Strategic System Comparisons via Targeted Relevance Judgments

Alistair Moffat
Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia
alistair@csse.unimelb.edu.au

William Webber
Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia
wew@csse.unimelb.edu.au

Justin Zobel
Computer Science and Information Technology
RMIT University
Victoria 3001, Australia
jz@cs.rmit.edu.au

ABSTRACT

Relevance judgments are used to compare text retrieval systems. Given a collection of documents and queries, and a set of systems being compared, a standard approach to forming judgments is to manually examine all documents that are highly ranked by any of the systems. However, not all of these relevance judgments provide the same benefit to the final result, particularly if the aim is to identify which systems are best, rather than to fully order them. In this paper we propose new experimental methodologies that can significantly reduce the volume of judgments required in system comparisons. Using rank-biased precision, a recently proposed effectiveness measure, we show that judging around 200 documents for each of 50 queries in a TREC-scale system evaluation containing over 100 runs is sufficient to identify the best systems.

Categories and Subject Descriptors

H.3.4 [Information Storage and Retrieval]: Systems and software – *performance evaluation*.

General Terms

Measurement, performance, experimentation.

1. INTRODUCTION

Text retrieval systems are used to search large collections of documents, and make use of similarity heuristics to identify documents that are likely to be *relevant* responses to user queries. Comparison of the effectiveness of text retrieval systems requires a collection of documents; a collection of queries; and human judgments as to which documents are relevant to which queries. This approach to system evaluation has been used for more than forty years.

Obtaining the necessary relevance judgments can be expensive, and for collections of realistic size it is necessary to be selective about which documents are considered. The standard approach is *pooling*, and has been used in the TREC ad hoc experiments since their inception [Buckley and Voorhees, 2005]. Experiments have shown that pooling is reasonably robust, both as a basis for comparing the contributing systems and, under certain conditions, for evaluating new systems that were developed after the pool was judged [Sanderson and Zobel, 2005]. However, pooling has a significant drawback – the number of judgments required. In the case of the TREC ad hoc experiments, typically a thousand or more judgments are undertaken for each query.

In this paper we propose methods for selecting which documents to judge, based on the motivation that for a given amount of experimental "buck", researchers seek to get the best value in terms of both qualitatively discriminating between competing systems, and at the same time computing quantitative performance benchmarks for the better systems. As was noted by Carterette et al. [2006], judging documents uniformly from the top of the systems' ranked lists accomplishes the second of these two objectives, but is not necessarily the best way of establishing the first.

We explore two different ways in which a set of documents can be selected for judgment as part of a retrieval experiment. In the first family, *static* selection, a list of candidates for judgment are chosen prior to inspection of any documents, based on their importance in the scoring regime. For example, if a document is ranked highly by several of the systems, then judging it is likely to be of considerable benefit in separating those runs from other runs that do not rank it highly. On the other hand, a document that is lowly ranked by a single run is probably not a good candidate for judging. We propose scoring functions that identify the documents that have the greatest potential to discriminate between runs, and set these functions in a context that embraces pooling.

In the second family, the *dynamic* selection methods, the outcomes of judgments already completed also influence the choice as to which documents will be judged next. While this has the potential to introduce assessor bias, our view is that this effect is not likely to be strong. Also, dynamic methods can be applied on a query-by-query static basis, so that each query is judged without feedback, but the evaluation as a whole is adaptive.

Our experiments with TREC data show that the number of judgments required can be greatly reduced, allowing the best of the 129 TREC8 ad-hoc systems to be identified using only two hundred judgments per query – fewer than two judgments per query per run, a significant saving over current practice. These results make use of the *rank-biased precision* (RBP) effectiveness measure proposed by Moffat and Zobel [2005], which is based on a simple model of user behavior and has attractive properties that make it particularly amenable to the treatment proposed here.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

2. EFFECTIVENESS MEASUREMENT

Retrieval experiments

Comparison of retrieval systems can be thought of as requiring four components: documents; queries; judgments; and an effectiveness metric [Saracevic, 1995]. The first component is a collection of documents that are in some way representative of data that might be searched in practice. The second component is a set of queries that might reasonably be applied to that collection. The common thread here is that the most plausible experiments are on real or realistic data; search tasks such as to find the documents on computer science in a collection of chemical abstracts seeded with a small number of articles by Knuth and Dijkstra are unlikely to be persuasive [Tague-Sutcliffe, 1992].

The third component is identification of documents for human relevance assessment. An approach that has become universal is to use a suite of retrieval systems to run each query and thus identify candidate documents for that query, and then to judge some or all of the documents that are so identified.

The fourth component is the process of deciding, given the query runs and the relevance judgments, which systems have provided the most "useful" performance. Typical ways of quantifying system performance involve some combination of *precision* and *recall*; respectively, the fraction of retrieved documents that are relevant, and the fraction of relevant documents that are retrieved.

A key observation is that the fourth stage can directly inform the third: if a decision is made to compare systems based on precision at depth 10, for example, then just 10 documents for each query for each system need to be judged. Our purpose in this paper is to describe principled ways in which that same feedback can be provided for measures like RBP.

Selecting documents for judgment

A simple way of choosing a set of documents for judgment is via a technique known as *pooling*. In pooling, each of the systems under consideration is used to evaluate all of the queries, and the k highest-ranked documents for each query are returned as a *run* for that system. The runs for each query are combined, with duplicates removed, to give a *pool of depth* k for that query. The documents in each query's pool are then judged. This approach constructs a set of relevance judgments that are argued to be reliable and durable [Sanderson and Zobel, 2005, Buckley and Voorhees, 2005].

However, pooling has several disadvantages. One is its vulnerability to faulty systems – most TREC participants have suffered the embarrassment associated with a buggy system that causes the assessors to evaluate thousands of junk documents. Another issue is that, even when systems are performing as they were designed to, pooling is insensitive, and assumes that all systems should be scored to the same level of (apparent) fidelity. A third problem is that pooling is unable to adjust to query-related variabilities – some queries might require more judgments than others.

Problems with Mean Average Precision

The need for large numbers of judgments is in part a consequence of the measures used to score effectiveness. In particular, the metric *mean average precision* (MAP), and similar measures like *bpref* [Buckley and Voorhees, 2004], are not convergent, and are bounded only by 0 and 1. Regardless of their values for a given set of relevance judgments, as more documents are judged, the value of the metric can alter to any value between zero and one. Hence, with these metrics, there is no sense in which doing more judgment work guarantees a higher-fidelity approximation to the underlying behavior of the system being measured.

Another key issue with MAP is that it requires that all of the relevant documents for each query be identified. However, in environments in which only partial judgments are undertaken (via pooling or any other method), only a subset of the relevant documents is identified, meaning that the MAP scores computed (using the size of that subset in the denominator of the calculation) tend to be upper bounds on actual underlying performance. This is unsatisfactory from the perspective of experimental integrity, where we should err on the side of caution, and strive to report lower bounds when we seek to claim something as being "better".

The relationship between MAP and user behavior is also problematic. Is a user actually 100% satisfied if they examine the top-ranked document for a query, find that it is relevant, and then look at another 999 irrelevant documents before they stop? Wouldn't they be happier if the documents ranked in positions 1, 2, and 5 were all relevant, or if they stopped looking after just 10 documents? And can the number of unexamined relevant documents alter the utility the user derives from the relevant documents that were examined?

Central to these hypotheticals is the need for a model of user utility, or *satisfaction*. With reciprocal rank, for example, the measure reflects the effort required by a user who seeks a single answer. Similarly, the metric P@k (precision at depth k) indicates the amount of unit satisfaction a user derives from examining the first k documents in the ranking. On the other hand, there is no plausible search model that corresponds to MAP, because no user knows in advance the number of relevant answers present in the collection they are addressing [Moffat and Zobel, 2005].

Rank-Biased Precision

For web search, studies using technologies such as eye tracking have analyzed typical user behavior; see for example Joachims et al. [2005]. These studies have shown that, even within a page of results, users tend to examine candidate documents in order. That is, users are more likely to examine the document at rank i than the document at rank $i+1$, and have a likelihood of only 50% or so of reaching the fourth-ranked document.

Moffat and Zobel [2005] propose a user model that approximates this behavior, in which users examine documents (or answer snippets) in turn, proceeding from each to the next with probability (or persistence) p, or terminating their search with probability $1-p$. The first document is always examined. The likelihood of inspecting the ith document in the ranking is thus p^{i-1}, and with (say) $p = 0.8$ the likelihood of inspecting the fourth document is about 50%. This model then leads to the notion of *rank-biased precision*:

$$\text{RBP} = (1-p) \cdot \sum_{i=1} u_i \cdot p^{i-1} \,,$$

where $u_i \in [0,1]$ is the (possibly fractional) relevance of the document at position i in the ranking. The normalization factor $1-p$ ensures that RBP values are between zero and one. Moreover, $1/(1-p)$ is the average number of documents examined. Hence, RBP measures the usefulness of a ranking, by quantifying the average per-document-examined utility obtained from the ranking. Actual user behavior is also influenced by factors such as user interface design, browser functionality, response time, and a variety of other factors. Nevertheless, the model underlying RBP provides a reasonable first-order approximation of actual user behavior, and RBP offers several benefits compared to other measures.

For our purposes, a key strength of RBP is that the error due to unjudged documents can be quantified, and that the error converges to zero as more judgments are supplied. With average precision and bpref, while the underlying score can be approximated based on incomplete judgments [Aslam et al., 2006], they can drift anywhere

in $[0, 1]$ as more judgments are performed. In addition, RBP directly supports graded relevance judgments, which are problematic for many existing metrics, MAP included.

As an example of an RBP calculation, suppose that $p = 0.8$, and in a given run the series of relevance judgments is 0, 1, 1, 0, 0, 1, ?, 0, 0, 1, in which the seventh document and all documents after the tenth are unjudged. Then, based on known relevance, RBP is calculated as $(1 - 0.8) \times (0.8^1 + 0.8^2 + 0.8^5 + 0.8^9) = 0.380$. But the unjudged documents, if all relevant, can contribute a further $(1 - 0.8) \times 0.8^6 + (1 - 0.8) \times \sum_{i=11}^{\infty} 0.8^{i-1} = (1 - 0.8) \times 0.8^6 + 0.8^{10} = 0.160$ to the score. Thus 0.380 is the *base* RBP value; and the *residual* of 0.160 means that with further judgments RBP could, in principle, reach 0.540. Going to deeper judgments reduces the residual: if the top 21 documents in the run are all judged and $p = 0.8$ is used, the residual is 0.01, and the calculated RBP score will be precise to two decimal digits of accuracy. That is, both the base and residual should be reported in any experimental output describing the run, or the experiment should be designed and reported in such a way that the residual can be inferred.

As a lengthening prefix of the run is judged, the sequence of RBP values obtained from the partial judgments can be thought of as being an increasingly accurate lower bound to the underlying score that would be obtained were exhaustive judgments to be available. Note also that RBP errs on the side of caution. More judgment effort is guaranteed both to not decrease a given base RBP score, and also to decrease the uncertainty associated with that score.

Statistical testing

A key element of good experimental methodology is statistical testing, to determine the likelihood that the observed relativity is the result of chance rather than system superiority. In particular, the IR community expects that claimed performance improvements be supported by sound experiments on plausible data against a realistic baseline system, and that confidence scores be given.

We agree wholeheartedly with these expectations. However we also comment that doing careful statistical tests on data in which the inherent uncertainty is not quantified is not good experimental practice. In this context, the fact that MAP values have unknown uncertainty makes them a poor input to a statistical test. It is possible to do good statistics on bad data to derive unsubstantiated conclusions. Our primary concern in this paper is how to obtain high-quality data – with the level of inherent uncertainty clearly quantified – that can then be fed into a statistical test. One of the experiments described below is intended to highlight the risks associated with casual use of statistical testing.

3. STATIC JUDGMENT ORDERINGS

Given RBP as an evaluation metric, we now consider how best to choose a pool of candidate documents for judging, so as to maximize the independent goals of both differentiating between systems, and allowing accurate estimation of the underlying scores.

To minimize the level of subscripting, we assume in our formulae that a single query is being considered, but all of the methods can be immediately generalized to environments of more than one query. We suppose that S systems are engaged in the evaluation (numbered from 1 to S), that an appropriate value of p has been set as part of the experimental design, and that each system has generated a ranking. Let $b_{s,d}$ be the rank position at which system s has placed document d, and take $b_{s,d}$ to be ∞ if d does not appear in the ranking generated by system s. Then define

$$c_{s,d} = (1 - p)p^{b_{s,d}-1}$$

to be the weight of the contribution of document d in system s for some parameter p.

Pooling

Given this notation, pooling is described by assuming that a weight

$$w_d = \max_{1 \le s \le S} c_{s,d}$$

is calculated for each document d in the collection, and then that judgment candidates d are selected in decreasing order of w_d, with ties broken arbitrarily. In a multiple-query evaluation, w_d ordering is applied across documents from all queries, with the result that different queries may receive different numbers of judgments.

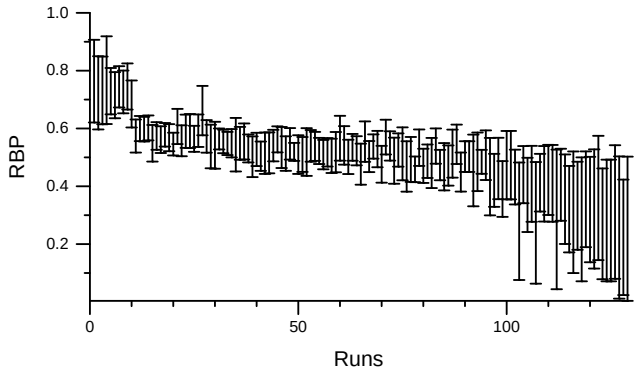
The top part of Table 1 gives four example runs. In the pooling approach, documents would be selected for judgment in the order 18, 22, 21, 10, 35, 15, 11, 16, 13, and so on, until the total pool capacity had been filled, or until some predetermined depth had been reached in each ranking. Note that, once the set of candidate documents has been selected, they can be judged in any order.

Figure 1(a) shows the effect that pooling has on the RBP residual errors of a group of retrieval systems. Each of the bars in the graph represents one of the 129 system *runs* that were assessed at TREC8 in the 1999 round of experimentation; see `http://trec.nist.gov`. The horizontal ordering of the runs is based on their official MAP scores, from highest to lowest. The same ordering is used in all of these figures in this paper; as can be seen, RBP using $p = 0.8$ is correlated with official MAP, but not perfectly so. The range spanned by the bar for each run s represents the base, $base_s$, and upper bounds on RBP score, $base_s + r_s$. A (small) pool of $5{,}000$ judgments across the 129 runs and 50 queries is used throughout Figure 1, so that the trends in the residuals can be observed.

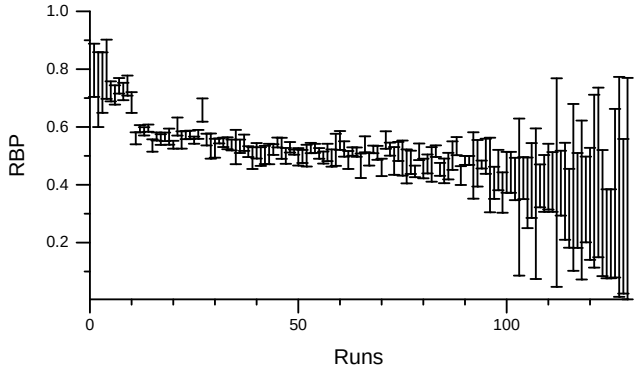
The judgments used to draw these graph are a subset of the official "qrels" file for TREC8, with judgments taken into the subset only when the corresponding document is identified by the methodology and parameter combination being tested. That is, while we did not create any fresh judgments, the evaluations reported reflect what would have been observed had our methodologies been used in the TREC8 evaluation in 1999. An exception arises if any candidate generation strategy called for the judgment of a document for which (in the TREC8 experiment) no outcome had been recorded. In this case we deemed that document to be "not relevant", as is the case in standard TREC-based evaluations. Note also that we evaluated all techniques first on the TREC5 judgments, and then applied them to the TREC8 data without further tuning in order to obtain the results presented here. That is, the results on the TREC5 data set were used as training, and TREC8 reserved for experimentation.

The TREC8 qrels file contains $86{,}830$ judgments, an average of $1{,}737$ per query, formed via a pooling approach (to depth 100) over a subset of the 129 contributing runs. Up to two runs per research group were used as the basis for the pool, making a total of 71 pooled runs. In forming the official TREC pools, the top 100 documents for each run and query for pooling were determined according to the scores provided in the run itself, with ties ordered by document number. Of the judgments, an average of 95 per query indicate relevance, with the maximum number of relevant documents for a query being 347, and the minimum being 6.

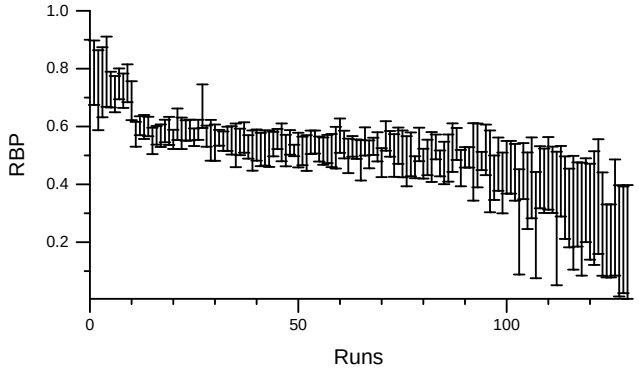
In Figure 1(a), a set of $5{,}000$ judgments are selected using the pooling approach, an average of (but not necessarily exactly) 100 per query, and equivalent to working with a pool depth of $k \approx 3$ across the 129 runs. As can be seen from the graph, at this level of effort spent judging, the overall system ordering can be observed in general terms, but there is considerable imprecision remaining in all of the RBP scores, and it is not possible to establish whether any of the systems in the top grouping is clearly superior to the



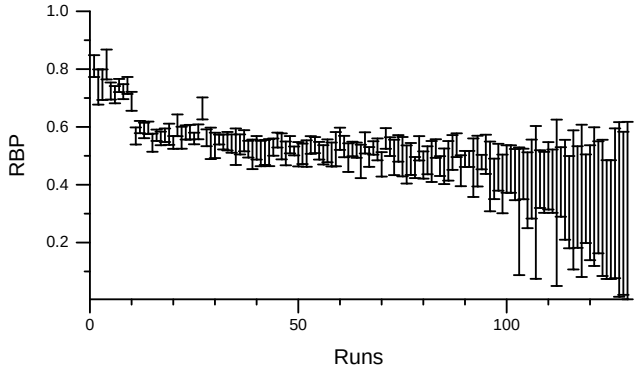
(a) Pooling



(b) Method A: RBP weight



(c) Method B: RBP weight and residual



(d) Method C: RBP weight, residual, and cubed midpoint

Figure 1: Measured RBP score ranges for TREC8 runs after $5{,}000$ document judgments are selected by different methods, using $p = 0.8$. Runs are ordered by decreasing official MAP score.

	0.20	0.16	0.13	0.10	0.08	0.07	0.05	0.04	
Run 1	**18**	**22**	15	**13**	**11**	25	**10**	84	···
Run 2	**22**	**10**	**11**	19	38	**18**	33	17	···
Run 3	**21**	35	16	**11**	38	33	**18**	17	···
Run 4	**10**	**18**	**11**	**22**	87	**13**	17	20	···

Doc.	18	22	11	10	21	13	···
Wght.	0.48	0.46	0.44	0.41	0.20	0.17	···

Run	Judged	Total	Residual
1	0.20+0.16+0.10+0.08+0.05	0.59	0.41
2	0.20+0.16+0.13+0.07	0.56	0.44
3	0.20+0.10+0.05	0.35	0.65
4	0.20+0.16+0.13+0.10+0.07	0.66	0.34

Table 1: The first eight elements from four systems (runs), together with the first six run-combined document weights calculated according to Method A that result when $p = 0.8$.

others. More than a dozen systems could, at the upper limits of their ranges, be better than System 1.

Method A: Summing contributions

Our first suggested alternative method is that documents be selected based on their overall contribution to the effectiveness evaluation, rather than their peak contribution. We now define

$$w_d = \sum_{1 \le s \le S} c_{s,d}\,,$$

being the sum of the RBP residuals associated with document d for a given value of the persistence parameter p. The documents are then sorted by decreasing w_d, and the required number are selected as candidates and judged. As with pooling, where multiple queries are being used in the evaluation, w_d ordering is applied across documents from all queries, allowing the average residual to be globally minimized.

Table 1 shows the Method A computation applied to four example systems, and computes the total weight of each of the documents, as a sum of their weights in the runs, assuming that $p = 0.8$. Document 18 has the highest total weight, and is the first to enter the judgment pool, followed by 22, 11, 10, 21, and 13. In this example, Run 1 gets five of its documents judged when the pool contains six judgments, while Run 3 has only three documents judged. The average residual over the four runs is 0.49.

Method A has the immediate benefit of downplaying the cost of erroneous runs. In the pooling arrangement, a faulty system is disproportionately expensive, because it introduces a large number of irrelevant documents, the judgment of which is of no assistance in the scoring of any other run. In Method A a rogue run still causes problems, but documents from that run are less likely to get amplified by references from other runs.

Figure 1(b) shows the way in which Method A reduces the residuals for some runs, notably those at the center of the MAP-ordered arrangement. This reduction is at the expense primarily of the runs at the bottom of the system ordering, but with some of the runs at the top of the ordering also having larger residuals. That is, while poorly performing runs are often composed of documents not reported by other systems, so too are some highly scoring runs.

Method B: Weighting by residual

While it maximally reduces the average residual, Method A has the drawback of leaving individual run residuals at widely varying levels, which may be undesirable in some situations. For example, in Table 1, Run 3 has a residual nearly twice as large as Run 4. This discrepancy arises because Run 3 has several documents near the head of its ranking that do not appear in other runs, and so do not get reinforced sufficiently to get judged.

To compensate for this effect, we propose Method B, in which each RBP contribution is also weighted by the residual r_s of the run s that it is coming from:

$$w_d = \sum_{1 \le s \le S} c_{s,d} \cdot r_s \, ,$$

where r_s is the current residual for Run s. The effect of the added factor is that runs with high r_s values are more likely to get documents judged. Note that r_s varies as candidate documents are inspected, but does not depend on the outcome of those judgments.

Figure 1(c) shows the effect that this additional change has on the set of TREC8 base-vs-residual pairs when $p = 0.80$ and $5{,}000$ judgments are undertaken. Compared to Figure 1(b), error ranges are somewhat diminished at the bottom of the performance spectrum, but at the cost of increased score ranges elsewhere, including the critical region at the left of the graph.

A key point brought out by the first three graphs in Figure 1 is that bad runs and good runs share a common propensity to introduce documents not proposed by other mechanisms, and that it is difficult in a static and pre-identified judgment pool to differentiate between the two. This point is the theme of Section 4.

4. ADAPTIVE METHODS

So far, we have only considered static methods for deciding the set of documents to be judged. But it is also possible to make judgment choices adaptively, guided by the results of previous judgments. Adaptive methods are particularly valuable when there are many contributing runs of variable quality. As judging proceeds, an indication of the retrieval performance of the different runs emerges, and favoring the higher-scoring runs with further judgments is the next tactic we explore. Judging resources can then be progressively concentrated on identifying and distinguishing the top-performing runs, and less effort can be spent on determining the precise scores and relativities of the poorer systems.

Weighting by predicted score

To employ an adaptive approach, a way of estimating the final RBP score from a current $[\mathit{base}_s, \mathit{base}_s + r_s]$ range pair is required. Simplest is to split the difference, and take $\mathit{base}_s + r_s/2$ as an estimate of retrieval effectiveness. It is then straightforward to multiply each RBP contribution (used in Method A) by both the residual r_s (factored in to obtain Method B) and the estimated RBP value:

$$c_{s,d} \cdot r_s \cdot (\mathit{base}_s + r_s/2) \, .$$

However, during our preliminary experimentation on the TREC5 data (not reported here), this approach to calculating w_d was relatively ineffective, and represented only a slight improvement on Method B. The problem is that the range midpoints are relatively similar – looking at Figure 1(c), for example, the midpoints vary from about 0.8 down to about 0.4, which is too small a factor to have any great effect.

Method C: Raising the power

To boost the strength of the current-score component of the estimator, we also experimented with increasing powers. On the TREC5 data, cubing the estimated RBP value gave a marked increase in performance compared to squaring it or using it directly, and it is this approach that we denote as Method C:

$$w_d = \sum_{1 \le s \le S} c_{s,d} \cdot r_s \cdot (\mathit{base}_s + r_s/2)^3 \, .$$

Figure 1(d) shows that even with just $5{,}000$ judgments it is possible to get reasonable accuracy at the top of a 129-system comparison. The same top group of high-scoring runs is in evidence, but with increasing confidence it is possible to say that Run 1 and Run 4 have the highest effectiveness scores. (Note that while MAP and RBP tend to be strongly correlated, there is no expectation that they should generate identical orderings.)

RBP Projections

Another interesting possibility is to extrapolate from the known base_s and r_s assuming that the unjudged documents are found to be relevant at the same rate as the judged ones. This is a reasonable estimate, since the unjudged documents tend on average to be lower in the system rankings than the judged ones, and, unless a system has particularly perverse behavior, the probability of a system identifying a relevant document is non-increasing down the ranking. That is, taking the midpoint of a range $[\mathit{base}_s, \mathit{base}_s + r_s]$ as an estimate of final RBP is a zero-order estimate; and taking a first-order estimate, based on extrapolation, yields the computation

$$\text{projected-RBP} = \mathit{base}_s + r_s \frac{\mathit{base}_s}{1 - r_s} \, ,$$

with $\mathit{base}_s/(1 - r_s)$ the weighted average fraction of judged documents that are relevant.

The advantages of using projections can be seen in Figure 2, which shows both forms of range for RBP with Method C after 10,000 judgments. (At 10,000 judgments, the error ranges are much reduced for all methods, but Method C remains clearly superior.) In our tests, the projected upper bound on the RBP value at all judgments depths was always greater than the "final" lower bound after all 86,830 judgments. At 10,000 judgments, this projected value is little different from the full range for the better runs, but shows that, for the weaker runs, the upper bounds are drastic overestimates. All the ranges are now small, and it seems likely that adding judgments will not substantially further separate the runs.

5. EXPERIMENTAL RESULTS

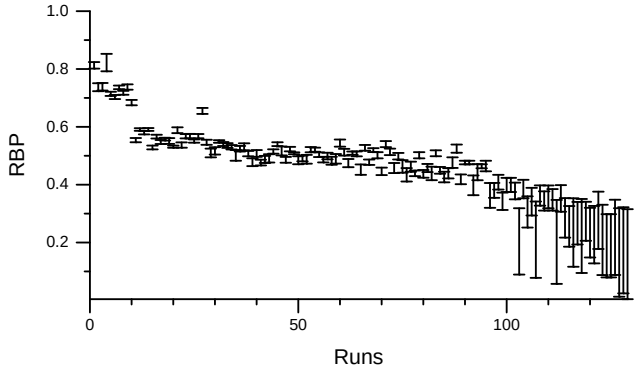
Recognizing that the graphs in Figures 1 and 2 provide qualitative indications of the usefulness of our approach, but not quantitative validation, we now present a range of numeric results.

Relevant documents

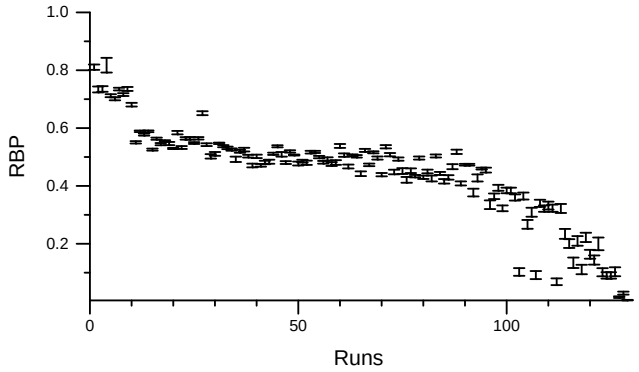
Table 2 shows the number of relevant documents identified by each of the methods, again using the TREC8 data. The biased methods are better than pooling at identifying relevant documents, and it is the relevant documents that provide the positive data points in effectiveness measurements. At $5{,}000$ judgments, for example, Method C identifies over 30% more relevant documents than does pooling. Note that there is an experimental artifact that has been compensated for in Table 2 – with only a finite number of documents actually judged, at larger pool sizes the various methods increasingly ask for judgments that the TREC8 data is unable to supply. In forming Table 2, these "unknown outcome" documents

Method	Number of judgements				
	1,000	2,000	5,000	10,000	20,000
Pooling	359 : 1,032	594 : 2,141	1,097 : 5,502	1,703 : 11,331	2,495 : 23,231
Method A	511 : 1,000	825 : 2,001	1,352 : 5,209	1,842 : 11,088	2,546 : 23,093
Method B	516 : 1,001	746 : 2,104	1,199 : 5,543	1,761 : 11,410	2,538 : 23,314
Method C	550 : 1,000	895 : 2,008	1,440 : 5,296	2,028 : 10,915	2,839 : 22,491

Table 2: Number of relevant documents identified in TREC8 using different methods for selecting candidate documents for judging, at different numbers of judgements performed. In this table (and only in this table), documents that are unjudged by the TREC8 assessors are bypassed rather than deemed to be irrelevant. The second number in each pair represents the total number of documents considered (relevant plus irrelevant plus bypassed) in order to obtain the given number of documents for which TREC8 judgements were available.



(a) The full RBP range



(b) The projected RBP range

Figure 2: RBP score ranges for TREC8 runs after 10,000 document judgments selected by Method C (adaptive cubicly biased total document weight), using $p = 0.8$. Runs are ordered by decreasing official MAP score.

are bypassed, rather then deemed to be irrelevant. The numbers after the colon in each cell in the table indicate the total number of documents handled in order to obtain the desired number of relevance judgments, including the ones bypassed.

Re-usability of judgments

One potential problem with the approach we have proposed is that a value of p is required at the time the judgments are performed, making it an attribute of the experiment as a whole, rather than of just the final evaluation phase. Table 3 shows the extent to which the candidate documents selected using one value of p and Method C would also have been selected if a different value of p was in use. There is quite marked difference in the various pools of documents,

p	0.50	0.80	0.95
0.50	10,000	9,215	6,972
0.80	—	10,000	7,517
0.95	—	—	10,000

Table 3: Overlap in relevance judgments when the value of p is varied. Each entry records the number of common judgment performed when $k = 10{,}000$, Method C is used, and p is varied.

Judgments set	Effectiveness evaluation		
	$p = 0.50$	$p = 0.80$	$p = 0.95$
Method C, $p = 0.50$	0.0006	0.0270	0.2340
Method C, $p = 0.80$	0.0007	0.0190	0.1938
Method C, $p = 0.95$	0.0134	0.0293	0.1309
Pooling	0.0022	0.0636	0.3226

Table 4: Effect on average residual of the choice of p made at the time document candidates are chosen, for the best one-third of the TREC8 runs. Each number is the average observed residual over 43 runs, when the RBP metric is evaluated using the value of p indicated in the column heading. Four different sets of 10,000 judgments are used, three of them chosen using Method C.

with as many as 30% of the documents in the judgment pool changing as p is raised from 0.5 to 0.95.

To balance that outcome, Table 4 shows the average residual over the top third of the TREC8 runs, presuming that these are the ones of greatest interest. Three p-differing 10,000-element judgment sets are formed, in each case using Method C; and then for each judgment set, actual performance is evaluated using the same three different values of p. The values down the diagonal show that average residual for "good" runs is minimized when the judgments-p matches the evaluation-p, but the off-diagonal values are also small enough that the difference in judgment sets is not critical. The last row of the table shows the same results for a set of 10,000 judgments formed by pooling; in all except one combination the Method C judgment sets give smaller average residuals.

Establishing confidence

It was noted above that confidence tests are an important part of any system comparison. Table 5 considers the top third of the TREC8 runs, and for each of the $43 \times 42/2 = 903$ pairwise system comparisons that are possible within that group, considers the hypothesis that the system with the higher base RBP value is better than the one with the lower base RBP value. Each entry in the first column records the fraction of these 903 "base to base" pairwise relativities that can be established at the 95% level when the 50 TREC8

Judgements	Fraction greater than 0.95 confidence		
	base–base	base–top	base–proj
5,000-Pooling	0.443	0.033	0.151
5,000-Meth A	0.484	0.313	0.367
5,000-Meth B	0.465	0.130	0.272
5,000-Meth C	0.504	0.336	0.398
10,000-Pooling	0.501	0.379	0.414
10,000-Meth A	0.499	0.427	0.463
10,000-Meth B	0.499	0.379	0.422
10,000-Meth C	0.499	0.430	0.467
20,000-Pooling	0.502	0.476	0.491
20,000-Meth A	0.503	0.485	0.496
20,000-Meth B	0.503	0.473	0.492
20,000-Meth C	0.503	0.493	0.499
86,130 RBP	0.502	0.498	0.502
86,130 MAP	0.421	—	—

Table 5: Fraction of 903 possible pairwise system comparisons between the top 43 runs (based on official MAP score) for TREC8 that are determined to be significant differences at the 0.95 confidence level, using a one-tailed paired Wilcoxon test. Comparisons labelled "base–base" are between the RBP base values of both runs; "base–top" compares the base values of one run with the top of the residual range of the second; and "base–proj" compares base values against projected RBP values. Four different methods for determining the candidate documents are used, and four different sizes of the judgement set. Except for the last row, all evaluations use RBP with $p = 0.80$ as the effectiveness metric.

queries are used. This fraction is relatively stable, as the size of the judgments set, and the method of forming it, are varied.

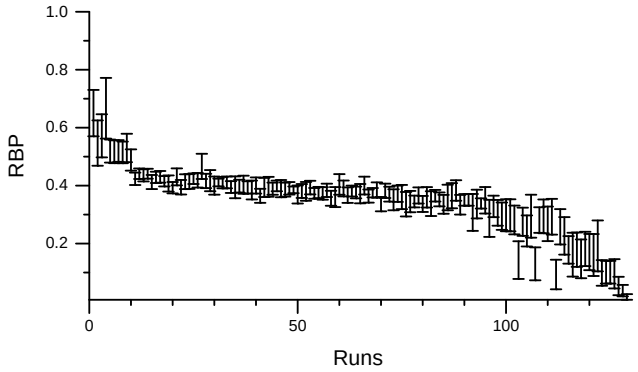
The middle column shows the fraction of times that statistical significance can be established when comparing "base of one run, taking *base*$_s$" against "top of the other run, taking *base*$_s + r_s$", giving unambiguous superiority. Now the number of judgments plays a key role, and, when the judgment set size is limited, the method used to form the judgment set is also a factor. The final column performs the same test, but between the base of one run's range and the projected RBP value in the other.

Base-vs-base comparisons are the ones most likely to lead to confidence in the comparison, but involve the least defensible use of the data. Claiming high levels of confidence based on just 5,000 judgments is statistically valid, but not sensible in practice. A more cautious approach is to perform base-vs-top comparisons, or base-vs-projected comparisons. Base-vs-base testing is the only option possible if MAP is used as the effectiveness metric.

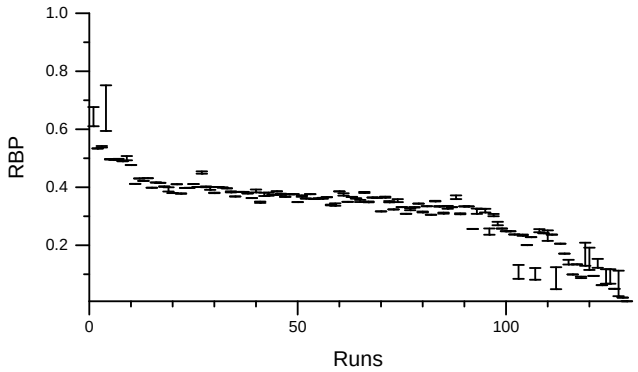
In the limit

Figure 3(a) shows what happens when a more persistent user is assumed, with an evaluation-p of 0.95, and can be compared with Figure 2(b), which used $p = 0.80$. This higher value of p measures effectiveness in a way that more closely matches MAP. Even when all available judgments are used (Figure 3(b)), there are still non-trivial residuals for certain key runs – several do not contribute documents into the official TREC8 pool, and two key high-scoring runs, which supplied than 100 answer documents for each query, have significant RBP residuals at this larger value of p.

The message of Figure 3 is clear – even when quite extensive judgments are available, system relativities should not be assumed by comparing base RBP values without also at least calculating the residuals. This level of experimental care is not possible for MAP.



(a) With $p = 0.95$, Method C, 10,000 judgments



(b) With $p = 0.95$, and all 86,830 TREC8 judgments

Figure 3: Projected RBP score ranges for TREC8 runs with $p = 0.95$ and (in part (a)) Method C used. Runs are ordered by decreasing official MAP score.

6. RELATED WORK

There have been numerous proposed evaluation metrics over the history of information retrieval, and new metrics continue to be proposed. Much of the early work in the area is summarized in a special issue of Information Processing & Management (see for example Harman [1992] and Salton [1992]). The history of MAP is reviewed by Buckley and Voorhees [2005]. Moffat and Zobel [2005] compare RBP to other evaluation metrics, including *discounted cumulative gain* [Järvelin and Kekäläinen, 2002], compared to which RBP has the significant advantage that the error is bounded and the size of the judgments set is not a parameter.

Several papers have examined issues arising from the TREC experimental methodology, such as the extent to which system comparisons can be extrapolated to unseen queries [Zobel, 1998, Buckley and Voorhees, 2000, Sanderson and Zobel, 2005]. Another issue considered in these papers is whether the judgments allow scoring of new systems, a topic also examined by Voorhees [2001], who reviews pooling and its strengths and limitations. Being able to quantify residuals provides key information in this regard.

A closely related work to ours is that of Cormack et al. [1998], who propose an interactive judgment process (adaptive, in our terminology) in which documents suggested by systems that are successful are favored. The aim of this approach to pooling is to maximize the number of relevant documents found, a worthy aim that does not necessarily reduce the uncertainty in measured effectiveness, as queries with few relevant documents may remain largely unjudged. Voorhees [2001] raises the concern that this style of document selection can lead to bias in the document pool "toward

systems that retrieve relevant documents early in their rankings" (p. 363). In response, we note that most effectiveness metrics, including both MAP and RBP, have exactly the same bias.

Aslam et al. [2006] (see also Yilmaz and Aslam [2006]) propose unbiased sampling from a large pool, rather than biased choice based on impact on the effectiveness measure. In the context of MAP, such an approach requires that the relevance of the unjudged documents be inferred, which may be sound on average but means that the performance of individual queries has high uncertainty. Their method allows estimation of final MAP values from small sets of documents, and might also be applied to projected RBP scores to tighten the error bounds and further reduce the number of judgments required.

Our work has been in the context of large, TREC-style evaluations. At the other end of the spectrum, researchers may wish to rapidly compare a small number of systems, a problem addressed by Carterette et al. [2006]. Using MAP, they give a pairwise method for finding documents that have the greatest potential to distinguish between systems. Their experiments found around 2,000 judgments sufficient to distinguish between eight systems on 50 queries, but due to the properties of MAP, absolute scores could not be computed. Their approach also assumes that relevance rankings are taken only to a fixed depth, and that the probability of relevance of unjudged documents can be estimated as a constant.

Cormack and Lynam [2006] use statistical methods to predict confidence intervals for MAP, and have made progress towards resolving some of the issues that are also addressed by RBP. However, MAP remains difficult to use as an input into selection methods, due to the fact that the absolute impact any particular document has on measured effectiveness is unknown.

7. CONCLUSION

We have proposed new methods for selecting documents to be judged when comparing a set of retrieval systems. These methods select those documents that best reduce the current uncertainty in the measured effectiveness, with (in Method C) a bias towards systems that are scoring well. Our experiments with more than 4100 TREC8 runs show that the new approaches provide rapid identification of the best performing systems, and shift the judgment effort away from the runs of weaker systems. Overall, an average of 200 judgments per query is sufficient to produce good bounds on the effectiveness of the competitive systems.

The basis of our new methods is use of the RBP rank-biased precision metric, in which uncertainty due to unjudged documents is precisely quantified. With calculable bounds on the effectiveness that would be determined with complete judgments, we also demonstrated that RBP values can more reliably be used for significance tests than can MAP scores.

Acknowledgment. This work was supported by the Australian Research Council, and by the NICTA Victoria Laboratory.

References

J. A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In Dumais et al. [2006], pages 541–548.

C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In N. J. Belkin, P. Ingwersen, and M.-K. Leong, editors, *Proc. Twenty-Third Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 33–40, Athens, Greece, September 2000. ACM Press, New York.

C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In M. Sanderson, K. Järvelin, J. Allan, and P. Bruza, editors, *Proc. Twenty-Seventh Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 25–32, Sheffield, England, August 2004. ACM Press, New York.

C. Buckley and E. M. Voorhees. Retrieval system evaluation. In *TREC: Experiment and Evaluation in Information Retrieval*, chapter 3, pages 53–75. MIT Press, Cambridge, Massachusetts, 2005.

B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In Dumais et al. [2006], pages 268–275.

G. V. Cormack and T. R. Lynam. Statistical precision of information retrieval evaluation. In Dumais et al. [2006], pages 533–540.

Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. Efficient construction of large test collections. In Croft et al. [1998], pages 282–289.

W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors. *Proc. Twenty-First Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval*, Melbourne, Australia, August 1998. ACM Press, New York.

S. Dumais, E. N. Efthimiadis, D. Hawking, and K. Järvelin, editors. *Proc. Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, August 2006. ACM Press, New York.

D. Harman. Evaluation issues in information retrieval. *Information Processing & Management*, 28(4):439–440, 1992.

K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In Marchionini et al. [2005], pages 154–161.

G. Marchionini, A. Moffat, J. Tate, R. Baeza-Yates, and N. Ziviani, editors. *Proc. Twenty-Eighth Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval*, Salvador, Brazil, August 2005. ACM Press, New York.

A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. October 2005. Submitted; preprint circulated by the authors.

G. Salton. The state of retrieval system evaluation. *Information Processing & Management*, 28(4):441–449, 1992.

M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In Marchionini et al. [2005], pages 162–169.

T. Saracevic. Evaluation of evaluation in information retrieval. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proc. Eighteenth Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 138–146, Seattle, Washington, July 1995. ACM Press, New York.

J. Tague-Sutcliffe. The pragmatics of information retrieval experimentation, revisited. *Information Processing & Management*, 28(4):467–490, 1992.

E. M. Voorhees. The philosophy of information retrieval evaluation. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Proc. 2001 Cross Language Evaluation Forum Workshop*, pages 355–370, Darmstadt, Germany, September 2001. LNCS volume 2406.

E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proc. Fifteenth ACM International Conference on Information and Knowledge Management*, pages 102–111, Arlington, Virginia, USA, November 2006.

J. Zobel. How reliable are the results of large-scale information retrieval experiments? In Croft et al. [1998], pages 307–314.