

The Effect of Assessor Errors on IR System Evaluation

Ben Carterette
Dept. of Computer and Information Sciences
University of Delaware
Newark, DE 19716
carteret@cis.udel.edu

Ian Soboroff
National Institute of Standards and Technology
Gaithersburg, MD 20899
ian.soboroff@nist.gov

ABSTRACT

Recent efforts in test collection building have focused on scaling back the number of necessary relevance judgments and then scaling up the number of search topics. Since the largest source of variation in a Cranfield-style experiment comes from the topics, this is a reasonable approach. However, as topic set sizes grow, and researchers look to crowdsourcing and Amazon’s Mechanical Turk to collect relevance judgments, we are faced with issues of quality control. This paper examines the robustness of the TREC Million Query track methods when some assessors make significant and systematic errors. We find that while averages are robust, assessor errors can have a large effect on system rankings.

Categories and Subject Descriptors: H.3.4 [Information Storage and Retrieval]: *Systems and Software—Performance evaluation*, H.3.5 *Online Information Services*

General Terms: Experimentation, Measurement

Keywords: assessor error, retrieval test collections

1. INTRODUCTION

Since TREC began in 1992 as the first large-scale use of pooling to create test collections, a great deal of research has focused on examining the quality of pooled test collections in spite of violations of the Cranfield assumptions [9, 17, 13, 14, 15] and on refining pooling to reduce costs and/or maximize quality [8, 12, 11, 3, 16, 5]. The TREC Million Query track [1] has emerged as a testbed for modern test-collection building methods, primarily those of Aslam and Pavlu [2] and Carterette et al. [6].

A primary motivation in many modern collection-building methods is to reduce the costs associated with making relevance judgments. Relevance assessors can be expensive to hire, train, and use, and particularly in the academic community where funding may not be available for collection building, low-cost (or zero-cost) methods have broad appeal, if not as yet broad application.

Recently, crowdsourcing and Amazon’s Mechanical Turk (MTurk)¹ have been used as sources of relevance judgments.

¹<https://www.mturk.com/mturk/welcome>

These approaches have a very low cost per judgment, but they may have somewhat higher design costs and require additional cost and effort to control for assessor error. In particular, if an MTurk worker’s only goal is to complete the task, the judgments may not look very different from random. Soboroff et al. has simulated a family of cases of random assessor errors: the assessors know roughly *how many* relevant documents there are (give or take a standard deviation or two), but they pay little to no attention to *which* documents they are judging relevant [12]. Soboroff et al. assigned relevance to documents in the pool randomly, creating a *pseudo-rels* that they used to evaluate systems. Surprisingly, evaluation results over the pseudo-rels correlated significantly to evaluation results over the “true” relevance judgments from the assessors.

That work had the relative luxury of deep pools of documents, which may have resulted in emergence of patterns. But even an assessor making judgments at random would take a fair amount of time to make, say, 12,000 relevance judgments. Recent work on test collections suggests that assessing more topics with fewer judgments each is a more cost-effective approach, since the topics are a larger source of variance than the missing judgments [11, 7]. But that work assumes the judgments are in some sense “perfect”, i.e. that errors made by assessors have inconsequential variance and no bias. This is almost certainly not the case; assessors make mistakes due to misunderstandings of the task or documents, fatigue, boredom, and for many other reasons. These mistakes surely have a larger impact on evaluation when there are only a few relevance judgments to begin with.

Furthermore, most TREC collections are built using trained relevance assessors. Practitioners and researchers using modern collection building methods such as those pioneered in the TREC Million Query track may wish to create their own test collections using available sources of labor, such as Amazon’s Mechanical Turk or other crowdsourcing methods. Bailey et al. [4] found that assessors of differing task and domain expertise could affect system rankings markedly; Kinney et al. [10] found that non-expert assessors judging domain-specific queries make significant errors affecting system evaluation. When assessors are not closely managed or highly trained, mistakes must be common.

Our goal is to investigate the effect of assessor errors by simulating different assessor “archetypes” in a low-cost large-scale test collection scenario. We propose several models of how assessors might make errors, then use the models to simulate assessors going through the process of making judgments. We focus specifically on the effect in the TREC

Million Query track test collections, where there are many queries with very few judgments each.

2. ASSESSOR BEHAVIOR AND ERRORS

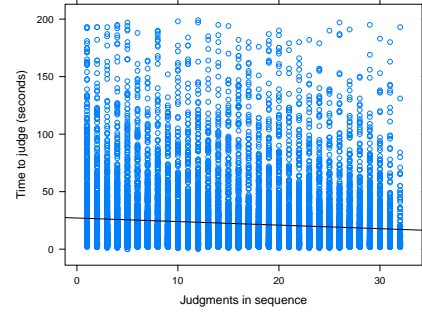
We mined a large log of assessor interaction data from the TREC 2009 Million Query (MQ) track for patterns of assessor behavior. The log contains a record for each judgment; each record consists of a timestamp, the query number, the document ID, the assessor ID, and the judgment itself. We examined the first 32 judgments per topic and excluded inter-judgment times in excess of 200 seconds. Based on this data, and further on our experience managing relevance assessors for various projects, we present some types of errors assessors might make, some ways we can model them, and hypothesized effects on evaluation in a MQ-type setting.

First we identify the following broad trends that can be used to model assessor behavior. These formed the basis for some of our models below.

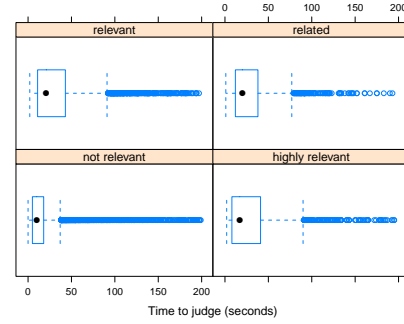
1. Time between judgments decreases slightly from the first judgments for a topic to the last judgment for a topic (Fig. 1(a)).
2. It takes less time to judge a nonrelevant document than to judge a document with any degree of relevance (Fig. 1(b)).
3. Assessors can vary in their judging times (Fig. 1(c))
Our assessors do not display much variation but we might see more so in a larger set.
4. Measured across non-overlapping but large sets of topics, assessors vary in the proportion of documents they judge relevant (Fig. 1(d)).

These observations clearly indicate that assessors behave differently (i.e., there is variance due to the assessor), and moreover that there is interaction between assessor and document. There may also be interaction between assessor and topic. Because each topic was judged only once we will not find evidence of that in aggregate, but we looked for a particular type of interaction: topics that an assessor gave less attention to than normal, possibly due to unhappiness with the documents they were asked to judge. To do this, we modeled time between judgments as a function of assessor, document judgment, and where in the sequence the judgment fell. We then looked for topics for which the actual judgment times were lower than those predicted by the model across most of the sequence, specifically cases where at least 90% of the judgments were faster than expected. There were about 30 such topics, and for all 30 all the judgments were nonrelevant.

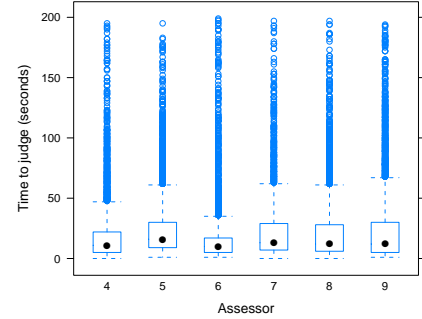
Finally, we looked for evidence of autocorrelation in judgments. We calculated the proportion of times an assessor judged a document relevant conditional on judging the previous document relevant, and contrasted that with the proportion conditional on judging the previous document nonrelevant (on a subset of queries with roughly equal proportions of relevance). Assessors are in fact more likely to make the same judgment twice in a row: $P(j_i = 1 | j_{i-1} = 1) = 0.22$, while $P(j_i = 1 | j_{i-1} = 0)$ is only 0.18. This difference is significant by a two-sample two-proportion test.



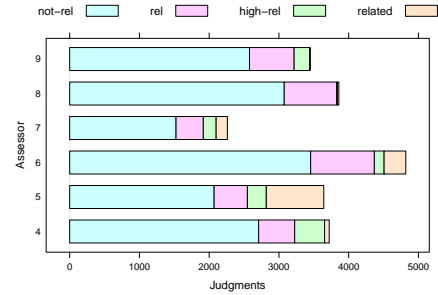
(a) Time to judge a document decreases slightly over the sequence of documents in a topic.



(b) Relevant, highly relevant, and related documents take longer to judge than non-relevant ones.



(c) Our assessors vary only slightly in their judgments times.



(d) Assessors vary in the proportion of documents they judge relevant, as measured over large non-overlapping topic sets.

Figure 1: Interaction log observations.

2.1 Model Distributions and Priors

We will simulate different types of assessors using models of systematic errors based on observations above. Given a set of (binary) judgments for a topic, we convert nonrelevant judgments to relevant and vice versa according to a model.

If we are not careful, we can easily inject too much bias into the evaluation. For instance, if we model an assessor that has a tendency to overrate relevance by changing a judgment of nonrelevance to relevance with fixed probability p , the expected effect is to simply increase the number of relevant documents to pn , where n is the number of judged documents. Increasing the number of relevant documents in this way does not really address the question; some of those topics may very clearly have no relevant documents and even an optimistic assessor would not say they do. We therefore tied model parameters to the number of known relevant documents for each topic.

To do this, we define a model in terms of “background” parameters modeling *average* behavior; these parameters are then adjusted by topic. This is easy to do using discrete distributions such as Bernoulli and Poisson distributions. The parameters of both distributions have natural conjugate priors that allow updating based on existing judgments.

The Bernoulli distribution can be seen as modeling the probability p that a document is relevant; the parameter p can be modeled as having a Beta distribution. A Beta distribution is specified with two parameters α, β and has mean $\alpha/(\alpha + \beta)$. The posterior of a Beta random variable also has a Beta distribution (hence it is conjugate); the posterior parameters are $\alpha + r, \beta + (n - r)$, where n is a number of observations and r is the number of positive observations. We can therefore view a Beta prior as a model of an assessor’s average probability of judging a document relevant, and a Beta posterior using the number of relevant documents r_q among n_q judgments for a topic q as a model of an assessor’s probability of judging a document relevant to q .

Similarly, the Poisson distribution can be seen as modeling the number of documents that will be judged relevant in some sequence length based on a rate parameter λ ; λ can be modeled as having a Gamma distribution. Like a Beta distribution, a Gamma distribution is specified with two parameters α, β , but its mean is α/β . The posterior of a Gamma random variable has a Gamma distribution with parameters $\alpha + r, \beta + n$. We can therefore view a Gamma prior as a model of an assessor’s rate of judging documents relevant, and a Gamma posterior using r_q and n_q as the rate adjusted for the topic.

2.2 Assessor Models

Our baseline model is that an assessor makes judgments **randomly**. Real assessors of course do not make random relevance judgments; this model only serves as comparison to the more informed models below. Using the Beta distribution as described above, the probability that a document will be judged relevant to a topic q will be $(\alpha + r_q)/(\alpha + \beta + n_q)$. So, for example, an assessor modeled by prior parameters $\alpha = 2, \beta = 8$ (expected to judge 20% of documents relevant) confronted by a topic for which all 32 documents have been judged nonrelevant will have a $2/42 = 0.05$ probability of judging each of those documents relevant. The effect of this is that the number of simulated relevant documents for each topic can be expected to be “near” the actual number of

relevant documents, but with enough noise that evaluation results will change.

Our first realistic model is an **unenthusiastic** assessor, that is, the assessor is not interested in reading or understanding documents and simply wants to complete the job. Judgments by this assessor may be characterized by a pattern such as judging everything nonrelevant, or alternating judgments in some pattern. Our discovery of topics that were completed faster than usual and with all nonrelevant judgments lends support to this model. For the unenthusiastic assessor we do not suppose that there is any particular probability model. The assessor just follows a fixed pattern, such as judging everything nonrelevant or alternating between relevant and nonrelevant judgments. This is in some sense the most biased model, in that the simulated judgments have nothing to do with the actual judgments.

Our second model is the **optimistic** assessor; he or she takes an overly-broad view of the topic and ends up judging things relevant that are not. It is well-known from work by Harman [9] and Voorhees [13] that assessors do differ reasonably in their judgment of relevance; in this model and the one following we presume a view of the topic which most observers would consider beyond reasonable, for example judging relevance solely by the presence of certain terms or not correctly identifying a spam document as not relevant. We model optimistic assessors as being more likely to judge a nonrelevant document relevant. The model parameters are again Beta parameters α, β , and the probability that a document will be judged relevant to a particular topic is $(\alpha + r_q)/(\alpha + \beta + n_q)$. However, for this model we can only change nonrelevant judgments to relevant; we will not change any of the relevant documents to nonrelevant.

Conversely, we model a **pessimistic** assessor as taking an overly-narrow view of the topic and judging documents nonrelevant that should be considered relevant. The model is identical to the optimistic model, except that relevant documents become nonrelevant with probability $(\beta + (n_q - r_q))/(\alpha + \beta + n_q)$ (which comes from treating a nonrelevant judgment as the “positive” outcome).

Another model we explore we call **topic-disgruntled**. This assessor chose a query for some reason (interest in topic, seemed easy), but the documents turned out to be something else (different topic, harder than expected). Disgruntled by the topic, the assessor begins to click through rapidly after the first few judgments. Again, the presence of topics completed faster than usual lends support to this model. The model is time-based. After k judgments, the assessor becomes disgruntled and judges the remaining documents nonrelevant. The parameter is a “patience parameter” λ ; after $n_q \lambda$ judgments (which are identical to the “true” judgments), the assessor judges everything else nonrelevant. This assessor has a Gamma prior specified by parameters α, β (resulting in prior patience $\lambda = \alpha/\beta$), and their posterior patience for a given topic is based on how many relevant documents there are: $\lambda = (\alpha + r_q)/(\beta + n_q)$.

Similar in execution to the disgruntled model is the **lazy/overfitting** assessor. The assessor sees a few very nonrelevant (or a few very relevant) documents at the start of judging and unjustly assumes all subsequent documents are likely to be the same. He or she begins rapidly entering judgments conforming to his early judgments. The model is implemented roughly the same as the topic-disgruntled

model, except that it only kicks in if the first $n_q\lambda$ judgments are all nonrelevant (or all relevant).

Another model is that the assessor is **fatigued**. The assessor starts each day alert and attentive, but tires as time passes. Judgments become more random as a result. This is also a time-based model. We again begin with a Beta prior. For this model, however, there is an assessor model with parameters α, β as well as separate priors for each judgment; we will denote the parameters γ_i, δ_i . The posterior for a given judgment will be $(\gamma_i + r_{qi})/(\gamma_i + \delta_i + i)$, where r_{qi} is the number of documents judged relevant to q up to judgment i . For $i = 0$ we will set γ_i, δ_i to zero. The first judgment for a topic, then, will be the same. For each subsequent judgment, the Beta parameters will grow with assessor parameters α, β : after each judgment, γ_i and δ_i increase by α and β respectively. The effect is that after k judgments, the posterior probability of judging a document relevant will be $(k\alpha + r_{qk})/(k\alpha + k\beta + k)$. As k increases, the assessor converges to judging every document according to their prior probability $\alpha/(\alpha + \beta)$.

Our final model is **Markovian**, that is, the assessor’s judgments are conditional on previous judgments. This could simulate an assessor who “feels bad” about judging too many nonrelevant documents in a row and thus takes a broader view of the topic over time, or one who takes a narrower view after judging many relevant documents in a row. The observation that assessors are more likely to make the same judgment twice in a row supports this model.

3. ASSESSOR SIMULATION

To analyze the effects of particular types of systematic error, we simulated assessors judging documents in a TREC-like setting: an assessor is given a topic description and reads documents to judge whether they are relevant to the topic. Since we are particularly interested in test collections with very many lightly-judged topics, we used the TREC 2009 Million Query track data as the starting point for our simulations. Before describing the simulation procedure and results, we briefly describe the track.

3.1 TREC Million Query Track

The Million Query track was designed to study the use of low-cost evaluation methods in the TREC setting. It produces test collections that consist of a large number of lightly-judged topics. Judgments in the Million Query track are either not relevant, related (but not relevant), relevant, or highly relevant. In 2009, 638 topics received a total of 34,534 judgments (54 per topic on average), of which 26% were either relevant or highly relevant. There were 95 topics for which no relevant documents were found.

The low-cost methods used by the track attempt to target judgments that are going to be more useful in evaluation. The Million Query track uses two methods to select documents to judge. One (statAP) is an approach based on statistical sampling, in which each judged relevant documents is taken to be representative of some population of relevant documents in the same “region” from which it was sampled [2]. The other (MTC) is an algorithmic approach that weighs documents according to how informative a judgment to them is expected to be; after each judgment, it recomputes the weights given the new information and presents the top-weighted document for judging [6].

Errors in judging can have unpredictable effects in both

methods. In statAP, an erroneous judgment of relevance can have a major impact on the estimated number of relevant documents for the topic, particularly if the judgment is to a document that has a low probability of being sampled. Conversely, an erroneous judgment of nonrelevance will cause the number of relevant documents to be underestimated. In MTC, an erroneous judgment can result in the algorithm taking an entirely different path.

The two different approaches to selecting judgments have different approaches to evaluation that are based on different assumptions. The sampling approach takes each judgment of relevance as representative of a set of relevant documents; this set is used to calculate an unbiased estimator of average precision. The algorithmic approach can take one of two tacks: it can either bound the differences in average precision between pairs of systems or it can compute a probability that the difference in average precision is less than zero over the space of possible judgments that could be made to unjudged documents. The probabilistic approach is generally more useful, and it has the advantage of being able to produce an estimate of average precision called “expected average precision” (EAP). Unlike the statMAP estimate, EAP is highly biased, but because it is meant for pairwise comparisons the bias can be expected to cancel out. Plots of EAP typically have very low values compared to plots of statAP.

We have decided to limit our focus to the effects on statAP. There are two reasons: first, because statAP estimates “look like” standard average precision, it is easy to see how errorful judgments are causing errors in the estimates. EAP estimates look very different from average precision, and because it is already very biased, changing the judgments will not necessarily cause obvious differences in their values. The second reason to prefer statAP is that MTC cannot effectively be simulated in the Million Query track data. If one judgment changes, it is likely that some future document selected for judging will be one that we do not already have a judgment on.

3.1.1 statAP

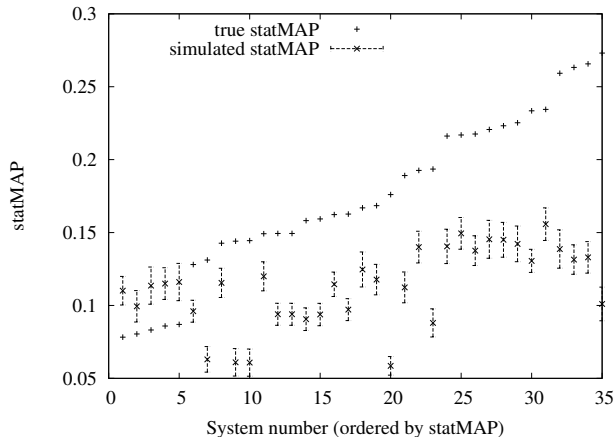
Some understanding of statAP is necessary to understand how an errorful judgment affects the estimate. statAP is a method for sampling a set of documents S to be judged, then using those judgments to estimate average precision. The statAP estimate is calculated as:

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

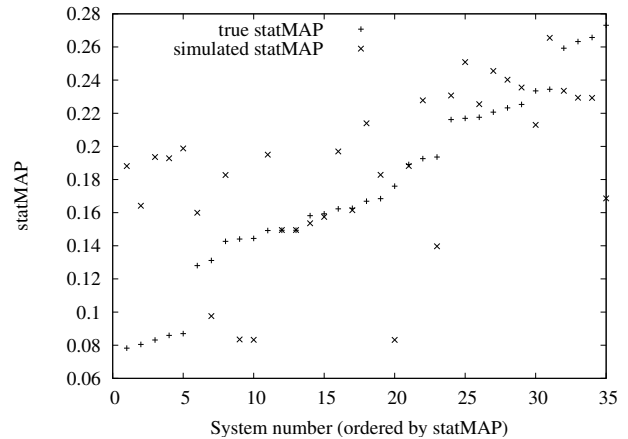
where x_d is the relevance of document d (1 for relevant, 0 for not relevant), π_d is an inclusion probability calculated for sampling, \hat{R} is an estimate of the number of relevant documents, and $\widehat{\text{prec@}r(d)}$ is an estimate of the precision at the rank at which document d appears. The estimates of precision and the number of relevant documents are:

$$\hat{R} = \sum_{d \in S} \frac{x_d}{\pi_d}, \quad \widehat{\text{prec@}k} = \frac{1}{k} \sum_{d \in S, r(d) \leq k} \frac{x_d}{\pi_d}$$

The ratio x_d/π_d can be thought of as the number of relevant documents that x_d is representative of in the same “region” of documents with similar inclusion probabilities. A lower π_d gives greater weight to a relevant document, increasing the estimated numbers of relevant documents compared to a relevant document with a higher π_d .



(a) Judgments from a random assessor give a small (but significant) rank correlation; $\tau = 0.35$.



(b) Judgments from an unenthusiastic assessor give a small (but significant) rank correlation; $\tau = 0.33$.

Figure 2: statMAP system scores after applying the two simplest assessor models.

3.2 Simulation Procedure

The simulation proceeds as follows: for a given topic, we start with the sequence of judgments in the same order they were originally made (this information is provided in the “fullrels” file distributed with the Million Query track data). We alter the judgment according to each of the models above. For those models that involve random sampling, we perform 25 trials on each judgment for each topic. We used increasing powers of two for parameter values, i.e. α and β ranged from 1 to 1024 independently. When complete, we have an errorful “prels” file that we can use to evaluate the Million Query track systems with statAP.

After re-evaluating Million Query track systems, the simplest approach to determining the effect of errors is to measure how well the new evaluation correlates to the “true” evaluation resulting from using the original relevance judgments. Kendall’s τ rank correlation is widely-used for this. Kendall’s τ is a function of the number of system pairs that swap between two rankings. The more swaps, the lower τ is; when $\tau = 1$ the rankings are identical.

3.3 Simulation Results

System evaluation results based on the judgments from the first two assessor models—random judging with prior parameters $\alpha = 1, \beta = 8$ and an unenthusiastic assessor that alternates between nonrelevant and relevant judgments to stay amused—are shown in Figure 2. In both cases the Kendall’s τ correlation to the official ranking is around 0.34, and remains consistently around 0.34 no matter what the prior parameters are and no matter what judging pattern is used (among those we tried). This can be thought of as a baseline for an assessor who is not actively malicious but is not interested in making an effort.

Figure 3 illustrates altered system rankings based on the other models for selected parameter values. For models with random sampling, error bars indicate the distribution of MAP estimates observed over 25 trials.

The optimistic and pessimistic models give very different results even when the prior parameter give equal probability of changing the judgment. Rank correlations based on optimistic judgments quickly degrade, while rank correlations

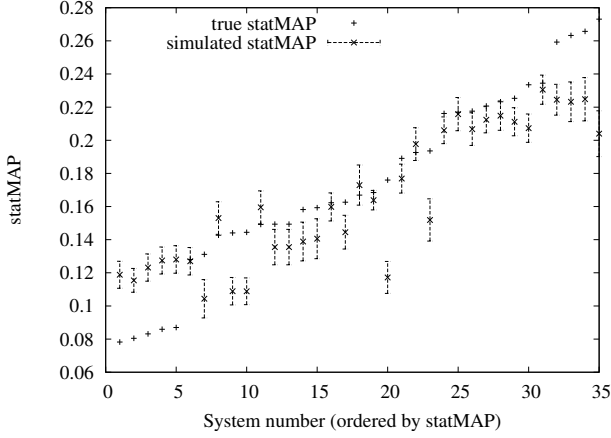
based on pessimistic judgments degrade much more slowly. Figures 3(a) and 3(b) demonstrate this for $\alpha = 1, \beta = 16$ (for the optimist) and $\alpha = 16, \beta = 1$ (for the pessimist). Roughly the same number of judgments changed in both cases, but the effect on performance is much worse when those changes create more relevant documents than when they create more nonrelevant documents. With the pessimistic model, in fact, the correlation is nearly perfect; the scores have simply shifted downward.

The disgruntled and lazy models are similar to the pessimistic model in that they result in fewer relevant documents than exist in the “true” judgments. However, they produce worse results in general. In the disgruntled case (Fig. 3(c), despite labeling roughly the same number of documents relevant as the pessimist, the τ correlations are on average 10% lower. The lazy assessor (Fig. 3(d)) actually found many more relevant documents than either the pessimist or the disgruntled assessor with the same prior parameters, but apparently found “worse” relevant documents than the pessimist, as its τ correlation is lower.

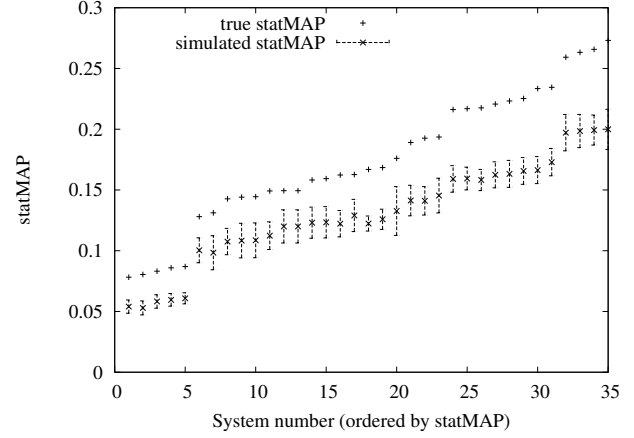
The fatigued and Markovian models are rather similar to each other in how they rerank systems. The Markovian model produces somewhat more pronounced effects on some of the systems, resulting in a lower τ correlation. Both result in more documents being judged relevant.

One conclusion we draw from these results is that it is generally better to underestimate relevance than to overestimate it. The models that result in fewer documents being judged relevant—the pessimist, the disgruntled, and the lazy—generally produce more accurate rankings of systems than those that result in more documents being judged relevant. This suggests that low-cost evaluation methods are sensitive to noise in the relevant documents. Among models that result in fewer relevant documents, the pessimist produces the best rankings overall, though the system scores are strongly biased downward.

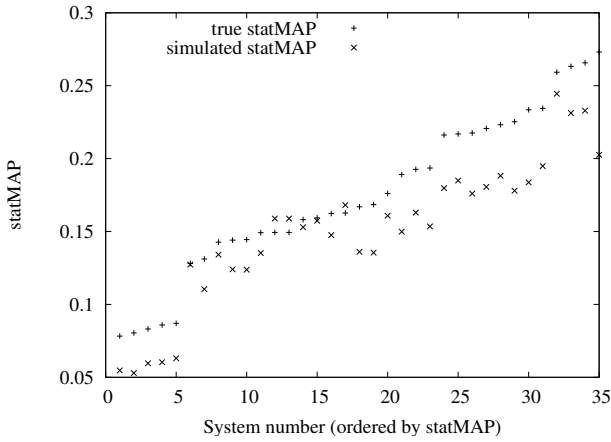
Of course, we do not conclude from this that assessors should be trained to be pessimists. This is an abstract model; the altered judgments had no relationship to any properties of the actual documents apart from their original relevance judgments.



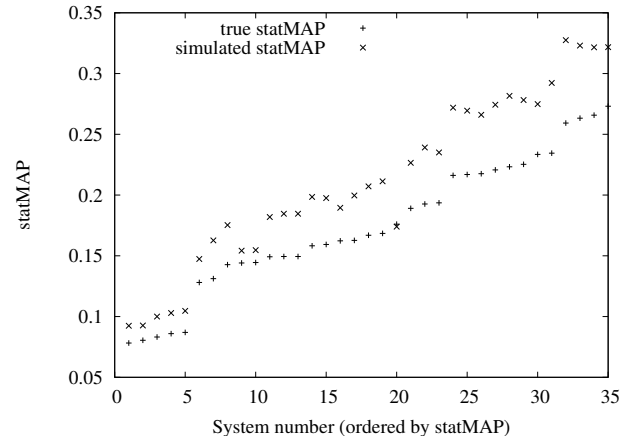
(a) Optimistic assessor ($\alpha = 1, \beta = 16$) judges many more documents relevant. $\tau = 0.72$



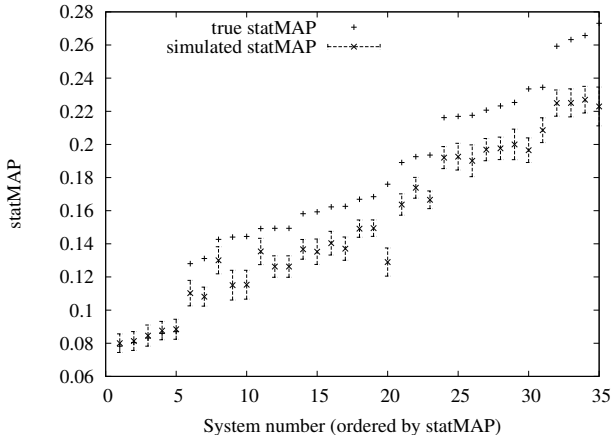
(b) Pessimistic assessor ($\alpha = 16, \beta = 1$) judges many fewer documents relevant. $\tau = 0.92$



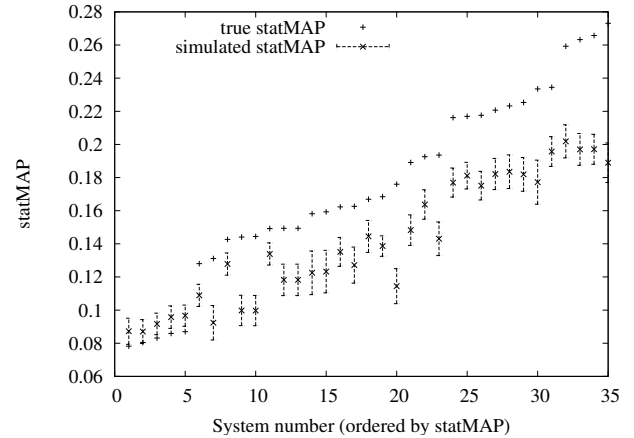
(c) Disgruntled assessor ($\alpha = 1, \beta = 16$) gives up early. $\tau = 0.81$



(d) Lazy assessor ($\alpha = 1, \beta = 16$) assumes first few judgments indicate the rest. $\tau = 0.9$



(e) Fatigued assessor ($\alpha = 0.05, \beta = 1$) becomes more random over time. $\tau = 0.9$



(f) Markov assessor ($\alpha = 1, \beta = 16$) makes each judgment based on the previous one. $\tau = 0.84$

Figure 3: Comparison between “true” statMAP system scores calculated over all Million Query 2009 topics+judgments and statMAP scores after each assessor model is applied to all topics with the specified parameters. The new rankings are evaluated by Kendall’s τ rank correlation (averaged over 25 trials when appropriate).

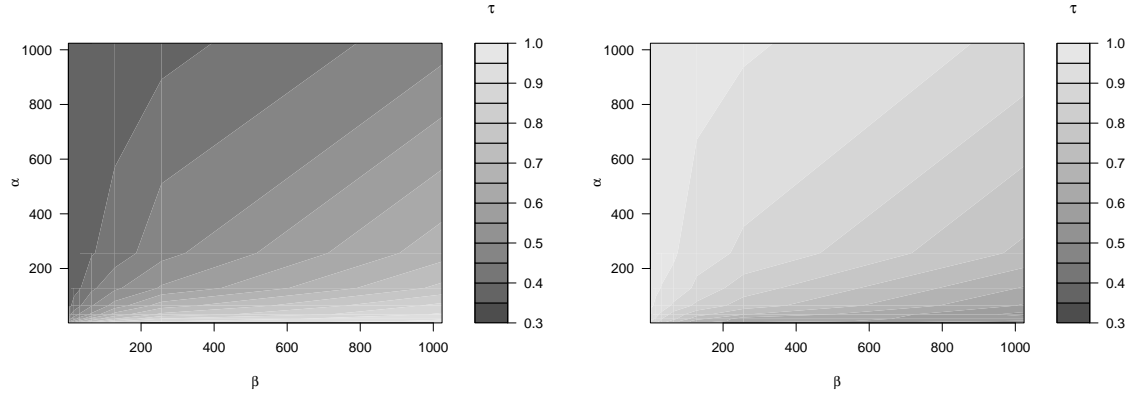


Figure 4: Contour maps illustrating the change in τ with prior parameters α, β in the optimistic model (left) and pessimistic model (right). Lighter areas indicate higher values of τ .

3.4 Worst Case Analysis

The results presented above represent relatively good parameter settings for each of the models. Depending on the prior parameters, the results can become quite bad. Figure 4 shows contour maps demonstrating the change in τ with prior parameters α and β in the optimistic and pessimistic models. The optimist is best (indicated by lighter shading) when α is low and β is very high, which is when it is least likely to incorrectly judge a document relevant. Its performance quickly degrades from there. The pessimist is best when β is low and α is high, which is when it is least likely to incorrectly judge a document nonrelevant, but it maintains good performance until β is high and α is low. Note that the optimistic is much darker in much more of the space than the pessimistic, indicating substantially lower τ correlations for any parameter settings.

Other models are similar to these two. Those that produce more relevant documents than originally existed in the relevance judgments tend to exhibit a faster drop-off in performance when parameters move away from the low-probability regions. Those that produce fewer relevant documents than originally existed tend to exhibit a slower drop-off.

4. ADJUSTING FOR ASSESSOR ERRORS

The effect of assessor errors is to add unplanned variance and bias into the evaluation. This increases the cost indirectly—though the judgments can be made for the same cost, the cost of the errors they introduce adds up. Thus it may be worth expending some extra cost to ensure that errors made by assessors cannot cause too much damage in the aggregate. Here we consider some simple approaches to adjust or correct their errors.

Since we are interested in cases where the assessors may be distributed around the world rather than present in person, and cases with many more assessors judging fewer topics each, we do not want to spend too much time on solutions that involve a great deal of interaction with the assessors.

4.1 Multiple Judgments

One possible solution is to have some documents judged multiple times. The cost clearly depends in part on how many rejudgments are made and how documents are chosen for rejudging, but it also depends on how the extra judgments are incorporated into the evaluation. Some of the

differences observed in the extra judgments will be due to reasonable disagreements about relevance rather than errors. While such differences could possibly be resolved by adjudication, this essentially adds another assessor—one who must be able to make a decision based on conflicting evidence—to the process, and that carries significant cost.

One alternative is to use a simple process like majority vote. If rejudgments converge on a particular decision, it is more likely that the original judgment was in error. This requires more duplicated effort, though, especially since rejudgments themselves are not immune to error.

Along similar lines, since pessimistic models seem to hurt performance less, we could require a supermajority of positives to call a rejudged document relevant. Thus it would take two of two judgments, or two of three judgments, being relevant before we are confident in concluding that a document really is relevant. This would only apply in the cases we actually decide to have a document rejudged; because of that and the additional cost in duplicated effort, the choice of documents to have rejudged must be made very carefully.

We hypothesize that for statAP evaluation, documents with lower inclusion probabilities are better candidates for rejudgment. These documents, if erroneously judged relevant, can have a much greater effect on the evaluation than documents with higher inclusion probabilities. The simulations bear this out: those models that resulted in worse ranking performance had lower inclusion probabilities on average among the judgments that changed. For example, the average inclusion probability among documents that the optimist in Figure 3(a) judged relevant was 0.09, while the average inclusion probability among documents the pessimist in Figure 3(b) judged relevant was 0.12.

To test the effect of rejudging low probability documents, we ran a second simulation to rejudge a few documents with low inclusion probabilities from a prior simulation. In this case, the simulated assessor uses the same model as the original, but only judges documents with inclusion probability less than 0.01. The new judgments are then merged with the existing judgments using the supermajority approach: any document that has been judged relevant twice is considered relevant, while the rest are nonrelevant. Since 90% of the judgments have inclusion probabilities greater than 0.01, most will not change, and most of the judged relevant documents will stay relevant.

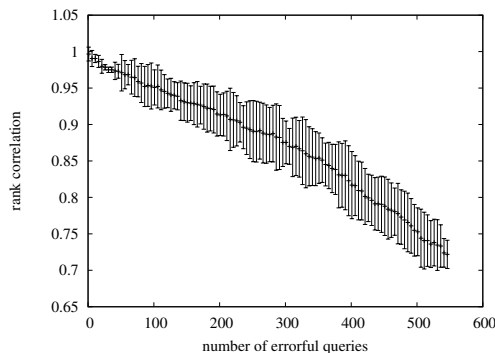


Figure 5: Kendall’s τ decreases linearly (within the given error bars) as the number of errorful topics among in an evaluation increases. Average τ does not fall below 0.9 until 232 of the original judged topics have been replaced with errorful versions.

The effect of this on the optimist is a small improvement in the τ correlation from 0.72 to 0.75, which may not be worth the cost of the extra judgments. However it seems that this could potentially be a useful starting point for selecting documents for rejudging and incorporating the rejudgments into the evaluation.

4.2 Quality Assurance

Another approach to handling erroneous errors is to treat relevance judgments as a quality-assurance problem. Given an estimate of the permissible number of badly-judged topics, we can sample the topics that have been judged and check whether those seem to be errorful in order to estimate the total number of problem cases. If the number is above what is permissible, we can impose tighter controls for a brief time until judging seems to be going smoothly again.

We investigated the permissible number of bad topics by starting with the full evaluation over the original judgments and gradually replacing topics with their errorful doubles from the models above. The goal was to see how many “bad” topics we could inject into the evaluation before we reached a τ correlation below 0.9, the threshold at which we might feel uncomfortable with the ranking.

The result is shown in Figure 5. The decrease in τ is roughly linear in the number of errorful topics, but it does not drop below 0.9 on average until 232—over 40% of the total number of topics—have been replaced. This suggests that statAP is actually fairly robust to errors in judgments, at least in terms of its ability to rank systems. An evaluation could proceed for a fairly long time before tighter controls would need to be enforced.

5. CONCLUSION

We argue that as test collection construction continues to take lower-cost routes away from well-trained, managed assessors to crowdsourcing or cheaper, faster assessors, the errors in evaluation estimates will have to be quantified and potentially adjusted for the errors that will almost certainly occur in judging. We presented eight models of possible errors and showed how each affects an estimate of average precision. We proposed two possible means to adjust for errors: 1) have certain documents selected for rejudging, then

use a voting algorithm to combine the judgments; 2) estimate how many problem cases there seem to be to determine whether judging needs to be more strictly observed.

As a next step, we plan to undertake a true crowdsourcing experiment using Mechanical Turk to investigate the degree to which the behaviors we posit actually occur in that population and the effect resulting errors have on evaluation. Beyond that, future work must consider that these errors will seldom happen independently. Most evaluations will be affected by some mixture of errors, and the parameters of that mixture could have a substantial effect on both the evaluation and adjustments.

6. REFERENCES

- [1] James Allan, Javed A. Aslam, Ben Carterette, Virgil Pavlu, and Evangelos Kanoulas. Overview of the TREC 2008 million query track. In *Proceedings of TREC*, 2008.
- [2] Javed A. Aslam and Virgil Pavlu. A practical sampling strategy for efficient retrieval evaluation, technical report.
- [3] Javed A. Aslam, Virgil Pavlu, and Emine Yilmaz. A statistical method for system evaluation using incomplete judgments. In *Proceedings of SIGIR*, pages 541–548, 2006.
- [4] Peter Bailey, Nick Craswell, Ian Soboroff, Paul Thomas, Arjen P. de Vries, and Emine Yilmaz. Relevance assessment: Are judges exchangeable and does it matter? In *Proceedings of SIGIR*, pages 667–674, 2008.
- [5] Ben Carterette. Robust evaluation of information retrieval systems. In *Proceedings of SIGIR*, 2007.
- [6] Ben Carterette, James Allan, and Ramesh K. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of SIGIR*, pages 268–275, 2006.
- [7] Ben Carterette, Virgil Pavlu, Evangelos Kanoulas, Javed A. Aslam, and James Allan. Evaluation over thousands of queries. In *Proceedings of SIGIR*, pages 651–658, 2008.
- [8] Gordon V. Cormack, Christopher R. Palmer, and Charles L.A. Clarke. Efficient construction of large test collections. In *Proceedings of SIGIR*, pages 282–289, 1998.
- [9] Donna Harman. Overview of the fourth Text REtrieval Conference. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 1–24, 1995. NIST Special Publication 500-236.
- [10] Kenneth A. Kinney, Scott Huffman, and Juting Zhai. How evaluator domain expertise affects search result relevance judgments. In *Proceedings of CIKM*, pages 591–598, 2008.
- [11] Mark Sanderson and Justin Zobel. Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings of SIGIR*, pages 186–193, 2005.
- [12] Ian Soboroff, Charles Nicholas, and Patrick Cahan. Ranking Retrieval Systems without Relevance Judgments. In *Proceedings of SIGIR*, pages 66–73, 2001.
- [13] Ellen Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of SIGIR*, pages 315–323, 1998.
- [14] Ellen M. Voorhees. The philosophy of information retrieval evaluation. In *CLEF ’01: Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, pages 355–370, London, UK, 2002. Springer-Verlag.
- [15] Ellen M. Voorhees and Donna K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.
- [16] Emine Yilmaz and Javed Aslam. Estimating average precision with incomplete and imperfect relevance judgments. In *Proceedings of CIKM*, pages 102–111, 2006.
- [17] Justin Zobel. How Reliable are the Results of Large-Scale Information Retrieval Experiments? In *Proceedings of SIGIR*, pages 307–314, 1998.