairwise preferences that ranks retrieval systems with no relevance judgments. The results of this algorithm suggest that baseline performance is very high, leading us to analyze the data sets to find out why it is so high. We then present an algorithm that illustrates that good results can be achieved almost by accident, and argue that the solution to the problem is to argue about algorithms using formal proof and hypothesis testing.

2 Previous Work

While there have been numerous studies on the evaluation of information retrieval tasks, there have been none (to our knowledge) on *meta-evaluation*: the evaluation of studies on evaluation of information retrieval tasks. There are generally-accepted meta-evaluations, such as Kendall's τ correlation, and of course we do not claim that these studies completely lack evaluation. However, many of the evaluation studies in the literature are roughly comparable to evaluating one or two retrieval experiments on a corpus like CACM.

Evaluation studies can be seen as falling into three broad categories: studies of evaluation metrics, studies of hypothesis testing, and studies of assessor effort. In this work we are concerned exclusively with the latter.

Our work is inspired by the results of Soboroff et al. (2001). They showed that simply by taking a random set of documents from retrieval systems to be "relevant", one could obtain a fairly good approximation to the evaluation obtained when all relevance judgments are known. We refer to this as "no-cost" evaluation since it does not require any assessor effort at all. On the surface this seems rather surprising: we should naively expect that randomly assigning relevance to documents will result in no correlation between predicted and true performance. But Aslam et al. (2003) showed that this algorithm was in fact rewarding the systems that retrieved the most popular documents. Knowing that TREC systems tend to retrieve more relevant documents in common than nonrelevant documents Lee (1997), Soboroff et al.'s result is less surprising.

Another inspiration is the work of Buckley and Voorhees (2004) introducing the *bpref* measure of performance. For one thing, much of the work we will do is couched in terms of pairwise preferences of documents, which is what bpref is based on. For another, their results suggest something similar to Soboroff et al.'s work: that there is some property of the data sets used for

testing that make it possible to accurately evaluate them with very few relevance judgments. In other words, these two works suggest that the baseline performance on these data sets is quite a bit higher than no correlation.

Finally, a table in Carterette and Allan (2005) shows that simply judging a pool of very shallow depth (as little as one document per system per topic) results in a positive and significant correlation between predicted performance and true performance for one of the standard data sets. This result translates to the other data sets as well.

These three results together suggest that the baseline performance for these data sets is quite high. These types of studies often make an implicit assumption that the baseline correlation between predicted evaluation and true evaluation should be 0 when no relevance judgments are available. In fact, the baseline is much higher, and this affects the amount of evidence that must be shown in order to draw conclusions. Showing that a particular method gives a high correlation between predicted evaluation and true evaluation, then, is not good enough; that is essentially presenting the results out of the context of the data sets they were tested on. It is because of that that we recommend that more rigorous standards of evidence be required in evaluation studies.

There has been work similar to ours on evaluation of retrieval systems. We recommend using hypothesis tests to evaluate an algorithm or retrieval metric; the use of hypothesis tests in retrieval evaluation has been studied in works including those by van Rijsbergen (1979), Savoy (1997), Hull (1993), and Sanderson and Zobel (2005). Furthermore, it is well known in IR that some corpora are "easier" than others, and as a result it is standard to test on multiple corpora. We argue that the data sets typically used in evaluation studies are "easy" and therefore more rigorous testing is needed.

Other evaluation studies that led to this work include that of Cormack et al. (1998), who introduce two algorithms for acquiring relevance judgments: "Interactive Searching and Judging" (ISJ) and "Move-to-Front Pooling" (MTF). Zobel (1998) questioned whether the relevance judgments formed at TREC are sufficient, and finding that they are, showed that in fact similar results can be achieved with many fewer judgments. Sanderson and Joho (2004) showed that accurate results could be achieved by judging the documents retrieved by one system—again reinforcing the idea that there is some property of these particular data sets that makes these results possible. Aslam et al. (2005; 2006) have presented two low-cost algorithms. Most recently, Carterette et al. (2006) presented an algorithm for acquiring relevance judgments that is optimal. We do not claim that the algorithms presented in these works are wrong; in fact, we believe the arguments are sound. We only claim that studies of this type should be tested with the same rigor as text retrieval studies.

There has certainly been some previous work that presented very convincing evidence. Some model studies include Zobel (1998), Voorhees (1998), and Buckley and Voorhees (2000). One of our contributions above the experimental methodology used in those papers is to show how hypothesis tests can be applied to studies such as these.

3 Data Sets

At TREC (the **T**ext **RE**treival **C**onference), participating sites submit retrieval runs over provided corpora. There are a wide variety of tracks, including ad hoc, robust, HARD, web, terabyte, and many more. Retrieval runs from each conference are archived and available for experimental evaluation. This gives the community a large number of real retrieval system results to work with for evaluation studies or data fusion studies.

The data sets we used are the ones nearly uniformly used in evaluation and data fusion studies: the sets of retrieval runs that were submitted to the TREC ad hoc tracks from 1994 through 1999

TREC	topics	no. runs	no. manual	docs per topic	rel per topic
3	151-200	40	11	1009.4	146.9
4	202-250	33	19	1436.2	109.7
5	251-300	61	31	1620.5	100.8
6	301-350	74	17	2200.5	88.0
7	351-400	103	17	2029.1	92.4
8	401-450	129	12	2335.5	94.2

Table 1: Number of runs, number of manual runs, average number of unique documents retrieved per topic (in the top 100), and average number of unique relevant documents retrieved per topic for each TREC ad hoc collection.

(TRECs 3 through 8). Each run includes ranked lists for all 50 topics used for evaluation that year (except for TREC-4, which used 49 topics instead of 50). Some of the runs are fully automatic; in these the only interaction between the system and a user is the submission of a pre-determined query to the system. Some of the runs are "manual", meaning that a user interacted with the system in some way, be it by reformulating the query, iteratively searching on different queries, or providing feedback to the system. Some statistics of these sets are shown in Table 1.

The data sets also include relevance judgments for each topic. These are binary indicators of the relevance of a document to the topic. Most of the documents retrieved by each run have been judged, so these judgments can be used to compute the "true" values of our evaluation measures.

4 Evaluation Without Relevance Judgments

Soboroff et al. (2001) showed that simply taking a random set of documents to be relevant could give results that correlate positively and significantly with true rankings. The following three subsections are devoted to presenting a more formal method for doing the same thing. Our algorithm is based on pairwise preferences of documents, which we will use to analyze the results.

4.1 Estimating Evaluation Metrics

Precision is simply the proportion of relevant documents retrieved by a certain rank. Precision is quite coarse: a precision of 0.3 at rank 10 could mean that the top 3 documents are relevant (which would be good), or that the documents at ranks 8, 9, and 10 are relevant (much less good). A good evaluation metric should reward systems not only for retrieving relevant documents but also for ranking them highly.

Average precision does this by looking at precision at several points in the ranked list. Specifically, it is an average of the precision at each rank that a relevant document appears at. For example, consider a ranking of three documents A, B, C. Suppose A and C are relevant. The precision at A is 1; the precision at C is $\frac{2}{3}$. The average precision is the average of those two numbers: $\frac{5}{6}$. Average precision ranges from 0 to 1, with 1 the best possible for any set of relevance judgments.

What if we have no relevance judgments? Obviously we cannot compute average precision. But we can say what values it *could* take: for our ranking of three documents above, depending on how the documents are judged, average precision could be $0, \frac{1}{3}, \frac{5}{12}, \frac{1}{2}, \frac{5}{6}$, or 1. If we then judge document A relevant, we know AP could not be 0, $\frac{1}{3}$, $\frac{5}{12}$, or $\frac{1}{2}$; it must be either $\frac{5}{6}$ or 1, depending on how B and C are judged.

This example illustrates the idea of treating an evaluation measure such as AP as a random variable over judgments of relevance, introduced by Carterette et al. (2006). Following that work, we formalize this as follows: first, let R be the set of judged relevant documents, and $r(i)$ be the rank at which document i was retrieved. Then

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

Let x_i be the relevance of the document at rank i. Our definition of relevance follows TREC's, and is a binary judgment: $x_i = 1$ if i is relevant, 0 otherwise. Then

$$\begin{aligned} prec@i &= \frac{1}{i} \sum_{j=1}^{i} x_j \\ AP &= \frac{1}{\sum_{i=1}^{n} x_i} \sum_{i=1}^{n} x_i \frac{1}{i} \sum_{j=1}^{i} x_j \\ &= \frac{1}{\sum x_i} \sum_i \sum_{j \le i} \frac{1}{i} x_i x_j \end{aligned}$$

where n is the total number of documents in the corpus.

If a document has not been judged, x_i is unknown. Let X_i be a Bernoulli random variable indicating the relevance of document i. Rewriting AP as a function of random variables X_i:

$$AP = \frac{1}{\sum X_i} \sum_i \sum_{j \le i} \frac{1}{i} X_i X_j$$

we can see that average precision itself is a random variable with a distribution over possible judgments of relevance. We make one more modification at this point: rather than index documents by rank, we want to be able to index them arbitrarily. The resulting expression is:

$$AP = \frac{1}{\sum X_i} \sum_i \sum_{j \le i} \frac{1}{\max\{r(i), r(j)\}} X_i X_j.$$

Let $p_i = P(X_i = 1)$, i.e. p_i is the probability that document i is relevant. For our example above, we might say $p_A = \frac{4}{5}, p_B = \frac{2}{5}, p_C = \frac{3}{5}$ (these numbers are chosen arbitrarily). These probabilities determine how probable each value of AP is: $P(AP = 0|p_A, p_B, p_C) = \frac{1}{5} \cdot \frac{3}{5} \cdot \frac{2}{5} = 0.048$, $P(AP = 1|p_A, p_B, p_C) = \frac{4}{5} \cdot \frac{2}{5} \cdot \frac{3}{5} + \frac{4}{5} \cdot \frac{2}{5} \cdot \frac{2}{5} + \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{2}{5} = .512$, and so on.

Expectation of AP is a sum over exponentially many terms, but we can approximate it in an intuitive way with the following expression:

$$E[AP] = \frac{1}{\sum p_i} \sum_{i=1}^{n} \left(\frac{1}{r(i)} p_i + \sum_{j=1}^{i-1} \frac{1}{\max\{r(i), r(j)\}} p_i p_j \right) + \epsilon \tag{1}$$

The error in the approximation is represented by ϵ, which is a negligible $\mathcal{O}(2^{-n})$. We ignore it for the remainder of this work.

Mean average precision (MAP) is simply the average of a set of average precisions calculated for each topic in a set T. The expectation of MAP follows directly from the expectation of AP:

$$\mathcal{E}MAP = \frac{1}{|T|} \sum_{t \in T} E[AP_t] \tag{2}$$

where AP_t denotes the average precision for topic t.

Calculating $\mathcal{E}$MAP requires choosing a probability of relevance for each document. Carterette et al. (2006) simply used a uniform $p_i = 0.5$ for all documents; in that case, all systems have the same $\mathcal{E}$MAP when no judgments are available. If probabilities are assigned non-uniformly by some algorithm, it becomes possible to rank systems with no judgments at all. The next section describes an expert aggregation algorithm for assigning probabilities of relevance.

4.2 Finding a Consensus Among Experts

We shall treat a ranked list as an information retrieval "expert" that is providing pairwise preferences of documents. For example, if ranked list ℓ has ranked document i above document j, we say expert ℓ prefers i to j, denoted $i \succ_\ell j$. It then follows that $i \succ_\ell j \Rightarrow r_\ell(i) < r_\ell(j)$.

Probabilistic methods for combining experts take expert opinions on events (usually expressed as probabilities) and compute a "consensus" probability of that event. Our events are pairs of documents i, j; the experts' opinions are whether $i \succ j$. Carterette and Petkova (2006) presented a maximum likelihood model for estimating the relevance of documents from expert opinions; though the application in that work was to metasearch, there is a connection between metasearch and evaluation (Aslam et al., 2003) that we can take advantage of to apply the same method to our problem.

We will find a consensus by maximizing the likelihood of observing all the pairwise preferences expressed by all the experts. Let n_{ij} be the number of experts that expressed the preference $i \succ j$. We wish to find $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$, a vector of "relevance weights" indexed in the same order as the documents, each of which indicates our belief that the corresponding document is relevant. Then the likelihood function is:

$$\begin{aligned}\mathcal{L}(\Theta) &= \prod_i \prod_j P(i \succ j | \theta_i, \theta_j)^{n_{ij}} \\ &= \prod_i \prod_j P(X_i > X_j | \theta_i, \theta_j)^{n_{ij}}\end{aligned}$$

If we define the log-odds of $P(X_i > X_j|\theta_i, \theta_j)$ to be a linear function:

$$\log \frac{P(X_i > X_j|\theta_i, \theta_j)}{1 - P(X_i > X_j|\theta_i, \theta_j)} = \theta_i - \theta_j \tag{3}$$

then maximizing $\mathcal{L}$ over Θ is equivalent to solving a logistic regression with variables equal to the number of documents and an instance for each of the pairwise preferences from all systems. The parameters Θ are then a measure of the relevance of each document, and

$$P(X_i = 1) = \frac{\exp(\theta_i)}{1 + \exp(\theta_i)} \tag{4}$$

This is somewhat similar to Joachims's *ranking SVM* (Joachims, 2002), except that the binary class labels are determined by the sign of the difference in rank rather than clickthrough counts, and the only feature is a binary feature indicating which document is under consideration.

4.2.1 Computational Issues

One potential problem with this model is that a parameter could grow without bound. If document i is preferred to all other documents by every expert (i.e. ranked first by every system), the likelihood has no maximum: as $\theta_i \to \infty$, $\mathcal{L} \to \infty$. The remaining parameters become irrelevant to the maximization; we cannot expect them to have any meaning.

To solve this, we follow Mease (2003) in introducing a prior for each document. Let $\xi_i = P(X_i = 1)$. Using the conjugate prior simplifies computation over using a Gaussian or other standard priors (Gelman et al., 2004). Since X_i is a Bernoulli trial, its conjugate prior is a Beta distribution: $\xi_i \sim Beta(\alpha, \beta)$. ξ_i acts as a penalization that keeps θ_i from increasing without bound. The likelihood function then becomes:

$$\mathcal{L}(\Theta) = \prod_i \prod_j P(X_i > X_j | \theta_i, \theta_j)^{n_{ij}} \prod_i \xi_i^{\alpha} (1 - \xi_i)^{\beta} \tag{5}$$

In the absence of any information about relevance, a reasonable choice of α and β is $\alpha = \beta = 1$; this is the uniform (noninformative) prior. This can be seen as introducing a "dummy" document and a set of preferences for which it is preferred to every other document and every other document is preferred to it.

Since the Beta distribution is the conjugate prior for the Bernoulli distribution, we can "update" the priors each time a document is judged. We simply increment α if the judged document is relevant, or β if the judged document is nonrelevant. We can then think of ξ_i as a Laplacian-smoothed topic prior, with $E[\xi_i] = \frac{|R|+1}{|R|+|N|+1}$, where $|R|$ is the number of judged relevant documents and $|N|$ is the number of judged nonrelevant documents.

The second computational issue is implementation of a maximization algorithm that can handle thousands of variables and millions of training instances. We used iteratively reweighted least squares (IRLS), using the conjugate gradient descent algorithm described by Komarek and Moore (2005). By taking advantage of our simple data to precompute matrices, we are able to solve the maximization problem very fast: for one set of 2.5 million preferences and 800 documents, the likelihood was maximized in about 3 seconds. We have made our code for this available at `http://ciir.cs.umass.edu/~carteret`.

4.2.2 Evaluating Probability Estimates

The expert aggregation model is used to infer a probability of relevance for each document. To evaluate the probabilities, we compare them to the actual relevance of the document. In Table 2, documents are separated into bins by their inferred probability of relevance. For each bin, we compute the percentage of documents in the bin that are relevant. If the probability estimates are good, the percentages should be within the bin boundaries, e.g. if the bin consists of all documents with probability of relevance between 0.7 and 0.8, we would like to see at least 70% of the documents in the bin be relevant. Since, as Table 2 shows, the percentages are not within the bin boundaries for any bin or any collection, the probability estimates do not appear to be very good. The goodness-of-fit statistics R^2 and deviance, both of which would be 1 if the predictions were perfect, confirm that probabilities are not very good.

However, the relevance percentages tend to increase as probability increases. As we will see in the next section, this is good enough to ensure a fairly accurate ranking of systems.

probability interval	percent relevant TREC-3	TREC-4	TREC-5	TREC-6	TREC-7	TREC-8
$[0.0, 0.1)$	–	–	–	–	–	–
$[0.1, 0.2)$	–	–	–	–	–	–
$[0.2, 0.3)$	0.104	–	0.016	0.042	0.019	0.003
$[0.3, 0.4)$	0.059	0.039	0.026	0.012	0.013	0.011
$[0.4, 0.5)$	0.134	0.029	0.061	0.012	0.035	0.024
$[0.5, 0.6)$	0.109	0.076	0.052	0.037	0.028	0.025
$[0.6, 0.7)$	0.189	0.113	0.067	0.055	0.060	0.042
$[0.7, 0.8)$	0.233	0.164	0.107	0.076	0.074	0.070
$[0.8, 0.9)$	0.322	0.274	0.141	0.109	0.101	0.102
$[0.9, 1.0]$	0.545	0.469	0.293	0.255	0.243	0.233
R^2	0.123	0.120	0.066	0.079	0.085	0.096
deviance	0.118	0.143	0.100	0.148	0.149	0.172

Table 2: Evaluation of the probability estimates produced by our maximum-likelihood pairwise preference method. R^2 and dev are measures of the correlation between probability and relevance.

4.3 Ranking Retrieval Systems With No Relevance Judgments

The above expert aggregation model produces probabilities of document relevance solely from the pairwise preferences expressed by each expert; it requires no relevance judgments. Plugging these probability estimates into Eq. 1 for each topic gives us $\mathcal{E}$MAP which we can then use to rank the systems with no judgments.

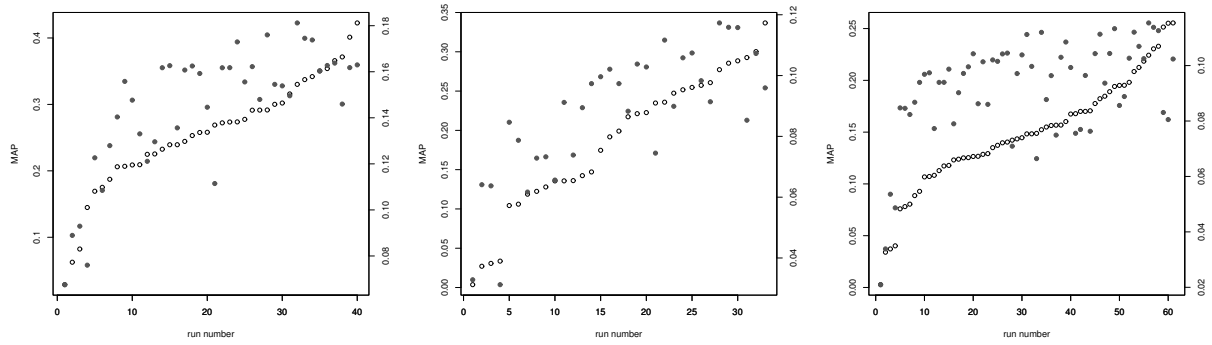
To evaluate the ranking by $\mathcal{E}$MAP , we compare it to the "true" ranking obtained by evaluating each system using the supplied NIST judgments. Kendall's τ rank correlation is the standard measure in evaluation studies for comparing rankings of systems. Kendall's τ ranges from -1 to 1, with 1 indicating a perfect correlation and -1 indicating perfect anti-correlation (the ranked lists are inverted). The τ correlation is based on pairwise swaps, so $\tau = 0$ means that 50% of pairs were swapped between rankings, $\tau = 0.5$ means that 25% of pairs were swapped, and $\tau = -0.5$ means that 75% of pairs were swapped.

Figure 1 shows the true ranking by MAP and estimated ranking by $\mathcal{E}$MAP for each of our six ad hoc collections. The rankings are quite good; the τ correlations are positive and significant. The errors are almost entire due to the performance of a handful of systems being dramatically underestimated.

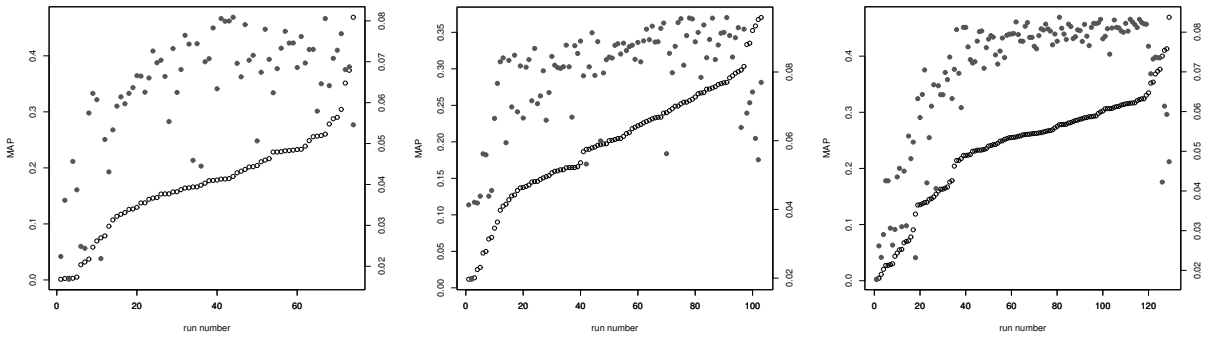
Recall from Section 2 that Soboroff et al. (2001) ranked systems with no relevance judgments by assigning relevance to a random subset of retrieved documents. In one experiment, Soboroff et al. took a random sample of documents from a pool with no duplicates; in another, each document was duplicated in the pool according to the number of systems it was retrieved by. Our results are compared to both of these experiments in Table 3.

4.4 Reweighting Manual Runs

In Figures 1(c)–1(f), there are systems for which $\mathcal{E}$MAP dramatically underestimates the true performance. These are uniformly manual runs. Manual runs are known to retrieve relevant documents that were not identified by automatic runs. Since manual runs are less well-represented in the set, and since they are retrieving some different documents, the documents they retrieve tend



(a) TREC-3 ($\tau = 0.528$) (b) TREC-4 ($\tau = 0.591$) (c) TREC-5 ($\tau = 0.355$)



(d) TREC-6 ($\tau = 0.425$) (e) TREC-7 ($\tau = 0.416$) (f) TREC-8 ($\tau = 0.500$)

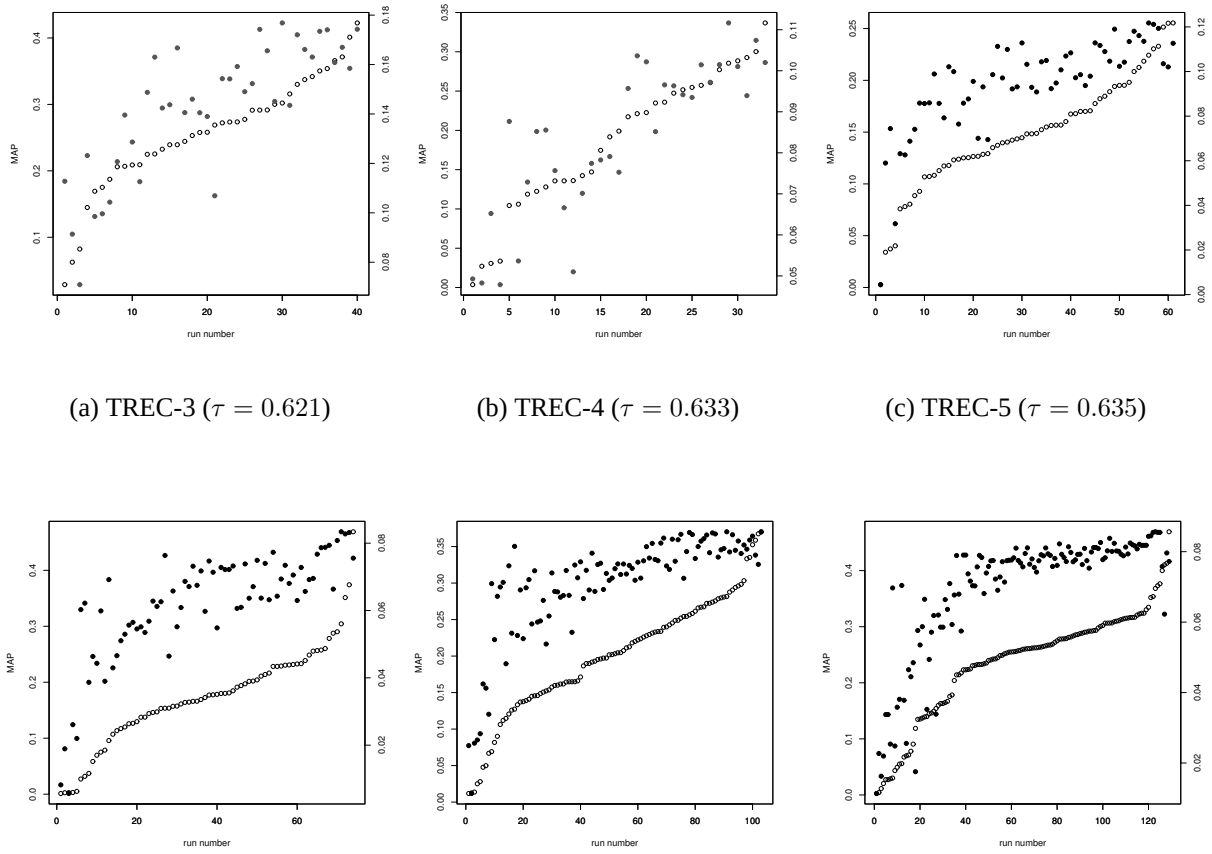
Figure 1: Ranking retrieval systems with no relevance judgments. Hollow circles show the "true" ranking by MAP (labeled on the left axis); filled circles show the corresponding $\mathcal{E}$MAP for each system (labeled on the right axis).

	TREC-3	TREC-4	TREC-5	TREC-6	TREC-7	TREC-8
preferences	0.528	0.591	0.355	0.425	0.416	0.500
no dups	0.430	–	0.487	0.408	0.369	0.459
dups	0.482	–	0.571	0.491	0.423	0.534

Table 3: Ranking retrieval systems without relevance judgments: τ correlations with pairwise preferences (top row) compared to τ correlations reported by Soboroff et al. (2001) (bottom two rows). "No dups" shows results when documents are not duplicated in the pool; "dups" shows results when they are.

to be "overlooked" by the pairwise preference algorithm. To account for this, we can interject our own knowledge and manually reweight the manual runs.

We take the penalized likelihood function Eq. 5 and replace the raw count n_{ij} with a weighted count $n'_{ij} = \sum_\ell w_\ell y_{ij}$, where w_ℓ is the weight given to expert ℓ and $y_{ij} = 1$ if $i \succ_\ell j$. This gives us



(a) TREC-3 ($\tau = 0.621$) (b) TREC-4 ($\tau = 0.633$) (c) TREC-5 ($\tau = 0.635$)

(d) TREC-6 ($\tau = 0.635$) (e) TREC-7 ($\tau = 0.666$) (f) TREC-8 ($\tau = 0.691$)

Figure 2: Weighting manual runs 8 to 16 times higher than automatic runs improves correlations dramatically over Figure 1.

the weighted likelihood function

$$\mathcal{L}(\Theta) = \prod_i \prod_j P(X_i > X_j | \theta_i, \theta_j)^{n'_{ij}} \prod_i \xi_i^{\alpha} (1 - \xi_i)^{\beta} \tag{6}$$

which effectively duplicates the preferences professed by each expert ℓ w_ℓ times.

The result of weighting manual runs 8 to 16 times higher than automatic runs is shown in Figure 2. The correlations improve dramatically for all sets.

5 Analysis

Our analysis of these results shall use a measure of similarity between ranked lists based on the pairwise preferences they share. Let $\mathcal{P}_{\ell_i}$ be the set of pairwise preferences expressed by expert ℓ_i. Define the similarity between experts ℓ_i and ℓ_j as the percentage of pairwise preferences they agree on:

$$sim(\ell_i, \ell_j) = \frac{|\mathcal{P}_{\ell_i} \cap \mathcal{P}_{\ell_j}|}{|\mathcal{P}_{\ell_i} \cup \mathcal{P}_{\ell_j}|} \tag{7}$$

TREC	$\overline{d}$	ρ	$\overline{d_{\text{Auto}}}$	ρ	$\overline{d_{\text{Man}}}$	ρ	$\overline{d_{\text{rel}}}$	ρ	$\overline{d_{\text{non}}}$	ρ
3	0.216	0.776	0.245	0.698	0.189	0.842	0.324	0.718	0.150	0.706
4	0.124	0.838	0.144	0.967	0.110	0.867	0.214	0.820	0.092	0.766
5	0.160	0.577	0.165	0.880	0.156	0.599	0.261	0.640	0.129	0.492
6	0.128	0.646	0.136	0.857	0.101	0.746	0.231	0.743	0.103	0.552
7	0.185	0.605	0.192	0.921	0.149	0.171	0.325	0.649	0.150	0.505
8	0.208	0.748	0.214	0.917	0.156	0.543	0.326	0.757	0.171	0.690

Table 4: Average similarity for subsets of systems for each TREC. $\overline{d}$ is averaged over all runs in the set. $\overline{d_{\text{Auto}}}$ is averaged over automatic runs only; $\overline{d_{\text{Man}}}$ is averaged over manual runs only. $\overline{d_{\text{rel}}}$ is calculated only over the relevant documents and averaged over all runs; likewise for $\overline{d_{\text{nonrel}}}$. The number ρ next to each average similarity is the correlation between that similarity and mean average precision.

This similarity is defined for a single topic; we define $Sim(\ell_i, \ell_j)$ as the average similarity over all topics. We then define the similarity between one expert and the other $k-1$ experts as:

$$d(\ell_i) = \frac{1}{k-1} \sum_{j \neq i}^{k} Sim(\ell_i, \ell_j) \tag{8}$$

From here on, we will use the word *similarity* to refer to $d(\ell_i)$. We also define *relevant similarity* $d_{\text{rel}}(\ell_i)$, computed by calculating Eq. 7 over preferences among relevant documents only, and *nonrelevant similarity* $d_{\text{nonrel}}(\ell_i)$ computed by calculating Eq. 7 over preferences among nonrelevant documents only, and in both cases averaging over all topics and all systems as in Eq. 8.

Aslam and Savell (2003) previously defined similar distance metrics, but over common documents rather than common pairwise preferences. Using documents alone abstracts away from differences in the way the documents are ranked, which has a strong effect on average precision. Using pairwise preferences takes the rankings into account: two systems will be more similar not only by retrieving the same documents, but also by putting them in the same order.

Table 4 shows mean similarity $\overline{d}$ for each data set. It is striking that the systems are not particularly similar to each other: on average they have only about 16% of preferences in common. Even if we remove the manual runs, systems do not exhibit much similarity ($\overline{d_{\text{Auto}}}$ in Table 4), and the similarity among the automatic runs is not much greater than the similarity among the manual runs ($\overline{d_{\text{Man}}}$ in Table 4).

5.1 Analysis of Probability Estimates

Table 2 shows that the probability predictions made by our maximum likelihood method are fairly bad at predicting relevance, but the ratio of relevant documents increases with the probability estimates.

The likelihood is

$$\mathcal{L}(\Theta) = \prod_i \prod_j P(X_i > X_j | \theta_i, \theta_j)^{n_{ij}} \prod_i \xi_i^{\alpha} (1 - \xi_i)^{\beta}$$

Recall from Section 4 that $P(X_i > X_j | \theta_i, \theta_j) = \frac{\exp(\theta_i - \theta_j)}{1 + \exp(\theta_i - \theta_j)}$ (Eq. 3). Since that function is monotonically increasing, it will generally be true that greater n_{ij} results in greater difference

between θ_i and θ_j. In other words, the more often $i \succ j$ by the experts, the greater the difference between p_i and p_j (since p_i is a monotonically increasing function of θ_i (Eq. 4)). This is confirmed in the data: the correlation between n_{ij} and $p_i - p_j$ is 0.999 (averaged over all collections).

The correlation between $d(\ell)$ and $\mathcal{E}MAP$ is 0.988, a near-perfect relationship. This is fairly easy to explain: consider two systems ℓ_i and ℓ_j with $d(\ell_i) > d(\ell_j)$. It follows from the definition of d that ℓ_i has more pairwise preferences in common with the other systems than ℓ_j does. By the argument above, the more often a particular preference is expressed by the experts, the greater the difference in probabilities will be between the two documents. Then the following theorem tells us that $\mathcal{E}MAP(\ell_i) > \mathcal{E}MAP(\ell_j)$:

Theorem 1. *Suppose ranked lists ℓ_i, ℓ_j are identical except that ℓ_i prefers document A to document B and ℓ_j prefers document B to A. If $p_A > p_B$, then $E[AP(\ell_i)] > E[AP(\ell_j)]$.*

The proof is presented in the Appendix.

If $d(\ell_i) > d(\ell_j)$ implies that $\mathcal{E}MAP(\ell_i) > \mathcal{E}MAP(\ell_j)$, then the correlation between d and $\mathcal{E}$MAP is guaranteed to be high.

5.2 Analysis of Similarity

Since d and $\mathcal{E}$MAP are so highly correlated (and are expected to be highly correlated for any data set), we can use d rather than the more complicated and less intuitive $\mathcal{E}$MAP to explore the relationship between predicted performance and actual performance.

Although both our algorithm and Soboroff's are capturing the "popularity" of the documents (as shown above and by Aslam and Savell (2003) respectively), the fairly low similarities shown in Table 4 motivated us to explore the relationship between similarity and performance in more depth.

We modified the runs in two ways, both of which kept mean average precisions constant while increasing or decreasing average similarity. The first experiment randomly permuted each ranked list by replacing each document with a randomly-chosen document with the same relevance. For the second, we replaced each document with a document of the same relevance chosen deterministically based on order. For example, the first relevant document retrieved by each system would be replaced by the same relevant document for all the systems.

The first experiment decreases similarity dramatically: after randomly permuting the TREC-5 ranked lists, for example, the mean similarity is around 5%. But the correlation between similarity and MAP has the opposite effect: it *increases* to 0.934, a near-perfect correlation. The second experiment has the opposite effect: it increases similarity dramatically to around 67%, but *decreases* the correlation between similarity and MAP to 0.059, nearly no correspondence at all. Table 5 shows the results of these experiments for each of the six data sets.

Popularity is therefore not necessarily a bad criterion for estimating performance; it depends on the distribution of the relevant and nonrelevant documents in the rankings. In the first experiment, the popularity of any given document or preference is low, but since systems are reasonably good at retrieving relevant documents, the expectation is that any given relevant document will be ranked above any given nonrelevant document. In the second, the popularity of any given document or preference is high, but since they are always in the same order, there will be some nonrelevant documents that are always ranked above the relevant documents.

Table 4 shows another factor: the similarity between these systems among relevant documents only is greater than the similarity among nonrelevant documents only; the ratio is about two to one.

experiment	statistic	TREC-3	TREC-4	TREC-5	TREC-6	TREC-7	TREC-8
random	$\overline{d}$	0.063	0.046	0.043	0.048	0.054	0.056
replacement	$\rho(d, MAP)$	0.967	0.964	0.934	0.848	0.932	0.932
deterministic	$\overline{d}$	0.750	0.672	0.803	0.772	0.780	0.792
replacement	$\rho(d, MAP)$	0.605	0.327	0.059	0.219	−0.049	0.159

Table 5: Results of permuting lists for all six data sets. The first experiment decreases average similarity $\overline{d}$ while increasing the correlation $\rho(d, MAP)$ between similarity and MAP. The second increases average similarity while decreasing the correlation.

Our first experiment increased this ratio to nearly three to one, while the second decreased it to the point that $\overline{d_{\text{non}}}$ was greater than $\overline{d_{\text{rel}}}$.

This suggests another reason our algorithm works as well as it does: these systems are more similar to each other in how they rank relevant documents than in how they rank nonrelevant documents. Lee (1997) provided evidence for this in a metasearch context, showing that the amount of overlap among relevant documents retrieved was greater than the amount of overlap among nonrelevant documents retrieved. Again, using pairwise preferences allows us to measure the similarity in ranking as well as in documents retrieved.

As alluded to above, these systems are on average reasonably good: as the MAPs in Figure 1 show, they rank relevant documents above nonrelevant documents more often than not. Our conclusion is that these data sets have particular properties that make good results easy to obtain; they consist of good systems with low similarity on average but higher similarity among relevant documents than among nonrelevant documents. These properties should not necessarily be expected to occur in other data sets. In particular, our second experiment above suggests that good systems with high overall similarity but lower relevant similarity than nonrelevant similarity are harder to evaluate.

5.3 Summary

We have demonstrated an algorithm that, like that of Soboroff et al. (2001), ranks retrieval systems without relevance judgments. We have shown that it works as well as it does for two reasons: first, it rewards systems that are most similar to the others (as Aslam and Savell (2003) showed for Soboroff's work); second, the systems are on average good at retrieving relevant documents. We have argued that the combination of low overall similarity, goodness of rankings, and greater similarity among relevant ranks than nonrelevant ranks result in these data sets being easy for any algorithm.

We believe it is possible to get good results on these data sets almost by accident. An algorithm need only identify those systems that are a bit different from the others but that have done a good job at retrieving relevant documents. This is what we did in Section 4.4 when we manually reweighted the manual runs. Next we shall show that the same effect can be achieved automatically.

6 Iterative Reweighting

In this section we illustrate how we can make a very small number of relevance judgments to give comparable results to manually reweighting manual runs as in Section 4.4, Figure 2. Our goal is to show that it is fairly easy to achieve good results on these data sets, even if our algorithm is not

doing what we necessarily think it is.

The algorithm is an iterative reweighting algorithm described in a general form by Arora et al. (2006). We apply it as follows: first, we assign a weight $w_\ell = 1$ to each expert ℓ. We then estimate probabilities of relevance for every document by maximizing the weighted likelihood Eq. 6. We judge the document with the highest probability of relevance for each topic.

We then look at the preferences of each expert. For each of an expert's preferences that were correct according to the relevance judgments, we increase its weight by a factor of $1 + \epsilon$[1]. For each preference that was incorrect, we decrease its weight by a factor of $1 - \epsilon$. For example, suppose document A is judged relevant and B nonrelevant. Experts that preferred A to B have their weight increased; experts that preferred B to A have their weight decreased. The weights of experts that expressed no preference (by retrieving neither A nor B) are unchanged.

We then maximize the weighted likelihood Eq. 6 to obtain new estimates of relevance Θ. We do this for $\lfloor \ln |\mathcal{L}| \rfloor$ iterations, where $\mathcal{L}$ is the set of retrieval systems being evaluated. This number is chosen to minimize the expected number of errors.

6.1 Results

Reweighted rankings are shown in Figure 3. Iteratively reweighting systems using advice from the relevance judgments increases the τ correlations above the levels seen by manually reweighting manual runs in Figure 2.

It is interesting to look at the weights of the experts. We might expect that the weights correlate well to the true ranking: since they are being increased for successes and decreased for errors, they should roughly reflect that better systems make fewer errors. But in fact they do not: the correlation is never better than about 0.3, and for some collections it is not significantly different from random! It is always much less than the correlation between $\mathcal{E}$MAP and MAP.

Why is this? It is because this algorithm is in fact identifying and reweighting the systems that have retrieved different relevant documents, not the ones that are better. To see this, note that the highest-probability documents will be those that were retrieved highly by many systems. This means we are gaining information primarily about the systems that retrieved the most similar documents: the automatic runs. As it turns out, the average weight of the automatic runs is 0.454, while the average weight of the manual runs is 0.961. The fact that the manual weights are so close to 1 indicates that we have gathered almost no information about them at all. Instead, we downweighted the runs that have the highest similarity to the cluster.

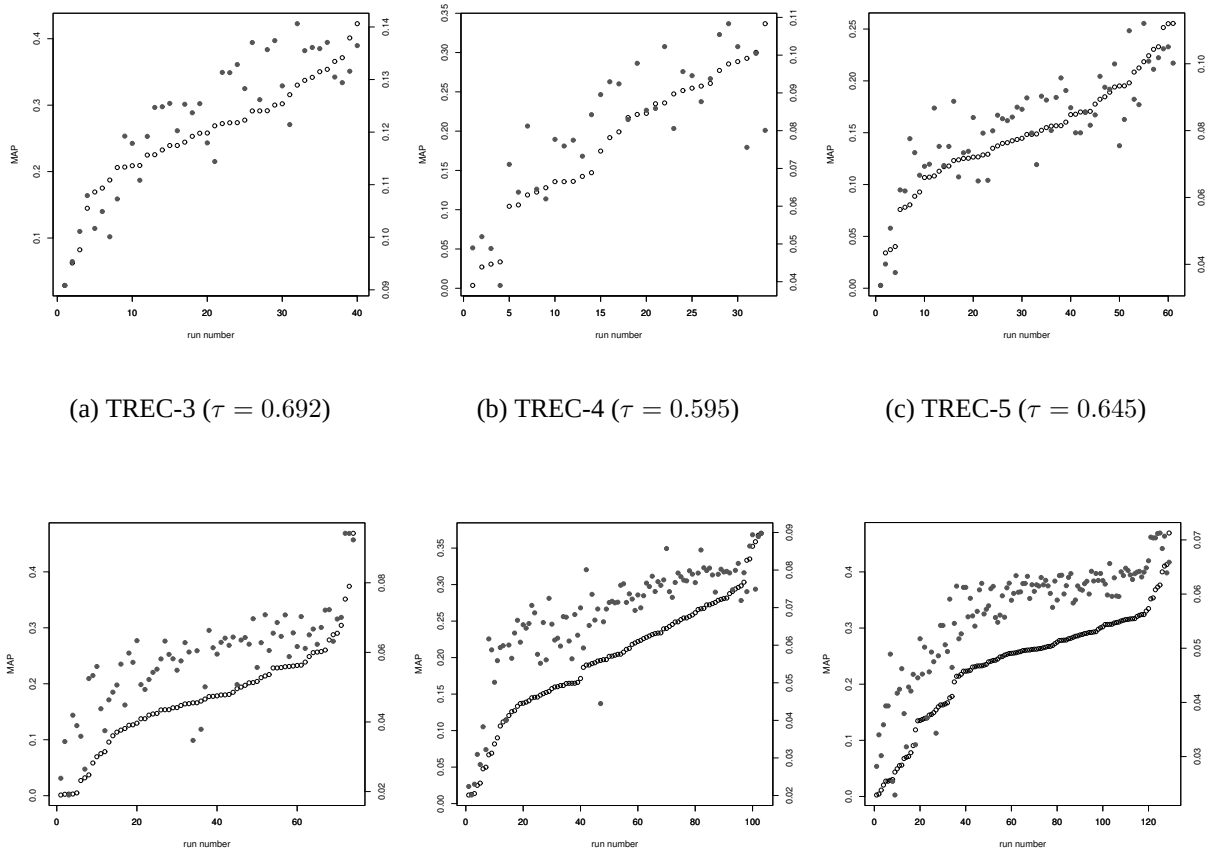
It is also interesting to look at the rankings of systems by calculating MAP using only the $\ln |\mathcal{L}|$ documents judged, and assuming that all unjudged documents are nonrelevant. In fact, the correlations are quite high, which again demonstrates our point that it is easy to get good results with very little effort.

7 Solutions

The results and analysis in the previous section point towards the TREC ad hoc sets having a very high baseline for any evaluation study. What can be done about it?

The obvious first idea is to find sets of systems that are "harder" than the TREC sets. So far, after examining sets such as Robust results, HARD results, and Terabyte track results, we have found that if anything they are *easier* than the ad hoc track results typically used in evaluation studies: the

[1] $\epsilon \leq \frac{1}{2}$; we chose $\epsilon = 0.1$.



(a) TREC-3 ($\tau = 0.692$) (b) TREC-4 ($\tau = 0.595$) (c) TREC-5 ($\tau = 0.645$)

(d) TREC-6 ($\tau = 0.667$) (e) TREC-7 ($\tau = 0.730$) (f) TREC-8 ($\tau = 0.731$)

Figure 3: Rank results after $\ln |\mathcal{L}|$ judgments and iterative reweighting.

systems are about as good, but have less similarity to each other overall, and the ratio of relevant similarity to nonrelevant similarity is higher. It appears that the ad hoc sets are the best currently available.

One alternative is to generate random ranked lists for testing. This is not entirely satisfactory, though, as it is not clear what a realistic simulated ranked list should look like. Furthermore, given the results of our random replacement experiment in Section 5.2, generating random ranked lists will most like create easier data sets!

Thus we recommend the use of formal proof (along with informal argument when necessary) and hypothesis testing to demonstrate why a particular algorithm works. In the following two subsections we illustrate how these may be used below before presenting general recommendations for hypothesis testing procedures.

7.1 Formal Proof and Informal Argument

The advantage of formal proof is that it can sidestep questions about data entirely. It can also suggest places where counterexamples may be lurking. The disadvantage is that it may only be possible to prove weak results; in that case reasoning informally may be sufficient.

We will not prove anything new about this algorithm; Arora et al. (2006) have proved the

following results. After t iterations of judging and reweighting, let m_ℓ^t be the number of mistakes made by expert ℓ, c_ℓ^t be the number of correct preferences by expert ℓ, and M^t be the expected number of incorrect preferences minus the expected number of correct preferences in the consensus.

Theorem 2 (Arora et al.). *After t rounds, for any expert ℓ we have*

$$M^t \leq \frac{\ln |\mathcal{L}|}{\epsilon} + m_\ell^t(1+\epsilon) - c_\ell^t(1-\epsilon)$$

(In fact, they prove a more general bound, but this is strong enough for our purpose.)

Note that ℓ could be the best expert—the one that has made the fewest errors in preferences. This means that our consensus pairwise preferences will approach those of the best expert, and therefore we can have confidence that the run it says is best really is one of the best.

Theorem 1 in Section 5.1 implies that the more similar an expert is to the consensus, the higher its $\mathcal{E}$MAP will be. Since the consensus preferences will trend towards the best expert's preferences, that expert will be identified as the best by $\mathcal{E}$MAP . It also means that the closer an expert is to the best, the more likely it is to be identified as good by $\mathcal{E}$MAP .

We argued in Section 5.1 that $\theta_i - \theta_j$ is strongly correlated to n_{ij}. It follows that the document with the greatest θ_i will be the one that is preferred most often to other documents, i.e. retrieved at the highest rank by the most systems. Therefore judging this document tells us the most about the majority set. If the majority set is frequently wrong, the systems in it will be down-weighted, while the minority set will remain the same or be up-weighted if they are good.

7.2 Hypothesis Testing

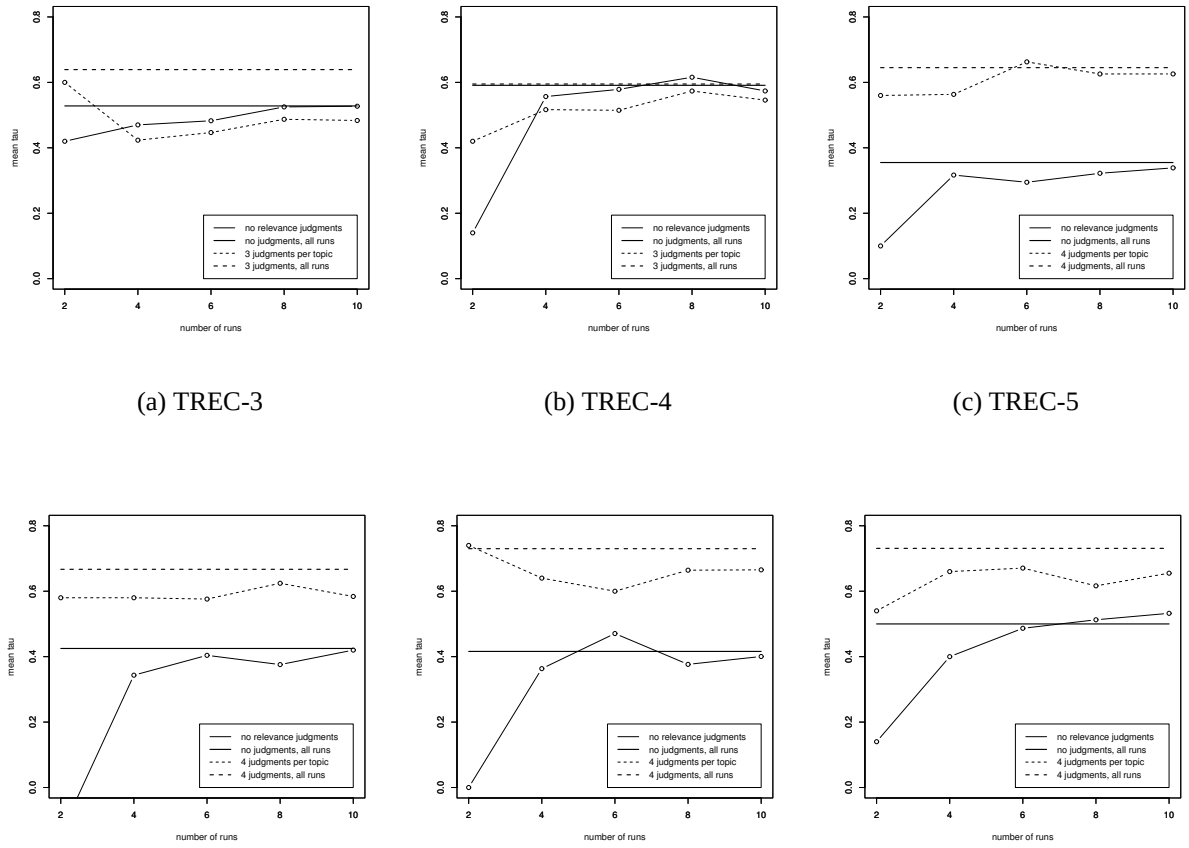
The purpose of hypothesis testing is to decide whether a difference in some measurement is unlikely to have occurred by chance. Hypothesis testing, while common in experiments on retrieval, has not been used to compare evaluation algorithms, to the best of our knowledge. We argue that it should be, even if formal proof is enough to justify the algorithm.

Our proposal for testing evaluation algorithms is as follows: first, choose random subsets of k systems from one of the TREC corpora. Run the algorithm to some stopping point on each set (an algorithm may not have a fixed stopping point; in that case it should be run to several different fixed stopping points). Run a baseline algorithm to the same point on the same sets. Since both algorithms were run on the same data and to the same point, they may be compared using a paired (one-sample) hypothesis test, thus accounting for possible explanatory variables such as corpus and topics. This procedure should be duplicated on several different corpora.

Our iterative reweighting algorithm stops after $\ln |\mathcal{L}|$ documents have been judged. Figure 4 shows the result of running the algorithm as well as the no-cost algorithm on randomly-chosen subsets of 2, 4, 6, 8, and 10 runs from each TREC. Note that the average performance on subsets of systems is nearly always lower than the performance on the full set of systems (indicated by the horizontal lines). This suggests that testing on subsets is "harder".

Table 6(a) shows the mean τ correlation for subsets of 10 runs from each TREC. It also shows the p-value of the paired one-sided t-test between iterative reweighting and no relevance judgments at all. We also tested whether ranking documents by $\mathcal{E}$MAP (plugging in our maximum likelihood estimates of probability) was significantly different than ranking documents by MAP alone (making the assumption that all unjudged documents are nonrelevant). The results are shown in Table 6(b).

The results in Table 6 show how much performance can vary over data sets. Judging three documents is actually significantly *worse* than judging none at all for TREC-4! Table 6(b) in



(a) TREC-3 (b) TREC-4 (c) TREC-5

(d) TREC-6 (e) TREC-7 (f) TREC-8

Figure 4: A comparison of the baseline no-judgment algorithm (solid lines) to the iterative reweighting algorithm (dotted lines) on randomly-chosen subsets of systems. The straight lines show the performance on the full set of systems.

particular reinforces the variance in performance over data set, showing that ranking my $\mathcal{E}$MAP is significantly better for two sets but significantly worse for one.

Algorithms of this type are often "anytime" algorithms, meaning they can run indefinitely. In order to compare them using hypothesis testing, they must be stopped at some point. The choice of stopping point depends on the measurement that is to be tested. For example, to test whether one algorithm gives a higher τ correlation than another, we would run both algorithms for the same number of relevance judgments. To test whether one algorithm requires fewer judgments than another, we would run both to a stopping point (determined possibly by τ correlation) and compare the number of relevance judgments it took to get there. Generally the choice of stopping condition should be clear from the hypothesis that needs to be tested.

8 Conclusions

Starting from the high performance baseline when evaluating with no relevance judgments at all, we argued that the TREC data sets usually used for experiments in evaluation studies have a much higher baseline than previously assumed. Following Aslam et al. (2003) and Lee (1997), we

TREC	baseline τ	reweight τ	p-value
3	0.527	0.513	0.205
4	**0.574**	0.542	0.029
5	0.339	**0.489**	0.000
6	0.420	**0.557**	0.000
7	0.400	**0.480**	0.000
8	0.532	**0.575**	0.004

(a) Mean τ correlations for the baseline (no judgments) and iterative reweighting.

reweight+$\mathcal{E}$MAP	reweight+MAP	p-value
0.513	**0.564**	0.001
0.542	0.532	0.286
0.489	0.456	0.056
0.557	0.574	0.190
0.480	0.355	0.000
0.575	0.438	0.000

(b) Mean τ correlations for iterative reweighting and ranking by $\mathcal{E}$MAP vs. iterative reweighting and ranking by MAP.

Table 6: Mean τ correlations when running three algorithms over randomly-chosen subsets of 10 systems. Bold numbers indicate significant differences at $\alpha = 0.05$.

analyzed the data sets to see why that is true: the systems are good, have fairly low similarity to each other, but are more similar in their relevant rankings than their nonrelevant rankings.

Because of this high baseline, more rigorous testing is needed in evaluation studies. We specifically recommend the use of hypothesis tests by evaluating on randomly-chosen subsets of systems from the TREC sets. We also recommend the use of formal proof to argue about algorithms.

Again, we do not claim that previous studies are wrong. We strongly believe that they will hold up to more rigorous testing. We simply believe that these types of studies should present more evidence to support their conclusions, just as studies on text retrieval are required to do.

Acknowledgments

We would like to thank the anonymous referees for their helpful feedback. This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings, and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

S. Arora, E. Hazan, and S. Kale. Multiplicative weights method: A meta-algorithm and its applications. Retrieved from http://www.cs.princeton.edu/ arora/pubs/MWsurvey.pdf, 2006.

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.

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.

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.

J. A. Aslam and E. Yilmaz. A geometric interpretation and analysis of r-precision. In *Proceedings of CIKM*, pages 664–671, 2005.

C. Buckley and E. M. Voorhees. Evaluating Evaluation Measure Stability. In *Proceedings of SIGIR*, pages 33–40, 2000.

C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of SIGIR*, pages 25–32, 2004.
B. Carterette and J. Allan. Incremental Test Collections. In *Proceedings of CIKM*, pages 680–687, 2005.
B. Carterette, J. Allan, and R. K. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of SIGIR*, pages 268–275, 2006.
B. Carterette and D. I. Petkova. Learning a ranking from pairwise preferences. In *Proceedings of SIGIR*, 2006.
G. V. Cormack, C. R. Palmer, and C. L. Clarke. Efficient Construction of Large Test Collections. In *Proceedings of SIGIR*, pages 282–289, 1998.
A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2004.
D. A. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of SIGIR*, pages 329–338, 1993.
T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of KDD*, pages 133–142, 2002.
P. Komarek and A. Moore. Making logistic regression a core data mining tool: a practical investigation of accuracy, speed, and simplicity. Technical Report CMU-RI-TR-05-27, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2005.
J. H. Lee. Analyses of multiple evidence combination. In *Proceedings of SIGIR*, pages 267–276, 1997.
D. Mease. A Penalized Maximum Likelihood Approach for the Ranking of College Football Teams Independent of Victory Margins. *The American Statistician*, Nov 2003.
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.
M. Sanderson and J. Zobel. Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings of SIGIR*, pages 162–169, 2005.
J. Savoy. Statistical inference in retrieval effectiveness evaluation. *Information Processing and Management*, 33(4):495–512, 1997.
I. Soboroff, C. Nicholas, and P. Cahan. Ranking Retrieval Systems without Relevance Judgments. In *Proceedings of SIGIR*, pages 66–73, 2001.
C. J. van Rijsbergen. *Information Retrieval*. 1979.
E. Voorhees. Variations in Relevance Judgments and the Measurement of Retrieval Effectiveness. In *Proceedings of SIGIR*, pages 315–323, 1998.
J. Zobel. How Reliable are the Results of Large-Scale Information Retrieval Experiments? In *Proceedings of SIGIR*, pages 307–314, 1998.

Appendix

Theorem 1. *Suppose ranked lists ℓ_i, ℓ_j are identical except that ℓ_i prefers document A to document B and ℓ_j prefers document B to A. If the consensus opinion is that A is preferred to B, then $E[AP(\ell_i)] > E[AP(\ell_j)]$.*

Proof. As defined in Section 4.2, the preference $A \succ_{\ell_1} B$ implies the rank of A is less than the rank of B, i.e. $r_{\ell_1}(A) < r_{\ell_1}(B)$. Likewise, $B \succ_{\ell_2} A$ implies $r_{\ell_2}(B) < r_{\ell_2}(A)$. The consensus preference

$A \succ B$ implies $p_A > p_B$. Since the two lists are identical except for A and B, it follows that for all documents $i \neq A, B$, $r_{\ell_1}(i) = r_{\ell_2}(i)$. It further follows that $r_{\ell_1}(A) = r_{\ell_2}(B)$ and $r_{\ell_1}(B) = r_{\ell_2}(A)$.

We will define δ such that $p_A = p_B + \delta$.

Eq. 1 defined the expectation of average precision as:

$$E[AP(\ell_1)] = \frac{1}{\sum p_i} \sum_i \frac{1}{r_{\ell_1}(i)} \left(p_i + \sum_{j \leq i} p_i p_j \right) + \epsilon$$

To simplify the algebra, we will split this into four parts E_1, E_2, E_3, E_4 such that $E[AP] = E_1 + E_2 + E_3 + E_4$.

$$\begin{aligned}
E_1[AP(\ell_1)] &= \frac{1}{r_{\ell_1}(A)} p_A + \frac{1}{r_{\ell_1}(B)} p_B \\
E_2[AP(\ell_1)] &= \frac{1}{r_{\ell_1}(B)} p_A p_B \\
E_3[AP(\ell_1)] &= \sum_{i \neq A,B} \frac{1}{\max\{i, r_{\ell_1}(A)\}} p_i p_A + \sum_{i \neq A,B} \frac{1}{\max\{i, r_{\ell_1}(B)\}} p_i p_B \\
E_4[AP(\ell_1)] &= \sum_{i \neq A,B} \left(\frac{1}{i} p_i + \sum_{j < i; j \neq A,B} p_i p_j \right)
\end{aligned}$$

We define the same four quantities for $AP(\ell_2)$. The only difference is that $E_2[AP(\ell_2)] = 1/r_{\ell_2}(A) p_A p_B$.

Since E_4 excludes documents A and B, it follows that $E_4[AP(\ell_1)] = E_4[AP(\ell_2)]$. $E_2[AP(\ell_1)] = E_2[AP(\ell_1)]$ because $r_{\ell_1}(B) = r_{\ell_2}(A)$. We will show that $E_1[AP(\ell_1)] > E_1[AP(\ell_2)]$ and $E_3[AP(\ell_!)] \geq E_3[AP(\ell_2)]$.

$$\begin{aligned}
E_1[AP(\ell_1)] &= \frac{1}{r_{\ell_1}(A)} p_A + \frac{1}{r_{\ell_1}(B)} p_B \\
&= \frac{1}{r_{\ell_1}(A)} (p_B + \delta) + \frac{1}{r_{\ell_1}(B)} (p_A - \delta) \\
&= \frac{1}{r_{\ell_2}(B)} p_B + \frac{1}{r_{\ell_2}(A)} p_A + \delta \left(\frac{1}{r_{\ell_1}(A)} - \frac{1}{r_{\ell_1}(B)} \right) \\
&> \frac{1}{r_{\ell_2}(B)} p_B + \frac{1}{r_{\ell_2}(A)} p_A = E_1[AP(\ell_2)]
\end{aligned}$$

And for each of the terms in the sums in E_3,

$$\begin{aligned}
\frac{1}{\max\{i, r_{\ell_1}(A)\}} p_i p_A &= \frac{1}{\max\{i, r_{\ell_2}(B)\}} p_i p_A \\
&\geq \frac{1}{\max\{i, r_{\ell_2}(A)\}} p_i p_A
\end{aligned}$$

From these equalities and inequalities, we can conclude that

$$\begin{aligned}
E[AP(\ell_1)] &= E_1[AP(\ell_1)] + E_2[AP(\ell_1)] + E_3[AP(\ell_1)] + E_4[AP(\ell_1)] \\
&> E_1[AP(\ell_2)] + E_2[AP(\ell_2)] + E_3[AP(\ell_2)] + E_4[AP(\ell_2)] = E[AP(\ell_2)]
\end{aligned}$$

□