Less is More

Probabilistic Models for Retrieving Fewer Relevant Documents

Harr Chen
MIT CSAIL
Cambridge, MA 02139, USA
harr@csail.mit.edu

David R. Karger
MIT CSAIL
Cambridge, MA 02139, USA
karger@csail.mit.edu

ABSTRACT

Traditionally, information retrieval systems aim to maximize the number of relevant documents returned to a user within some window of the top. For that goal, the *probability ranking principle*, which ranks documents in decreasing order of probability of relevance, is provably optimal. However, there are many scenarios in which that ranking does not optimize for the user's information need. One example is when the user would be satisfied with *some* limited number of relevant documents, rather than needing *all* relevant documents. We show that in such a scenario, an attempt to return *many* relevant documents can actually reduce the chances of finding *any* relevant documents.

We consider a number of information retrieval metrics from the literature, including the rank of the first relevant result, the %no metric that penalizes a system only for retrieving *no* relevant results near the top, and the diversity of retrieved results when queries have multiple interpretations. We observe that given a probabilistic model of relevance, it is appropriate to rank so as to directly optimize these metrics in expectation. While doing so may be computationally intractable, we show that a simple greedy optimization algorithm that approximately optimizes the given objectives produces rankings for TREC queries that outperform the standard approach based on the probability ranking principle.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Retrieval Models*

General Terms: Algorithms

Keywords: Information Retrieval, Formal Models, Machine Learning, Subtopic Retrieval

1. INTRODUCTION

It is a common rule of thumb in that the *Probability Ranking Principle* (PRP) is "optimal." Under reasonable assumptions, one can prove that ranking documents in descending order by their probability of relevance yields the maximum expected number of relevant documents, and thus maximizes the expected values of the well known precision and recall metrics [14].

Simply returning as many relevant documents as possible, however, is not the only possible goal. For example, since a single relevant result often provides "the answer" to the user's query, we might be concerned only with whether our system returns *any* relevant results near the top. This is plausible for question answering, or for finding a homepage. It also captures a notion of "bare minimum success" that can be meaningful for hard queries. The TREC robust track, focusing on such hard queries, defines and uses the *%no* metric—the fraction of test queries on which a system returned no relevant results in the top ten [20].

As we shall argue below, the probability ranking principle is *not optimal* in such a case. For if the system's model of relevance is wrong, it will be wrong over and over again, returning an entire list of irrelevant documents—one might say that the expected number of relevant documents is large, but the variance in the outcome is also high. Similarly, it is common wisdom that some queries such as "Trojan horse" can express multiple, distinct information needs—about a type of malware, a Trojan war artifact, or other minor usages. A PRP-based approach may choose one "most likely" interpretation of the query, and provide results that satisfy only that interpretation, leaving users with rarer interpretations unsatisfied.

Given that we have stated a clear metric (success in finding at least one relevant document) we argue that under a probabilistic model of document relevance, there is a particularly natural approach to designing a retrieval algorithm for it—namely, to rank documents so as to *optimize the expected value of the metric*. In particular, we should rank so as to *maximize the probability of finding a relevant document among the top n*. While exactly optimizing this quantity is NP-hard, we derive a greedy heuristic for approximately optimizing it. Intriguingly, our greedy algorithm can be seen as a kind of blind *negative* relevance feedback, in which we fill each position in the ranking by assuming that all previous documents in the ranking are *not* relevant.

We demonstrate that our approach is effective in practice. We evaluate the performance of our greedy algorithm on queries from various TREC corpora. We show that it retrieves at least one relevant document more often than the traditional ranking (with statistical significance). We give special attention to the robust track, where one of the goals is to minimize the chance of returning no relevant results, and show that our algorithm does well.

In addition to the robust track's %no metric, we consider a number of other standard metrics from the literature. For example, we might be interested in how far down the ranked result list we must go to find the first relevant document. The *search length* (SL) [4] and *reciprocal rank* (RR) [15] metrics measure this quantity in different ways. On the other hand, if we believe that the results of a query may have different "subtopics" (facets of the query) or that multiple queriers might have different relevance judgments, we

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.
Copyright 2006 ACM 1-59593-369-7/06/0008 ...$5.00.

might want to ensure that a result set offers good "coverage" of the different possibilities. The *instance recall* metric [9, 21] measures the number of different subtopics or queriers who are satisfied by a given result set.

One can apply our approach, of ranking to optimize the expected value of the metric, to all of these metrics. For each metric the exact optimization problem is different, and in each case it appears intractable. In a fortunate coincidence, however, our greedy algorithm for the %no metric is also a natural greedy algorithm for *all* of the metrics listed above. We report results for all of these metrics over TREC corpora, and show that our greedy algorithm outperforms the PRP baseline on them. Conversely, our analysis leads us to the possibly surprising conclusion that PRP ranking is *not optimal* for the heavily used *mean average precision* metric.

We also explore the goal of *perfect precision*, where the objective is to not retrieve *any* irrelevant documents. We show how blind *positive* relevance feedback arises naturally as a greedy heuristic for achieving this goal. To tie together the disparate goals of nonzero precision (at least one relevant document) and perfect precision, we introduce *k-call*, a class of metrics that ranges smoothly between the two extremes. We argue that it captures a desire to trade "quality" for "diversity," and discuss the application of our approach to these metrics.

The broad applicability of using probabilistic models to optimize for specific metrics suggests a general principle, which we call the *Expected Metric Principle* (EMP). The EMP states that, in a probabilistic context, one should directly optimize for the *expected value* of the metric of interest. The PRP is a special case of the EMP for the precision and recall metrics.

One possible criticism of the EMP is that it "teaches to the test"—it encourages the algorithm to do well *only* on the evaluation criterion. This has led to much gaming of the SPECFP benchmarks for numerical computation, for example, as companies incorporated special purpose code for solving only the SPECFP instances. But this gaming is a consequence of the evaluation criterion failing to accurately measure what is wanted out of the system. We argue that the metrics that we study are truly the metrics that matter in certain cases, so that algorithms optimized for them are desirable.

1.1 Retrieving for Diversity

Intriguingly, while explicitly aiming only to find one relevant document, we demonstrate the unplanned effect of increasing the *diversity* of documents at the top. This highlights one way in which seeking *one* relevant document is different from seeking *many*. If a query has multiple interpretations (as was the case for "Trojan horse" above), or if there are multiple subtopics, it may be hard to decide which is the proper one. PRP ranking puts all its eggs in one basket—it identifies the most likely interpretation, and finds many results for that one. But an algorithm that needs only one relevant document can do better by retrieving one document for each case, thus satisfying the goal *whichever* interpretation or subtopic is desired.

Recent work [3, 21] has developed heuristics for increasing diversity for this precise purpose, but our approach appears to be the first in which diversity arises automatically as a consequence of the objective function rather than being manually optimized as a proxy for the true objective of interest. As a benefit, there are no new "parameter knobs," beyond those already used in probabilistic document models, that must be tweaked in order to tune our algorithms.

We give anecdotal evidence that our approach promotes diversity by looking at ambiguous queries on the Google search engine. We observe that while the probability ranking principle tends to return documents only relevant to the "majority vote" meaning of the query, our approach satisfies that meaning but simultaneously returns results relevant to other, rarer meanings of the query. We follow with more quantitative evidence based on TREC results with multiple raters, where our approach satisfies more raters than PRP, and TREC results with subtopic annotations, where our approach retrieves more subtopics than PRP.

2. RELATED WORK

Our discussion of related work splits into three categories: definitions of and motivations for retrieval metrics, algorithms for optimizing those metrics, and approaches to diversity in result sets.

2.1 Beyond Precision and Recall

The main metric we examine is essentially the %no metric, which is studied by Voorhees [19]. She finds that the metric was less stable than traditional measures. However, this instability does not affect our ability to probabilistically model and optimize for it.

Cooper [4], who introduces the search length metric, argues that trying to retrieve as many documents as possible is not necessarily the appropriate objective for meeting user information need. Cooper explicitly divides an information request into a "relevance description" (i.e., a query) and a *quantification* that specifies the desired number of relevant results. He defines a class of search length metrics, which measure the number of irrelevant documents a user would have to examine before finding a "sufficient" number of relevant documents. Our paper focuses on the case of "one document sufficiency," though we also touch on the "k document sufficiency" case when we define k-call later in the paper.

Shah and Croft [15] explore the problem of *high accuracy retrieval*, where the objective is to have high precision in the top document ranks. They argue that *mean reciprocal rank* is a useful metric for this scenario. As previously mentioned, we also demonstrate the applicability of our heuristics to MRR.

2.2 Algorithms

Our approach fits within a general *risk minimization* framework propounded by Zhai and Lafferty [22]. They observed that one could define an arbitrary numeric *loss function* over possible returned documents rankings, which measures how unhappy the user is with that set. The loss function generally depends on unavailable knowledge about the relevance of particular documents. But given a probabilistic model, one can compute an *expected value* for the loss function, or *expected loss*, and return a result that optimizes the expected loss. Much of our paper deals with the loss function that is (say) -1 when the top ten results contain a relevant document (indicating a positive satisfaction) and 0 when it does not.

Like us, Gao et al. [6] follow the approach of letting the metric directly drive the retrieval algorithm. However, instead of using a document model from which the optimal algorithm can be determined through analysis, they *train* a system to weight document features so as to optimize the metric (average precision in their case) and show that such training leads to an algorithm that achieves good performance on the metric with new queries.

Bookstein [1] proposes a sequential learning retrieval system that bears some similarity to ours. He argues that a retrieval system should sequentially select documents according to the probability of relevance conditioned on the selection and relevance of previous documents (essentially relevance feedback). However, his procedure requires explicit user feedback after each result retrieved, whereas our system proposes an objective function and then uses a sequential document selection algorithm to heuristically optimize that objective without further user input.

Our greedy algorithm for achieving perfect precision seems related to *pseudo-relevance feedback*, an approach commonly used in the literature to improve overall retrieval performance on standard metrics [2, 5]. Our metric for retrieving at least one relevant document, on the other hand, produces an algorithm that appears to be doing *negative* pseudo-relevance feedback. In either case, rather than feeding back all of the top documents at once, we progressively feed back more and more top relevant documents in selecting latter-ranked documents.

2.3 Diversity

In their subtopic retrieval work, Zhai et al. [21] posit, as we do, that there may be more than one meaningful interpretation of a query. They assume that a query may have different subtopic interpretations, and reorder the results so that the top includes some results from each subtopic. Their system involves separate consideration of *novelty* and *redundancy* in a result set, which are then combined via a cost function. Our approach, in contrast, aims directly at the goal of maximizing the chances that the user will get an answer to "their" interpretation of the query. Aiming directly arguably is beneficial in that it reduces the number of system elements, such as novelty and redundancy, whose interactions we have to design and tweak. Conversely, it is possible that by modeling novelty and redundancy richly, the Zhai et al. model can outperform our simpler one.

The work of Zhai et al. is in turn based on Carbonell and Goldstein [3]'s *maximum marginal relevance* (MMR) ranking function. They argue for the value of diversity or "relevant novelty" in the results of a query, and propose MMR as an objective that introduces such diversity in a ranked result set. Our greedy heuristic for optimizing the "one relevant document" objective simplifies to a computation that bears some relation to the MMR computation. However, MMR is advanced as a heuristic *algorithm* for reducing redundancy and achieving the hard-to-define notion of *diversity*, which in turn is believed to be *related* to the desired objective. Our ranking algorithm arises naturally from the application of a simple greedy heuristic to the optimization of a clear, natural, formally defined objective function. In addition, while the iterative greedy approach is implicit in the definition of MMR, our greedy approach is simply one heuristic applied to optimizing our well-defined objective function; we expect that better optimization algorithms such as local search would yield improved values for our objective, which should translate into improved retrieval performance.

Our goal of retrieving one relevant document, and its inherent diversifying tendency, bears superficial similarity to clustering, in the sense that clustering is also used as an approach to quickly cover a diverse range of query interpretations [10]. Our technique sidesteps the need for clustering interface machinery, utilizing the standard ranked list of documents instead. Furthermore, we aim to directly optimize the probability that a user finds a relevant document, rather than going through the intermediate notion of separate document clusters. Again, this avoids the need to define and tweak new algorithmic parameters.

3. EVALUATION METRICS

We consider several metrics in this paper. *Search length* [4] (cf. section 2) is the rank of the first relevant document in a result list minus one, and *reciprocal rank* [15] is one over the rank of the first relevant result. With multiple queries we can take the mean of both quantities, yielding the *mean search length* (MSL) and *mean reciprocal rank* (MRR) metrics.

Introduced for the TREC robust track [20], the %no metric measures the percentage of queries for which no relevant documents are retrieved. Put another way, it assigns value to any result set containing *at least one* relevant document.

In line with Cooper's [4] notion of quantification, we generalize the %no metric with a new class of binary metrics under the name k*-call at* n. Given a ranked list, k-call at n is one if at least k of the top n documents returned by the retrieval system for the given query are deemed relevant. Otherwise, k-call at rank n is zero. In particular, 1-call is one if a relevant document is found and zero otherwise. Averaging over multiple queries yields *mean* 1*-call*, which is just one minus the %no metric used in the TREC robust track. On the other hand, n-call at n is a measure of *perfect precision*: returning only relevant documents. Varying k between n and 1 offers a way to express "risk tolerance": do we wish to aim for many relevant documents and take the chance of finding none, or will we settle for fewer documents if it improves our chances of finding them?

When we explicitly have a notion of different subtopics of a query, and of different documents covering different subtopics, we can define *instance recall* [9] at rank n (also called *S-recall* [21]) as the number of unique subtopics covered by the first n results, divided by the total number of subtopics.

4. BAYESIAN RETRIEVAL

Our work is rooted in standard Bayesian information retrieval techniques [11, 16]. In this approach, we assume that there are two distinct probability distributions that generate the relevant and irrelevant documents respectively. Let d be a document, and r a binary variable indicating the relevance of that document. The probability ranking principle suggests that documents in a corpus should be ranked by $\Pr[r \mid d]$—that is, the likelihood that a document was generated by the relevant distribution. An application of Bayes' Rule followed by a monotonic transformation gives us a *ranking value* for documents:

$$\frac{\Pr[d \mid r]}{\Pr[d \mid \neg r]}. \tag{1}$$

Here, $\Pr[d \mid r]$ and $\Pr[d \mid \neg r]$ represent respectively the probabilities that the relevant and irrelevant distributions assign to the document.

We thus need to compute $\Pr[d \mid r]$ and $\Pr[d \mid \neg r]$. In our paper we emphasize the objective function, rather than the modeling issues associated with Bayesian retrieval. Therefore we use the familiar and simplistic *Naïve Bayes* framework, with multinomial models as the family of distributions. A document is thus a set of independent draws from a word distribution over the corpus. A multinomial distribution is described by parameters θ_i, one for each term (word) i in the corpus. A document's probability is the product of each of its term's corresponding θ_i, normalized so the distribution sums to one. In our experiments, we used the heuristic of applying a log-transformation to the term frequencies (that is, substituting $\log(1 + t_i)$ for t_i), which has been shown in the literature to improve Naïve Bayes performance for text applications [13].

It remains to determine the parameters θ_i for each distribution. To model the fact that we do not know exactly what terms appear in relevant and irrelevant distributions, we specify a *prior probability distribution* over the parameters (a distribution over possible document distributions). The prior reflects our initial beliefs (e.g. that a given θ_i parameter is likely to be small). We proceed to revise our beliefs about those parameters by incorporating observed data. We use a standard *Dirichlet prior*, centered on the background word distribution over the entire corpus. We then take the user query as "training data"—a sample from the relevant document distribution that gives evidence about the parameters of that distribution. This evidence leads us to a new *posterior* estimate of the probability of parameters of the relevant distribution (in particular, one in which the query term's parameters are likely to be large). Given these two distributions, we are able to measure the probabilities that certain sets of documents are relevant or irrelevant. For our baseline PRP model, this is all the training we do. As we will show later, our new EMP-based algorithms will feed back selected corpus documents into the document distributions as additional "training data."

Our use of priors creates a "linkage" between documents. Although each document is assumed to be generated independently from its (relevant or irrelevant) distribution, positing one document to be relevant leads us to believe that the parameters associated with that document's words are stronger in the relevant distribution, which in turn leads us to believe that similar documents are more likely to be relevant.

5. OBJECTIVE FUNCTION

In many systems, the evaluation metric (e.g., mean reciprocal rank) is different from the objective function used to rank documents (e.g., probability of relevance). The EMP posits that the right objective function is the (expected value of the) evaluation metric itself. Consider optimizing for the k-call at n metric. Since k-call is always 0 or 1, this is equivalent to maximizing the probability that we find k relevant documents among the first n.

Let d_i denote the ith document of the ranked result set, and r_i denote a binary variable indicating that the d_i is relevant. Result numbering is 0-based, so the first result is d_0.

The $k = 1$ version of our objective function is the probability that at least one of the first n relevance variables be true, that is:

$$\Pr\left[r_0 \cup r_1 \cup \cdots \cup r_{n-1} \mid d_0, d_1, \ldots d_{n-1}\right]. \qquad (2)$$

In general, the objective function for arbitrary k is the probability that at least k documents are relevant, that is:

$$\Pr\left[\text{at least } k \text{ of } r_0, \ldots, r_{n-1} \mid d_0, d_1, \ldots d_{n-1}\right]. \qquad (3)$$

Contrast these objectives with the PRP ranking by $\Pr[r \mid d]$—by defining a objective in line with the metric, we are explicitly aiming for results that the metric rewards.

Note that our objectives are indifferent to the ordering of documents within the top n. This is to be expected—because our metric is insensitive to where the relevant results are within the top n, just that there are enough of them, our objective function will be insensitive to the same conditions.

The next two sections discuss the optimization and calculation of the objective function in depth.

6. OPTIMIZATION METHODS

Notably, while the PRP objective could be optimized by selecting each document independently, our new objectives (equations 2 and 3) seem to be more complex, requiring us to consider interactions between multiple documents in the result set. It no longer seems possible to judge each document individually. So more complex optimization algorithms are needed.

One way to perfectly optimize the k-call at rank n objective function for a corpus of m documents would be to evaluate, for each possible returned sets of n documents, the probability that that set has at least k relevant documents. For any specific set of n documents this evaluation is tractable, but the tremendous number of $\binom{m}{n}$ distinct subsets make this approach impractical for most reasonable values of m and n.

In general, finding the optimum subset is NP-hard. Space limitations preclude a full proof, but one can show that by assigning specific weights to the distributions, one can reduce the NP-hard clique problem to optimizing the expected k-call. Since solving our problem would let us solve an NP-hard problem, our problem is NP-hard as well, implying that exactly optimizing our objective is intractable. Therefore we consider a greedy approach that reduces the search space of result sets.

A greedy algorithm is an algorithm that always selects a locally optimal intermediary to a solution. They tend to be simple approaches that work well in a variety of contexts. A greedy algorithm for our problem is to successively select each result of the result set. Consider finding the optimal result set for k-call at rank n. We select the first result by applying the conventional probability ranking principle. Each result thereafter is selected in sequence. For the ith result, we hold results 1 through $i-1$ to their already selected value, and consider all remaining corpus documents as a possibility for document i. We calculate an expected k-call score for the result set including each such document, and pick the highest scoring document as the ith result. If $i < k$, we maximize the i-call score instead as a stepping stone towards maximizing k-call.

Unlike PRP ranking, optimizing our objective function exactly may require knowing both k *and* n. That is, the set of 10 documents optimizing the odds of getting a relevant document in the top 10 need not contain the 5 documents optimizing the odds of getting a relevant document in the top 5. Thus, it might not be possibly to simultaneously optimize these two quantities. Our greedy heuristic, on the other hand, is not affected by the value of n that we choose; thus, while it may not be returning the *best* document subset for any particular n, it may arguably be returning a ranking that is *reasonably good* for all n.

If our goal were to maximize precision and recall, then the natural greedy approach would be to select each successive document to maximize its probability of relevance (independent of the previously selected documents). This is exactly the PRP ranking mechanism. In this sense, the greedy algorithm we have proposed is a generalization of the greedy algorithm that optimizes for PRP.

7. APPLYING THE GREEDY APPROACH

In this section we examine how we would use the greedy algorithm described previously to actually optimize our objective function. We focus on the $k = 1$ and $k = n$ cases, where the greedy algorithm has a particularly simple instantiation. We also touch on the general case for intermediate values of k.

7.1 $k = 1$

Consider the case where $k = 1$. The first result is obtained in the conventional fashion—by choosing the document d_0 maximizing $\Pr[r_0 \mid d_0]$. Having chosen the first document d_0, we want to select the second document d_1 so as to maximize $\Pr[r_0 \cup r_1 \mid d_0, d_1]$. This is merely an instantiation of equation 2 with $n = 2$. We can expand this expression by partitioning the event of interest $r_0 \cup r_1$ into the disjoint events r_0 and $r_1 \cap \neg r_0$:

$$\begin{aligned}
&\Pr[r_0 \cup r_1 \mid d_0, d_1] \\
&\quad = \Pr[r_0 \mid d_0, d_1] + \Pr[r_1 \cap \neg r_0 \mid d_0, d_1] \\
&\quad = \Pr[r_0 \mid d_0, d_1] + \Pr[r_1 \mid d_0, d_1, \neg r_0] \cdot \Pr[\neg r_0 \mid d_0, d_1] \\
&\quad = \Pr[r_0 \mid d_0] + \Pr[r_1 \mid d_0, d_1, \neg r_0] \cdot \Pr[\neg r_0 \mid d_0]
\end{aligned}$$

where the simplification in the last line follows because r_0 is independent of d_1. We wish to choose the document d_1 maximizing this quantity. Only one of the three probabilities in the equation depends on d_1, however, so it is sufficient to maximize that term:

$$\Pr[r_1 \mid \neg r_0, d_0, d_1]$$

A similar analysis shows that we can select the third result by maximizing $\Pr[r_2 \mid \neg r_0, \neg r_1, d_0, d_1, d_2]$. In general, we can select the optimal ith document in the greedy approach by choosing the document d_i that maximizes:

$$\Pr[r_i \mid \neg r_0, \ldots, \neg r_{i-1}, d_0, \ldots, d_i]$$

This expression tells us that for each new result, we should assume that all the past results were irrelevant, and find the document of greatest relevance conditioned on that assumption. This makes intuitive sense—if a previous document were able to satisfy the user query (i.e., was relevant), then we would not care about what documents were displayed subsequently. Thus we try to select the best new document assuming that all previous documents were irrelevant. This formula also fits nicely with the Bayesian information retrieval model: the assumption that the previous documents are irrelevant is incorporated in a straightforward fashion as an update to the probability distribution associated with the irrelevant documents; the relevance probabilities of new documents are then computed using that updated irrelevant document distribution.

7.2 $k = n$

We can perform a similar greedy-algorithm derivation for the case where $k = n$. In that case, we find that we should select the ith document according to:

$$\Pr[r_i \mid r_0, \ldots, r_{i-1}, d_0, \ldots, d_i]$$

Again this is intuitive—if we want to maximize the odds of perfect precision then once we select even one irrelevant document we have failed; thus, we must forge ahead on the assumption that all documents ranked so far are relevant. As in the $k = 1$ case, this leads to a simple update rule for the prior probability distributions.

Because these simplified forms for $k = 1$ and $k = n$ do not involve addition of probabilities, they have the advantage that we can use the ranking value form of $\Pr[r \mid d]$ (equation 1) rather than the full Bayesian expansion, just as in PRP.

7.3 $1 < k < n$

We briefly turn to the more general problem of trying to get an arbitrary k relevant documents among the top n. In this case, our objective be to maximize the probability of having at least k relevant documents in the top n. Using the same technique of breaking the objectives into chained conditional probabilities, we can derive a ranking value formulation for each step of the greedy algorithm. For brevity we omit the actual derivation here, and focus the remainder of the paper on the $k = 1$ and $k = n$ cases.

8. OPTIMIZING FOR OTHER METRICS

We have remarked on two other metrics aimed at "the first relevant document"—search length and reciprocal rank. Using our EMP approach, our goal should be to optimize the *expected values* of these quantities: minimize expected search length and maximize expected reciprocal rank (note that $E[1/X] \neq 1/E[X]$, so these two objective may optimize differently).

Let us consider the expected (over result sets) search length[1] and develop a greedy algorithm for it. Suppose that the first i documents in the ranking $d_0, \ldots, d_{i-1}$ have been chosen and we wish to greedily select d_i. For those events in which there is a relevant document already in the ranking, our choice does not affect the expectation. So, we should choose greedily *conditioned on there being no previous document relevant*. Subject to this condition, the natural document to choose is the one that has the largest probability of relevance subject to this condition, since this greedily maximizes our chance of "terminating the search" at document d_i.

In other words, we should choose the document d_i that has maximum probability of relevance subject to no previous document in the list being relevant—exactly the same heuristic as we used for optimizing 1-call.

A similar argument shows that our greedy algorithm is also a natural heuristic for optimizing expected reciprocal rank. These observations mean that we can experiment with three metrics for the price of one. Our tables report all three metrics, and demonstrate that our greedy algorithm improves on PRP for all of them.

Now consider the instance recall metric, which measures the number of distinct subtopics retrieved. If a query has t subtopics, then instance recall can be written as $(S_1 + S_2 + \cdots + S_t)/t$, where S_j is an indicator variable that is 1 if a document from the j^{th} instance (subtopic) is included in the result set. Our approach calls for maximizing the expectation of this quantity, which (due to linearity of expectation) is proportional to $\sum E[S_j]$. Our algorithm greedily optimizes for each S_j separately (since S_j is simply the 1-call metric for the jth subtopic) and thus for the sum as well.

It is important to note that although we have described a *heuristic* that is effective for four metrics, they are in fact distinct metrics, and it is conceivable that more sophisticated algorithms could lead to divergent results that optimize one at the expense of the others. On the other hand, it is also conceivable that since these metrics are closely related, there is a ranking that (in expectation) optimizes several or all of them simultaneously.

We briefly consider one other metric, *mean average precision*. This is one of the most commonly used metrics for evaluating retrieval systems. As with the standard precision metric, it is natural to assume that the PRP holds as the (easy) way to optimize it. Note, however, that when there is only one relevant document, average precision simplifies to RR. Our discussion above, which argues that one should "hedge" one's retrieval in order to optimize RR, thus indicates that *PRP does not yield the optimum ranking for average precision*.

9. GOOGLE EXAMPLES

In the introduction, we argued that optimizing 1-call would automatically lead a system to select a more "diverse" result set. To explore this, we first present results from running our procedures over the top 1000 results returned by Google for two canonically ambiguous queries, "Trojan horse" and "virus." We used the titles, summaries, and snippets of Google's results to form a corpus of 1000 documents for each query.

The titles of the top 10 Google results, and the PRP and greedy rerankings, are shown in figure 1. (The titles have been shortened for fit.) Our greedy algorithm was set to optimize for 1-call—that is, the probability of returning one relevant document. Different table cell shadings indicate different broad topic interpretation of a result (e.g., white for computer Trojan horses and various grays for other interpretations). In the "Trojan horse" example, the greedy algorithm returns a significantly more diverse set of results in the top 10 (spanning five distinct interpretations) than PRP and the original Google results, which return respectively three and two interpretations. The diversity for "virus" is also notable; greedy returns the most medical (non-computing) virus results in the top ten, beating PRP. Interestingly, Google does not return any medical virus information in its top ten, so a user looking for that interpretation would be disappointed by Google.

[1] Our terminology overloads Cooper's [4] definition of expected search length, which addressed ties in the ranking by randomly ordering tied results.

Figure 1: Example results from Google; different shadings indicate different interpretations

Query "Trojan horse" (T.H.)			Query "virus"		
Google	PRP	Greedy for 1-call	Google	PRP	Greedy for 1-call
T.H. Attacks	T.H. Removal	T.H. Removal	Symantec	McAfee Virus Defs.	McAfee Virus Defs.
T.H. - Webopedia	T.H. Removal	T.H. Detector	Symantec Virus Hoaxes	Anti Virus Directory	Anti Virus Directory
Symantec - T.H.	T.H. Detector	T.H. Inn	Symantec Security Response	Virus Threat Center	Latest virus descriptions
Symantec Glossary	Symantec - T.H.	T.H. Info	Norton AntiVirus	Virus Threat Center	Virus Threat Center
T.H. - Wikipedia	Symantec - T.H.	Achilles in T.H.	Trend Micro	Latest virus descriptions	West Nile Virus
Trojan War history	Achilles in T.H.	T.H. - Wikimedia	McAfee	Vaccinia virus	Vaccinia virus
T.H. Myth	T.H. Inn	Gay Activism as T.H.	Virus Bulletin	Symantec - FatCat Hoax	Summary virus table
What is T.H.?	T.H. Info	T.H. Removal	AVG Anti Virus	Symantec - Londhouse Hoax	alt.comp.virus FAQ
CERT Advisory T.H.s	T.H. - Wikimedia	T.H. game expansion	Vmyths.com	West Nile Virus	Panda Software
T.H. - Whatis.com	T.H. Scandal	Acid T.H.	Sophos Anti-Virus	Symantec - Hairy Palms Hoax	Sophos virus analyses

10. EXPERIMENTS

In this section, we discuss our results with the greedy algorithm on various TREC corpora. We denote the greedy algorithm that optimizes for (expected) 1-call as 1-greedy. We also look at using the greedy approach to optimize for 10-call (perfect precision), which we denote as 10-greedy. All experiments are done with result sets of size ten. Each corpus was filtered for stop words and stemmed with a Porter stemmer. For both PRP and k-greedy, the query was weighted at one fiftieth of the relevant distribution prior (this was the best weighting for PRP, and we kept it unchanged for 1-greedy). In each case, we ran k-greedy over the top 100 results from PRP. (Generally we found that our algorithms would select from within the top 100 PRP results even when given a choice from the entire corpus.)

Because we did not rank the entire corpus in our results (as doing so would be prohibitively slow), we compute search length only over the first ten results. If there are no relevant results in the top ten positions, we assume a search length of ten. Similarly, we assume a reciprocal rank of zero if a relevant result is not found in the top ten. Therefore our reported MSLs and MRRs are slight underestimates of what their values would be over a full ranking.

We used the set of *ad hoc* topics from TREC-1, TREC-2, and TREC-3 to set the weight parameters of our model appropriately. Using the weights we fond, we then ran experiments over the TREC 2004 robust track, TREC-6, 7, 8 interactive tracks, and TREC-4 and TREC-6 *ad hoc* tracks.

10.1 Tuning the Weights

As with any model, our model has a set of tweakable weights that could greatly affect retrieval performance depending on how they are set. For our model, the key weights to consider are the strength of the relevant distribution and irrelevant distribution priors with respect to the strength of the documents that we add to those distributions.

To tune these weights, we used the corpus from the TREC-1, TREC-2, and TREC-3 *ad hoc* task, consisting of about 742,000 documents. There were 150 topics for these TRECs (topics 51 through 200).

We find that when the prior weight is well tuned, 1-greedy outperforms PRP on the metrics where it should—that is, 1-call at 10, MRR, and MSL. Similarly, with a well tuned prior weight, 10-greedy outperforms PRP on 10-call. For brevity, we do not report the full weight tuning results in this paper.

Our tuning shows that 1-greedy performs best when the irrelevant distribution prior is set very low, to less than the weight of one document, whereas 10-greedy performs best when the relevant distribution prior is set at an intermediate value of approximately the weight of 10 documents. The former indicates that feeding back negative documents strongly is important for diverging the results away from PRP, whereas the latter indicates that we may be "drowning out" the query if our feedback documents are weighted too strongly.

Since TRECs 1, 2, and 3 were used for tuning weights, retrieval results on them were not meaningful. Instead, for evaluation we applied 1-greedy and 10-greedy with the prior weight settings we found in this section, to a different corpus and set of topics.

10.2 Robust Track Experiments

We turn our attention to the TREC 2004 robust track. The robust track uses a standard *ad hoc* retrieval framework, but is evaluated with an emphasis on the overall reliability of IR engines—that is, minimizing the number of queries for which the system performs badly. There were 249 topics in total[2], drawn from the *ad hoc* task of TREC-6,7,8 (topics 301 to 450), the 2003 robust track (topics 601-650), and the 2004 robust track (topics 651-700). The corpus consisted of about 528,000 documents. Note that there is no overlap between this corpus and the TREC-1,2,3 corpus, in either documents or topics.

From the 249 robust track topics, 50 were selected by TREC as being "difficult" queries for automatic search systems. We separately call out the results for these 50 topics.

Table 1: Robust Track Results (249 Topics)

All topics					
Method	1-call	10-call	MRR	MSL	P@10
PRP	0.791	0.020	0.563	3.052	0.333
1-greedy	**0.835**	0.004	**0.579**	**2.763**	0.269
10-greedy	0.671	**0.084**	0.517	3.992	**0.337**
50 difficult topics only					
Method	1-call	10-call	MRR	MSL	P@10
PRP	0.580	0.000	0.303	5.500	**0.160**
1-greedy	**0.620**	0.000	**0.333**	**5.000**	0.158
10-greedy	0.420	0.000	0.254	6.500	0.152

Table 1 presents the results for the robust track. We show a noticeable improvement in 1-call by using 1-greedy instead of PRP. When we restrict our attention to just the 50 difficult queries, the results overall are lower, but 1-greedy is still more likely than PRP to return relevant results. Similarly, 10-greedy's 10-call score is higher than the corresponding PRP and 1-greedy scores. The difficult queries live up to their name for 10-call—none of our algorithms satisfy the strict 10-call criterion over that subset.

We also note that 1-greedy actually has worse precision at 10 than PRP. However, as we argued earlier, precision is not the appropriate metric for our task, so a lower precision score is not problematic. Interestingly, 10-greedy does not affect precision noticeably,

[2] One topic was dropped because the evaluators did not deem any documents relevant for it.

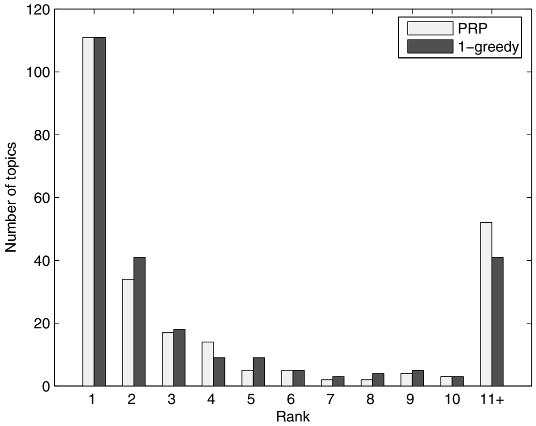

Figure 2: Robust Track Earliest Relevant Ranks

likely because precision rewards result sets that have lots of relevant documents, and 10-greedy is more likely to have *every* result be relevant.

Finally, our performance on the other metrics for which we are greedily optimizing, namely MRR and MSL, is better under 1-greedy than with PRP. Because our 1-greedy procedure attempts to diversify the result set after selecting the first result, we would expect that it would be more likely to find a relevant result for the next few positions than PRP (recall that both methods choose the first result identically). In other words, if the first result was not relevant, 1-greedy will be more likely to select something different, and thus, something relevant, for the second result. On the other hand, PRP will stick with the same interpretation of the query, so if the first document was of the wrong interpretation (and thus irrelevant) the second document would more likely continue that trend. To examine the gains we are making in MRR and MSL, consider figure 2, which graphs the location of the first relevant document for the topics. As the figure demonstrates, it is more often the case that 1-greedy chooses a relevant document for the second position, but the effect disappears for higher ranks, as we would expect.

We conducted statistical significance tests on the robust track experiment's results to compare our greedy algorithms against the PRP baseline. For 1-greedy vs. PRP on 1-call, a one-tailed McNemar test gives $p = 0.026$, which indicates significance at the 5% level. For 10-greedy vs. PRP on 10-call, $p = 0.0002$, which indicates significance at the 1% level. Using a one-tailed Wilcoxon test, we find that for 1-greedy vs. PPR on MRR and MSL, $p = 0.314$, which is not statistically significant. The MRR and MSL gains are interesting but too minor to be significant.

10.3 Instance Retrieval Experiments

As described in section 8, optimizing for 1-call has the side-effect of seeking diversity in the result set—it returns more distinct interpretations of the query in expectation. The TREC-6, 7, and 8 interactive track runs afford us a unique opportunity to test the performance of our system for diversity, because each run's topics were annotated with multiple "instances" (i.e., subtopics) that described its different facets [9]. The document judgments were also annotated with the instances that they covered. In total, there were 20 topics, with between 7 and 56 aspects each, and a corpus of about 210,000 documents.

Table 2: Interactive Track Results (20 Topics)

Method	Instance recall at 10
PRP	0.234
1-greedy	0.315
LM baseline [21]	0.464
Cost-based, ρ = 1.5 [21]	0.429
Cost-based, ρ = 5 [21]	**0.465**

Table 2 lists the instance recall at rank ten results, along with instance recall scores computed on result sets from the subtopic retrieval work, corresponding to configurations presented in table 2 of Zhai et al.'s paper [21]. In their work, they looked at reranking a mixed pool of relevant and irrelevant documents drawn from the top documents selected by a language model baseline. For parity of comparison, we simulated their experiment conditions by reranking the same starting pool of documents as they did.

We note that 1-greedy outperforms our own PRP baseline, as we would expect. However, 1-greedy underperforms Zhai et al.'s system. Zhai et al.'s language model baseline appears to be a much better model for aspect retrieval than Naïve Bayes in the first place. If we had a well-tuned baseline, our 1-greedy would presumably perform better as well. Indeed, Zhai et al.'s reranking systems do not improve upon their baseline on instance recall, though this is probably due to their focus on optimizing the more sophisticated metrics of S-precision and WS-precision, and the (W)S-precision/S-recall curves.

10.4 Multiple Annotator Experiments

Another way of viewing the 1-call goal is from a multi-user perspective. Different users may intend different interpretations, as was evident from the Google examples presented earlier. For TREC-4 and TREC-6, multiple independent annotators were asked to make relevance judgments for the same set of topics, and over the same corpus [18, 7, 17]. In the TREC-4 case, these were topics 202 through 250, over a corpus of about 568,000 documents, and in the TREC-6 case, topics 301 through 350 over a corpus of about 556,000 documents (the TREC-6 topics are a subset of the robust track topics). TREC-4 had three annotators, TREC-6 had two.

Table 3: TREC-4, 6 Multiple Annotator Results

TREC-4 (49 topics)				
Method	1-call (1)	1-call (2)	1-call (3)	1-call (total)
PRP	0.735	0.551	0.653	1.939
1-greedy	**0.776**	**0.633**	**0.714**	**2.122**
TREC-6 (50 topics)				
Method	1-call (1)	1-call (2)	1-call (3)	1-call (total)
PRP	0.660	0.620	N/A	1.280
1-greedy	**0.800**	**0.820**	N/A	**1.620**

The individual 1-call scores for each run and each annotator are presented in table 3. The last column is the sum of the previous columns, and can be considered to be the average number of annotators that are "satisfied" (that is, get at least one result they consider relevant in the top ten) by the respective result sets. Over both corpora, 1-greedy on average satisfied more annotators than PRP.

10.5 Query Analysis

To better understand 1-greedy's improvements, we also looked specifically at instances where 1-greedy returned a relevant result in the top ten (that is, satisfied the 1-call criterion) and PRP did not. The results for topic 100 from the TREC-1,2,3 weight-tuning

Figure 3: Example results from TREC topic 100: Controlling the Transfer of High Technology

Rank	PRP	1-greedy
1	Data transfer software	Data transfer software
2	Disk controllers are getting smarter (SCSI and IDE)	Disk controllers are getting smarter (SCSI and IDE)
3	Caching hard-disk controllers (PC Week buyer's guide)	Environmental Protection Agency tech transfers
4	Environmental Protection Agency tech transfers	Wax vs. dye printers (PC Week buyer's guide)
5	Wax vs. dye printers (PC Week buyer's guide)	Engineering corporation tech transfer
6	Engineering corporation tech transfer	Serial-to-parallel network transfers
7	Department of Energy tech transfers	Whole-Earth technology (international tech transfer)
8	Serial-to-parallel network transfers	Department of Energy tech transfers
9	EISA and MCA technology	Fiber optic telecom line tech transfer through Soviet Union
10	Panel on tech transfer	Simon-Carves PCB tech transfer to Soviet Union

development set is presented in figure 3 (document titles have been summarized for clarity and space), with the relevant result shaded.

The topic description is:

> Document will identify efforts by the non-communist, industrialized states to regulate the transfer of high-tech goods or "dual-use" technologies to undesirable nations.

It is notable that PRP wastes its time on the wrong interpretation of the title—looking for *technologies* that *control* data *transfer*, such as hard drive controllers. While 1-call also pulls up that interpretation, it moves away quickly enough that it can bring back more results on actual tech transfers, including the relevant Soviet Union-related result. [3]

11. CONCLUSIONS AND FUTURE WORK

While the probability ranking principle is appropriate in many settings, it is not the universally "right" approach for optimizing all objective functions. We have identified a common scenario in which the principle is not optimal, and given an approach—the Expected Metric Principle—to directly optimizing other desired objectives. We have shown that this approach is algorithmically feasible, and that it does yield better results for the given metrics.

Much remains to be done to explore heuristics that optimize our new, or other, objective functions. While the greedy approach performs reasonably well, we might expect more sophisticated techniques, such as local search algorithms, to perform better.

We have focused on the "extreme points" $k = 1$ and $k = n$. There is likely to be some value in filling in the middle. For example setting $k = 3$ says that a user wants several relevant documents but does not need them all to be relevant. As in the 1-call case, this would seem to allow the optimization algorithm to hedge—it has room, for example, to include 3 distinct interpretations in the top 10.

Our focus on an objective function means that our approach can theoretically be applied to *any* probabilistic model in which it is possible to discuss the likelihood of relevance of collections of documents. This includes, for example, the two-Poisson model [8], or the language modeling approach [12]. Those better models would hopefully yield better performance.

In general, our work indicates the potential value of "teaching to the test"—choosing, as the objective function to be optimized in the probabilistic model, the metric used to evaluate the information retrieval system. Assuming the metric is an accurate reflection of result quality for the given application, our approach argues that optimizing the metric will guide the system towards desired results. As an example, it may be worth using this approach with the well known average precision metric as the objective function.

12. ACKNOWLEDGMENTS

The authors would like to thank Susan Dumais, Jaime Teevan, and the anonymous reviewers for their comments and feedback, and ChengXiang Zhai, William Cohen, and Ellen Voorhees for their code and data. Harr Chen is supported by a National Defense Science and Engineering Graduate Fellowship (NDSEG).

13. REFERENCES

[1] A. Bookstein. Information retrieval: A sequential learning process. *Journal of the American Society for Information Science (ASIS)*, 34(5):331–342, 1983.

[2] C. Buckley, G. Salton, and J. Allan. Automatic retrieval with locality information using smart. In *Proceedings of TREC-1*, pages 59–72, 1992.

[3] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of ACM SIGIR 1998*, pages 335–336, 1998.

[4] W. S. Cooper. Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems. *American Documentation*, 19(1):30–41, 1968.

[5] E. Efthimiadis. Query expansion. In *Annual Review of Information Systems and Technology*, pages 121–187, 1996.

[6] J. Gao, H. Qi, X. Xia, and J.-Y. Nie. Linear discriminant model for information retrieval. In *Proceedings of ACM SIGIR 2005*, pages 290–297, 2005.

[7] D. K. Harman. Overview of the fourth text retrieval conference (trec-4). In *Proceedings of TREC-4*, 1995.

[8] S. P. Harter. A probabilistic approach to automatic keyword indexing: Part i, on the distribution of specialty words in a technical literature. *Journal of the ASIS*, 26(4):197–206, 1975.

[9] W. R. Hersh and P. Over. Trec-8 interactive track report. In *Proceedings of TREC-8*, 1999.

[10] A. V. Leouski and W. B. Croft. An evaluation of techniques for clustering search results. Technical Report IR-76, University of Massachusetts, Amherst, 1996.

[11] D. D. Lewis. Naïve (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of ECML 1998*, pages 4–15, 1998.

[12] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of ACM SIGIR 1998*, pages 275–281, 1998.

[13] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of ICML 2003*, pages 616–623, 2003.

[14] S. E. Robertson. The probability ranking principle in ir. In *Readings in information retrieval*, pages 281–286. Morgan Kaufmann Publishers Inc., 1997.

[15] C. Shah and W. B. Croft. Evaluating high accuracy retrieval techniques. In *Proceedings of ACM SIGIR 2004*, pages 2–9, 2004.

[16] J. Teevan and D. R. Karger. Empirical development of an exponential probabilistic model for text retrieval. In *Proceedings of ACM SIGIR 2003*, pages 18–25, 2003.

[17] E. M. Voorhees. Overview of the sixth text retrieval conference (trec-6). In *Proceedings of TREC-6*, 1997.

[18] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of ACM SIGIR 1998*, pages 315–323, 1998.

[19] E. M. Voorhees. Measuring ineffectiveness. In *Proceedings of ACM SIGIR 2004*, pages 562–563, 2004.

[20] E. M. Voorhees. Overview of the trec 2004 robust retrieval track. In *Proceedings of TREC 2004*, 2004.

[21] C. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of ACM SIGIR 2003*, pages 10–17, 2003.

[22] C. Zhai and J. Lafferty. A risk minimization framework for information retrieval. In *Proceedings of the ACM SIGIR 2003 Workshop on Mathematical/Formal Methods in IR*, 2003.

[3] Result 10, on the PCB tech transfer to the Soviet Union, could possibly be judged relevant as well, but was not in the document judgments at all, indicating that it was never judged.