

# Predicting Query Performance

Steve Cronen-Townsend  
crotown@cs.umass.edu

Yun Zhou  
yzhou@cs.umass.edu

W. Bruce Croft  
croft@cs.umass.edu

Center for Intelligent Information Retrieval  
Department of Computer Science  
University of Massachusetts, Amherst, MA 01003

## ABSTRACT

We develop a method for predicting query performance by computing the relative entropy between a query language model and the corresponding collection language model. The resulting *clarity score* measures the coherence of the language usage in documents whose models are likely to generate the query. We suggest that clarity scores measure the ambiguity of a query with respect to a collection of documents and show that they correlate positively with average precision in a variety of TREC test sets. Thus, the clarity score may be used to identify ineffective queries, on average, without relevance information. We develop an algorithm for automatically setting the clarity score threshold between predicted poorly-performing queries and acceptable queries and validate it using TREC data. In particular, we compare the automatic thresholds to optimum thresholds and also check how frequently results as good are achieved in sampling experiments that randomly assign queries to the two classes.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Query formulation*

## General Terms

Algorithms, Experimentation, Theory

## Keywords

ambiguity, clarity, information theory, language models

## 1. INTRODUCTION

Dealing effectively with poorly-performing queries is a crucial issue in information retrieval systems. Even what a user believes to be well-formulated queries may, in fact, perform poorly depending on the nature of the collection. Since no user knows the full nuances of the documents in

the collection they are searching, all queries are poorly formulated, to varying degrees.

For example, suppose a user interested in the competitors in the 1988-1989 soccer World Cup issues the query “World Cup” against the TREC AP88 collection of news articles. If those two words are the only evidence the system has about what the user means, it is simply impossible for the system to return the soccer articles consistently higher in the ranked list than the articles about World Cup chess tournaments. Articles about World Cup chess tournaments are predominant, in fact, in the chosen collection among the articles that use the query terms frequently. Despite the fact that the user might not have known that there was a World Cup in anything other than soccer, he or she would get a ranked list with chess articles predominating and with soccer articles interspersed sporadically throughout.

As highlighted in the above example, the degree of ambiguity of a query with respect to the collection of documents being searched is often closely related to query performance. Thus we seek to measure the degree of ambiguity of a query with respect to a collection of documents. Specifically, we measure the degree of dissimilarity between the language usage associated with the query and the generic language of the collection as a whole.

A query whose highly ranked documents are about a single topic (high coherence) has a model characterized by unusually large probabilities for a small number of topical terms. On the other hand, a query returning a mix of articles about different topics (low coherence) has a model that is smoother and more like the model of the collection as a whole. Hence the high-coherence query would get a high score (since its associated language is very different from the overall collection language), a while the low-coherence query would get a low score. Thus, this measure is closely related to the lack of ambiguity, and we call it the *clarity score*.

There is a strong correlation between the clarity score of a test query with respect to the appropriate test collection and the performance of that query. We believe this is due to the fact that a low-coherence retrieval is likely to contain many irrelevant documents in the top ranks and a high-coherence retrieval often contains many relevant documents in the top ranks. Hence it is possible to predict, to some degree, the performance of a query without relevance information. We show how to set the clarity score threshold optimally to predict whether a test query falls in to the upper or lower half of a set of test queries. Finally, we suggest a way to set the threshold without relevance information, making these techniques applicable to real systems.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR '02, August 11-15, 2002, Tampere, Finland  
Copyright 2002 ACM 0-58113-561-0/02/0008 ...\$5.00.

## 2. COMPUTING CLARITY

Throughout this work we use a language modeling approach, similar to the approach first used in information retrieval by Ponte and Croft[16]. Language models, long used in speech recognition, capture statistical aspects of the generation of language[9]. In information retrieval they are often applied in a simple manner, modeling term occurrences at the document level with little or no regard to sequential effects. In the computations of this paper, a “language model” refers to a probability distribution over all single terms (morphologically-normalized words) and may be estimated based on a single document, or a query and collection of documents. Thinking about language models more generically, however, is often fruitful since simple language models can often be replaced by more sophisticated models in future work.

### 2.1 Definition

The first step in computing a clarity score is estimating a query language model. We have investigated both of the methods put forward by Lavrenko and Croft[13] for estimating such models<sup>1</sup>. Here we use Lavrenko and Croft’s *Method 1*. In this approach one assumes, in effect, that the query terms and the terms in the documents are sampled identically and independently from the query model unigram distribution. This results in high probability estimates for terms that occur frequently in documents containing many query terms.

The query language model (unigram distribution over terms) is given by

$$P(w|Q) = \sum_{D \in R} P(w|D)P(D|Q), \quad (1)$$

where  $w$  is any term,  $Q$  the query,  $D$  is a document or the model estimated from the corresponding single document, and  $R$  is the set of documents that contain at least one query term.

As weights  $P(D|Q)$  in Equation (1) we estimate the likelihood of an individual document model generating the query[19] as

$$P(Q|D) = \prod_{q \in Q} P(q|D), \quad (2)$$

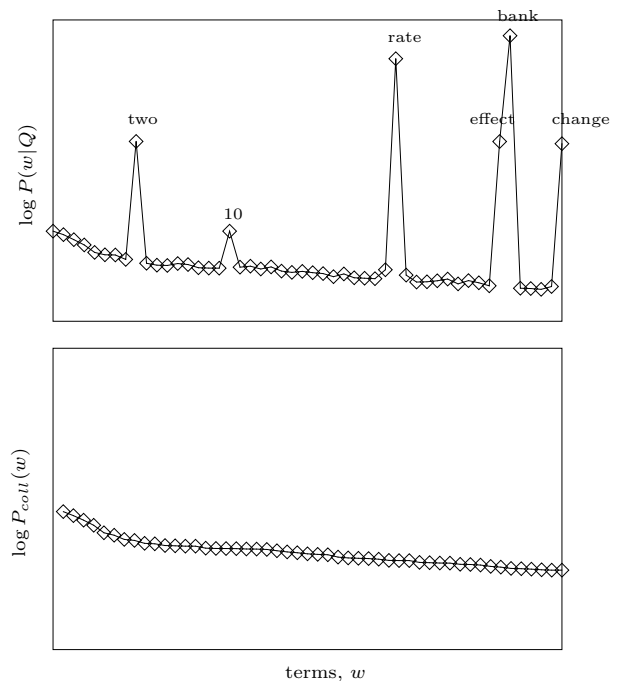
and obtain  $P(D|Q)$  by Bayesian inversion with uniform prior probabilities for documents in  $R$  and a zero prior for documents that contain no query terms.

We estimate  $P(w|D)$  and  $P(q|D)$  in (1) and (2) by relative frequencies of terms linearly smoothed[14] with collection frequencies as

$$P(w|D) = \lambda P_{ml}(w|D) + (1 - \lambda)P_{coll}(w), \quad (3)$$

where  $P_{ml}(w|D)$  is the relative frequency of term  $w$  in documents  $D$ ,  $P_{coll}(w)$  is the relative frequency of the term in the collection as a whole, and  $\lambda = 0.6$  throughout this study. Figure 1 shows example query and collection language models.

The clarity score for the query is simply the relative entropy, or Kullback-Leibler divergence[4], between the query and collection language models (unigram distributions),



**Figure 1: The query language model for query  $A$ , “Show me any predictions for changes in the prime lending rate and any changes made in the prime lending rates” in TREC disk 1 and the collection language model for that collection. The top 50 terms are plotted in order by their collection probabilities.**

given by

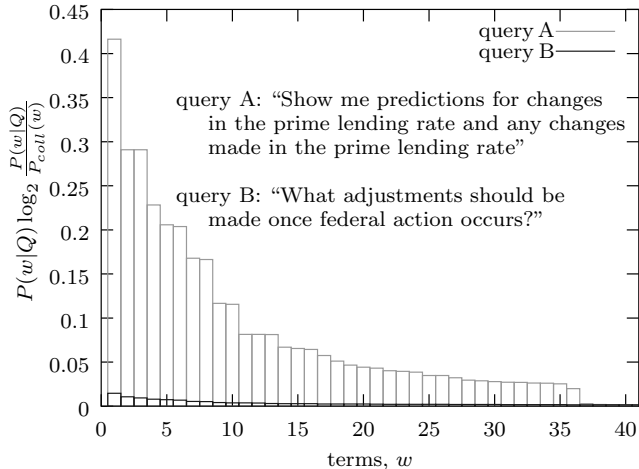
$$\text{clarity score} = \sum_{w \in V} P(w|Q) \log_2 \frac{P(w|Q)}{P_{coll}(w)}, \quad (4)$$

where  $V$  is the entire vocabulary of the collection.

The efficiency of clarity score computation with Equation (4) is dominated by the estimation of the query language model by Equation (1), since the collection model can be precomputed at index-time. The query model is estimated most efficiently by sampling documents using standard techniques[10]. Throughout this paper we estimate query models by sampling until reaching a limit of 500 unique documents. Alternatively to (4), one can compute the divergence with the roles of the query language model and the collection language model reversed. The method shown in Equation 4 consistently performs slightly better in TREC evaluations of clarity score correlation with average precision (see section 4) and is used throughout this paper.

One can use Lavrenko and Croft’s *Method 2* models as an alternative to Equation (1). The weaker independence assumptions employed in this method result in high probability estimates for terms that commonly co-occur in documents with individual query terms, but not necessarily many query terms at once. Except in AP88+89 tests, the *Method 1* models used in this section result in slightly better correlation between clarity score and average precision (see section 4) and are used unless otherwise noted.

<sup>1</sup>Lavrenko and Croft refer to these as *relevance models*.



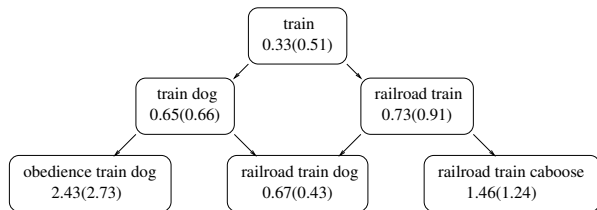
**Figure 2: A clear versus a vague query in TREC disk 1 with the top 40 contributions plotted term-by-term. The top six contributions, in order, for query A are for the terms “bank”, “hong”, “kong”, “rate”, “lend” and “prime” while the top six for query B are “adjust”, “federal”, “action”, “land”, “occur” and “hyundai”**

## 2.2 Examples

Consider two queries, *A* and *B*. *A* is “Show me any predictions for changes in the prime lending rate and any changes made in the prime lending rates” and *B* is “What adjustments should be made once federal action occurs?”

Figure 1 shows the language models used in computing the clarity score for query *A*, which is 2.85 bits. In terms of these figures, the clarity score is the top graph minus the bottom graph, times 2 to the power of the top graph and summed over terms. The difference between the two graphs makes a function with spikes at the most topical words and the final step of taking the expectation value of this quantity using the top distribution enhances the peaks further still. This procedure can also be interpreted as the number of bits wasted on average when one encodes term-occurrence events sampled from the query language model with a code optimally designed for the collection language model.

The best way to visualize the difference between clear and vague queries is to plot the clarity contributions on a term-by-term basis as in Figure 2. Here  $P(w|Q) * \log(\frac{P(w|Q)}{P_{coll}(w)})$  is plotted for each of the top 40 contributing terms *w*, sorted in descending order of contribution. For example, the leftmost gray bar in Figure 2 shows that the term “bank” made the biggest contribution (0.416) to the clarity score of the query *A*. This term is prominent in the query model (Equation (1)) because it occurs frequently in documents whose models are likely to generate the query. Note that “bank” does not, itself, occur in the query. This term makes a high contribution to the clarity score (Equation (4)) because its estimated probability is much higher in the query model than in the collection model. Visually, compare its value in the top graph of Figure 1 (the biggest spike) to the corresponding value directly below it on the collection model graph. Similarly, the small black bar in the leftmost position indicates that term “adjust” made the highest contribution



**Figure 3: Clarity scores for some related queries against the TREC-7 collection. The scores are computed using Equation (4) and *Method 1* query models (Equation (1)), with scores using *Method 2* models in parenthesis. Arrows indicate the addition of a query term.**

to the clarity score of query *B* at 0.015. In this representation, the clarity score for a query is the total of all the heights of the corresponding bars, or the total area of the bars. The clarity score of query *B* is the total area of all the black bars (the tallest 40 of which are shown in Figure 2) and is 0.37 as compared to the total of 2.85 for query *A* (grey bars). This representation makes the contrast evident between high and low clarity score queries.

## 3. CLARITY AND AMBIGUITY

Clarity scores were invented in an attempt to quantify the ambiguity of a query. Looking at the entropy of the query language model is a natural approach since entropy measures how strongly a distribution specifies certain values, in this case terms. Refining to the *relative* entropy of the model as compared to the collection model, allows the method to ignore large and fluctuating contributions due to generic terms.

The clarity scores for a series of related queries is shown in Figure 3. The left and right edges of this figure represent two distinct meanings starting with the term “train”. The left side refers to training of dogs and the right side refers to railroad cars. The lowest central query “railroad train dog” is an ambiguous combination of the two meanings. With *Method 1* models this ambiguous query is far lower scoring than its more specific neighbors to the left and right but comparable to the two two-term queries up one level. Interestingly, though, with *Method 2* models (scores in parentheses in Figure 3) the score of this query is not only lower than the more specific three term queries, but is even lower than the score of the original one-term query “train.” Clarity scores from *Method 2* models seems to “punish” more harshly the addition of a term that does not co-occur with the other query terms in documents in the collection. Clarity scores computed this way correlate less strongly with average precision in evaluations of the type we describe next. More detailed discussion of clarity scores as a measure of ambiguity appear in [6].

## 4. CLARITY AND AVERAGE PRECISION

### 4.1 TREC Ad Hoc Track Evaluations

To lay the groundwork for using clarity scores to predict the performance of a query, we measure the correlation between clarity scores and average precision scores for various TREC Ad Hoc Track test collections and queries. The re-

Collection	Queries	Num.	R	P-value
AP88+89	101 – 200	100	0.368	$1.2 \times 10^{-4}$
TREC-4	201 – 250	50	0.490	$3.0 \times 10^{-4}$
TREC-5	251 – 300	50	0.459	$6.5 \times 10^{-4}$
TREC-7	351 – 400	50	0.577	$2.7 \times 10^{-5}$
TREC-8	401 – 450	50	0.494	$2.7 \times 10^{-4}$
TREC-7+8	351 – 450	100	0.536	$4.8 \times 10^{-8}$

**Table 1: The correlation of clarity scores with average precision in several TREC test collections. The queries are the titles of the TREC topics (usually a few words), except for TREC-4 where the description field is used due to a lack of titles, resulting in queries of 16.1 words, on average.**

trieval is done with a simple multinomial language modeling approach[19]<sup>2</sup>.

Since the score distributions are unknown, an appropriate test is the Spearman rank correlation test[8]. As a first step, each list of scores is replaced by the corresponding ranks (e.g. the lowest score is replaced with 1, the second lowest score with 2, et cetera, and the highest score is replaced with the number of queries). Then one computes what appears to be a correlation coefficient from elementary statistics between the two rankings. A score of 1 indicates perfect agreement in the rankings and a score of  $-1$  indicates opposite rankings. This is a distribution-free statistic whose null distribution (the distribution if the two rankings are unassociated) is well-approximated by a normal for sample sizes as large as 50, our smallest sample size. Thus it is straightforward to estimate the p-values, or probabilities that results as extreme or more extreme occur by chance.

The results (Table 1) show a strong positive association between the clarity score of a query and the average precision of that query. The first row shows, for example, that with AP88 and AP89 TREC test collections combined and using the titles of the TREC topics numbered 101 – 200 as queries (of which there 100), there is a rank correlation of 0.368 between clarity score and average precision. The corresponding p-value of  $1.2 \times 10^{-4}$  indicates the estimated probability that an apparent correlation as extreme, or more extreme, would occur by chance if clarity scores and average precision scores were actually unassociated.

Query length does not seem to effect the correlation results much since the rank correlation coefficient for TREC-4, where the average query length is 16.1, is similar to the other entries farther down the table, where the queries are a few words. The fact that the correlation persists at about the same level when TREC-7 and TREC-8 queries are combined into one set<sup>3</sup> results in a drastically lowered p-value, since such a correlation is much less likely to occur by chance with twice as many queries.

## 4.2 TREC Query Track Evaluations

We also show correlation results for the TREC-9 Query Track (see Table 2). In this evaluation, we check the correlation with precision among different queries for the same TREC topic. The queries, submitted by the 6 research groups that participated in the Query Track, include a vari-

<sup>2</sup>Standard TF.IDF retrieval gives similar correlation results.

<sup>3</sup>This is legitimate since the collections are the same.

Queries	Num.	R	P-value
aggregate	1804	0.39	$2.2 \times 10^{-61}$
by topic	$36.1 \pm 3.7$	$0.247 \pm 0.244$	$2.0 \times 10^{-21}$

**Table 2: The correlation of clarity scores with average precision in the TREC Query track. “Aggregate” indicates all queries taken together while “topic ave.” values are the averages over each of the 50 query track topics.**

ety of very short and one or two sentence queries[2]. There are 43 versions of queries for each of the 50 topics numbered 51 – 100 for TREC disk 1, giving us 50 different values for the rank correlation coefficient. This evaluation is the most relevant our measure’s ability to distinguish between user queries that are likely to work well and ones that are likely to perform poorly for the same information need.

Once queries are stopped and stemmed[11] there are quite a few duplicates, resulting in 36.1 queries per topic, on average, and 1804 queries for the 50 topics in total. The Spearman rank correlation coefficients and associated p-values are computed as explained previously. The p-value listed in the “by topic” results is the probability estimate for a result more extreme than the particular set of 50 rank correlation coefficients seen in that experiment.

It is interesting to note that, for a few individual topics, the rank correlation coefficient is actually negative. This is reflected in the entry showing  $R = 0.247 \pm 0.244$  in Table 2. Examination of several of these poorly-correlating topics show that such topics have some high clarity queries that have low average precision, as well as some low clarity queries with high average precision.

For example, in topic 57, about the long term financial health of the company MCI, the queries “Future of MCI” and “MCI’s profit” are low ranking in clarity but happen to have high ranking average precision (essentially, they place relevant documents near the top of the ranked list by luck) while there are some queries, like “Multiport Communications Interface” that are high ranking in clarity score but low ranking in average precision (this query actually receives a perfect zero in average precision). The query “Multiport Communications Interface” is in fact fairly specific and *should* get a reasonably high clarity score in the TREC disk 1 collection but clearly will not retrieve documents about the telecommunication company MCI. This query seems to have been generated by automatic (and mistaken) expansion of the acronym MCI. In this case and others like it, the failure of the clarity score ranking to predict the average precision ranking is clearly not a fault of our method; the query is essentially a high-clarity score query for a different information need. Since clarity scores are not computed with any reference to the underlying information need, clarity scores can not predict the poor performance of such off-topic queries or, indeed, any queries that specify a coherent set of documents that just happen to be about the wrong topic.

We believe that the slightly lower value of  $R = 0.247$  (average) on a per-topic basis is the result of the presence of extremely low quality queries, such as the one mentioned in the previous paragraph. However, the persistence of the correlation over so many queries still results in an extremely

Collection	Queries	Num.	R	P-value
AP88+89	101 – 200	100	0.409	$2.4 \times 10^{-5}$
TREC-4	201 – 250	50	0.298	0.019
TREC-5	251 – 300	50	0.289	0.022
TREC-7	351 – 400	50	0.467	$5.4 \times 10^{-4}$
TREC-8	401 – 450	50	0.474	$4.5 \times 10^{-4}$
TREC-7+8	351 – 450	100	0.449	$4.0 \times 10^{-6}$

**Table 3: The correlation of the average IDF of query terms with average precision in several TREC test collections. The queries are the titles of the TREC topics (usually a few words), except for TREC-4 where the description field is used, resulting in queries of 16.1 words, on average.**

low p-value since such an apparent correlation is extremely unlikely to occur by chance over such a large data set.

## 5. ALTERNATIVE PREDICTORS

In order to better assess the significance of the clarity measure, we compare it to various other predictors of query performance. In particular, we evaluate the average and total term weights of query terms as predictors of performance using the methods of Kwok[12] and Wong[21], as well as the standard inverse document frequency measure given by

$$IDF(w) = \log_{10} \frac{\text{number of docs}}{\text{number of docs containing } w}. \quad (5)$$

The results for the average *IDF* of query terms are given in Table 3. Average Kwok score of query terms is less correlated with performance than average *IDF* and average Wong weight performs similarly to *IDF*. Using the sum of weights performs worse with all methods.

The results in Table 3 seem to show the average *IDF* of query terms as predicting the performance of queries to some degree, though not as well as clarity scores, in general (cf. Table 1). On TREC-7 plus 8, for example, the results with average *IDF* are about 83 times as likely to occur by chance as the correlations with clarity scores.

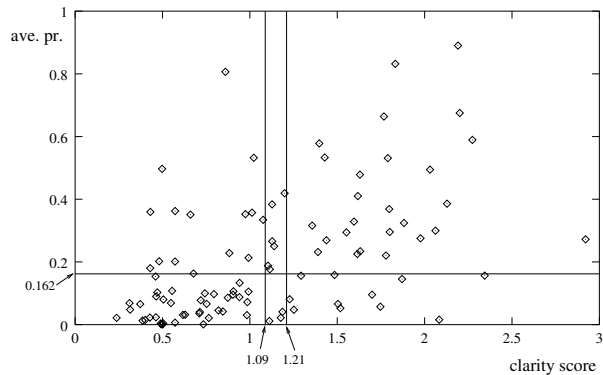
It is important, however, to look at the correlations of the term-weight-based predictors on the query track, the largest and most diverse test query set currently available. Table 4 shows the best performance, which is for the average *IDF* predictor.

All three term-weight-based methods show near random performance on the aggregate data. When looking at the correlations on a topic-by-topic basis the average *IDF* measure does much better than it does on the aggregate. The clarity results (Table 2, second row) are still about a million times less likely to occur by chance, however. The disparity between the aggregate and by-topic performance of average *IDF* of query terms as a performance predictor seems to indicate that it is particularly poor at comparing queries across topics.

In addition to the term-weight based predictors we consider the negative of the entropy of  $P(D|Q)$  as a predictor of query performance. For this predictor we find no appreciable rank correlation with average precision, except at extremely high and low values of the entropy. This correlation makes sense, for example, since the highest values of this entropy indicate the most uniform distributions  $P(D|Q)$  which

Queries	Num.	R	P-value
aggregate	1804	0.025	0.14
by topic	$36.1 \pm 3.7$	$0.220 \pm 0.224$	$2.0 \times 10^{-15}$

**Table 4: The correlation of the average IDF of query terms with average precision in the TREC Query track. “Aggregate” indicates all queries taken together while “topic ave.” values are the averages over each of the 50 query track topics.**



**Figure 4: Average precision versus clarity score for the 100 title queries from the TREC-7 and TREC-8 adhoc tracks. The 0.162 threshold in average precision divides the estimated probability density in half (see Figure 5). The threshold of 1.09 in clarity score is the Bayes optimal level for classifying queries as high or low precision based on their clarity scores, based on the estimated probability densities(see Figure 6). The threshold of 1.21 is the automatic threshold set without relevance information at level where 80% of one-term queries have lower clarity scores (see Figure 7).**

never leads to a high average precision since documents are valued nearly evenly in such a case. Moreover, the overall lack of correlation in this case shows that our inclusion in clarity score computation of language statistics from the documents beyond just their likelihood of generating query terms is necessary for good prediction performance.

## 6. THRESHOLDING

We plan to use clarity scores to make a binary decision about each user query, namely, whether should it be singled out for special treatment on the basis of predicted poor performance, or not. We frame this task, in test collections, as classifying whether the query is likely to score better than a certain average precision threshold, or worse. We show how to set the optimal threshold in order to use clarity scores to make this classification. For the general case where no relevance information is available, we develop a heuristic for setting a clarity score threshold that is reasonable and performs nearly as well as the optimal method in various test collections.

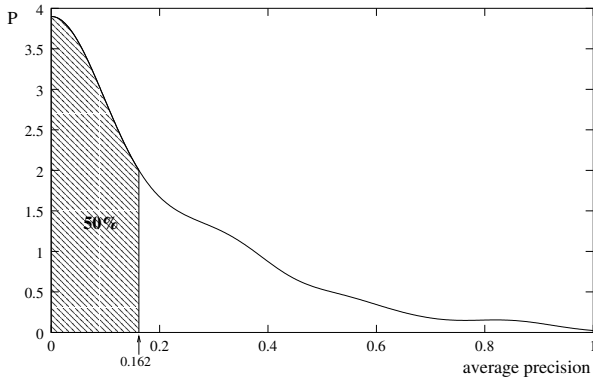


Figure 5: The kernel-estimated probability density for average precision based on the 100 title queries from the TREC-7 and TREC-8 adhoc tracks. The shaded area (average precision less than 0.162) is one half of the total area. So, based on the kernel estimated density for these queries, there is a 50% chance that a query is below this level and a 50% chance it is above this level in performance.

## 6.1 Optimal Thresholding

When relevance information is available for the queries, we set the threshold for clarity scores optimally using standard Bayes decision theory[7]. The feasibility of this approach for the combined Ad Hoc Tracks of TREC-7 and TREC-8 is evident in Figure 4, where the strong positive association between clarity score and average precision can be seen.

We seek to divide the test queries into two classes, “good” (above a certain average precision threshold) and “bad” (below the same threshold). We set this threshold by performing kernel density estimation with automatic setting of the degree of smoothing[1]. Kernel density estimation is a smoothing technique that allows us to estimate the underlying probability density by summing gaussians centered at each observed data point. We require half of the estimated probability density to be below the threshold, as shown in Figure 5. That is to say, we pick the average precision threshold so the estimated probability of a good test query is 50%, and the probability of a bad test query is 50%. This is the horizontal line in Figure 4.

With the high and low average precision classes thus defined, we estimate the probability density functions for the clarity scores of queries as shown in Figure 6, again using kernel density estimation. Since the probability that a test query is in either class is 50%, the Bayes optimal decision boundary is simply where the two class-conditional distributions intersect. Put another way, we predict a query to be good if the estimated likelihood of its clarity score would be greater if it were known to be good than if it were known to be bad. This setting of the threshold is optimal in that it minimizes error rate, where the error of predicting a good query to be bad and the error of predicting a bad query to be good are scored equally. Optimal threshold settings for several collections are shown in Table 5, along with the automatically-set thresholds that we discuss next.

## 6.2 General Thresholding

In the general case there is no relevance information available for the queries. Our approach is to use the scale of pos-

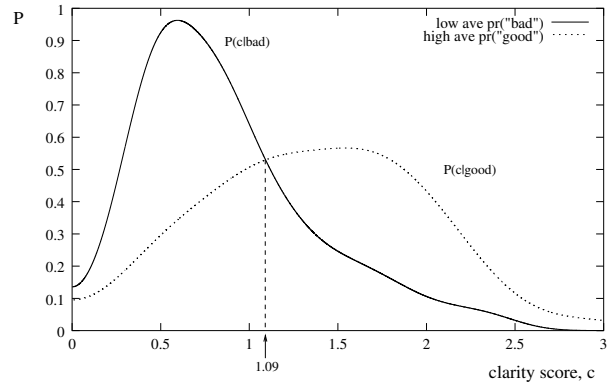


Figure 6: The class-conditional probability densities for the clarity scores of “bad” queries (below the 50% average precision threshold of 0.162) and “good” queries (above the same threshold) in the TREC-7 and TREC-8 adhoc tracks.  $P(\text{bad}) = P(\text{good}) = 0.5$ . The minimum Bayes error is obtained using the indicated decision boundary of 1.09 as a threshold to predict the whether queries fall in the high or low average precision classes.

sible clarity scores for the collection at hand to heuristically set the clarity score threshold.

To obtain information about the possible clarity scores for a given collection we sample single terms randomly from the vocabulary and evaluate their clarity scores as single term queries. We only consider terms that appear in at least 100 documents to avoid estimation problems in the query language models (we have observed that models estimated from too few documents result in clarity scores that are too high).

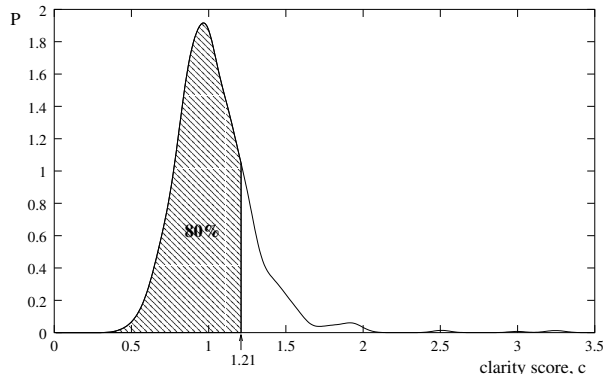
After estimating the probability density over single term queries with kernel density estimation, we set the threshold heuristically so that 80% of the probability density is below threshold (Figure 7). Simply put, a query is deemed “clear enough” if an estimated 80% or more of single term queries would have a lower clarity score. Since the automatic threshold can be computed when the collection is indexed, the decision takes no additional time once the clarity score of the query is computed.

Table 5 shows the results for several test collections. The bottom row, for example, represents the TREC-5 test collection and queries that are the titles of the topics numbered 251-300 (of which there are 50). Here the optimal threshold is 1.27 with a crudely-estimated p-value of  $5.2 \times 10^{-3}$  while the automatically set threshold (80%-rule) is 0.99 with a crudely-estimated p-value of  $2.6 \times 10^{-3}$ . The crudely estimated p-values are the relative frequencies of results as good or better in sampling experiments of  $10^8$  samples. In these sampling experiments queries are randomly assigned to the two classes with probabilities estimated from the observed number of test queries in the two classes.

The automatically set threshold performs nearly as well in the test collections as the optimal threshold (see Table 5). The clarity score thresholds for the TREC-4 and TREC-5 data, which were computed after the 80% rule was developed, do not agree as well as others, but the similarly low likelihood of classification results as good by chance in-

Collection	Queries	Number	Optimal	Automatic
AP88+89	101 – 200	100	0.84( $6 \times 10^{-4}$ )	0.68( $6.2 \times 10^{-5}$ )
TREC-7+8	351 – 450	100	1.09( $2 \times 10^{-5}$ )	1.21( $1.2 \times 10^{-5}$ )
Query(Aggregate)	51 – 100 <i>var.</i>	1804	0.96(0)	1.07(0)
TREC-4	201-250	50	1.49( $2.1 \times 10^{-3}$ )	0.95( $5.4 \times 10^{-3}$ )
TREC-5	251-300	50	1.27( $5.2 \times 10^{-3}$ )	0.99( $2.6 \times 10^{-3}$ )

**Table 5: Bayes Optimal and automatic clarity thresholds for a few test collections. The values in parentheses are relative frequencies of classification results as good or better in  $10^8$  samples where the test queries are randomly assigned to the two classes. The assignment probabilities are estimated to be the observed relative frequencies of queries in the two classes in the set of test queries.**



**Figure 7: The kernel-estimated probability density for the clarity scores of queries whose single terms occur in at least 100 documents in the TREC-7 and TREC-8 adhoc collection. The shaded area is 80% of the total area under the curve, indicating an estimated 80% chance that a one-term query has a clarity score less than 1.21).**

indicate that the automatic thresholds are still reasonable. We believe this variation is caused by the quality of the test queries, since queries of very low average precision (and average clarity scores) tend to push the optimal threshold higher, obtaining more correct classifications of these low-quality queries.

## 7. RELATED WORK

Prediction of query performance has long been of interest in information retrieval, though previous attempts have met with little success. This state of affairs is implicit in the emphasis on combination in information retrieval[5], for example, to combine the results from various versions of a query for the same information need. Without the ability to predict the performance of such multiple versions of a query, all one can do is combine their results equally. Several studies closely related to ours have been published recently, however.

In work on automatic query expansion, Carpineto, de Mori, Romano, and Bigi[3] use a weight very similar to the individual term contributions to the clarity score of a query (as shown in Figure 2, for example) to rank and weight terms within Rocchio query expansion. Also focussed on automatic query improvement, Pirkola and Jarvelin[15] examine individual term contributions to the retrieval effectiveness

of queries and have some success at identifying the most important query term when there is no information as to the actual relevance of the documents to the query. In seeking to classify questions as easy or hard, Sullivan[20] models very long question text directly and compares questions in a sophisticated way to an existing set of questions whose effectiveness at retrieving relevant documents (when viewed as information retrieval-style queries) has been measured. The later two studies rely on relevance information to various degrees, though Pirkola and Jarvelin’s work seeks to be free of this reliance.

Additionally, speculation has been made about using the dispersion of the top documents as a measure of query difficulty[18], and a nearly identical mathematical framework has been used to model selectional preferences in natural language[17].

## 8. FUTURE WORK

We plan to incorporate our methodology into a full information retrieval system. The planned system will compute the clarity score of a user query, before showing the user any results, and attempt to predict whether the query is going to be high- or low- performing. For a predicted high-performing query a normal ranked list will be presented to the user, but for a predicted low-performing query the retrieval results will be clustered by word usage and the user given a choice of seeing one of the cluster’s members, or adding words to the query in an attempt to improve it. To return to the initial example, if the user issued the query “World Cup” against the AP88 collection, we would like the system to return representations of a clusters of documents related World Cup Soccer, World Cup Chess, perhaps several other topics, and a miscellaneous category. The user would be asked to choose a cluster that seems to correspond to their actual information need, or to add words to the query in an attempt to improve it.

As part of this system we have begun to investigate a novel language-model-based clustering method. The documents associated with a query (top-ranked documents, say) are modeled individually and each document defines a point in the space of all possible language models (i.e. the space of all unigram probability distributions, in our case). Kernel density estimation techniques are then applied to estimate the continuous probability density from which the documents are sampled. Alternatively, one can think of this as a smoothing technique. The coordinates of peaks in the estimated density represent cluster language models, which we loosely think of as topic language models, and the nearest peak to a given document point is that document’s

cluster. The degree of smoothing determines the granularity and the number of clusters found. With a high degree of smoothing, there is one cluster corresponding to the collection language model and lower amounts of smoothing result in greater numbers of clusters, with possible numbers of clusters determined by the data itself. To help deal with the high dimensionality of the problem in a sensible way, we plan to restrict the dimensions considered to be the probabilities of top-contributing terms to the query language model (i.e. For query  $B$ , the terms with the tallest bars in Figure 2).

We also plan to explore the use of clarity scores of various possible translations of a query to improve effectiveness in cross-lingual information retrieval and see possible applications of clarity score contributions (Figure 2) in selecting documents for retrieval and other core information retrieval technologies.

## 9. CONCLUSIONS

We have established that the query clarity score, as defined, correlates well with average precision in test collections, even for multiple versions of queries for the same information need. This indicates the possibility of predicting query performance using this measure. We have further grounded these results by comparing the clarity score correlations with the weaker correlations between the average  $IDF$  of query terms and performance. To facilitate applications, we have proposed a simple method of setting the threshold in clarity scores that does not require relevance information. We have validated this method by comparison with minimum Bayes error rate thresholds in a variety of test collections, in conjunction with sampling experiments that randomly classify documents. We believe that these strong results will open up interesting research pathways in information retrieval.

## 10. ACKNOWLEDGEMENTS

We thank Victor Lavrenko for advice on estimation of query models and probability densities. This work was supported in part by the Center for Intelligent Information Retrieval, in part by the National Science Foundation under grant number IIS-9907018, and in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912, and in part by Advanced Research and Development Activity under contract number MDA904-01-C-0984. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

## 11. REFERENCES

- [1] A. Bowman and A. Azzilini. *Applied Smoothing Techniques for Data Analysis*. Oxford University Press, New York, 1997.
- [2] C. Buckley. The trec-9 query track. In E. Voorhees and D. Harman, editors, *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, 2000. NIST Special Publication 500-249.
- [3] C. Carpineto, R. de Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001.
- [4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, 1991.
- [5] W. B. Croft. Combining approaches in information retrieval. In W. B. Croft, editor, *Advances in Information Retrieval: Recent Research from the CIIR*, pages 1–36. Kluwer Academic Publishers, Boston, 2000.
- [6] S. Cronen-Townsend and W. B. Croft. Quantifying query ambiguity. In *Proc. of Human Language Technology 2002*, pages 94–98, March 2002.
- [7] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.
- [8] J. D. Gibbons and S. Chakraborty. *Nonparametric Statistical Inference, 3rd ed.* Marcel Dekker, New York, New York, 1992.
- [9] F. Jelinek. *Statistical Models for Speech Recognition*. MIT Press, Cambridge, 1997.
- [10] M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods, Volume I: Basics*. Wiley-Interscience, New York, 1986.
- [11] R. Krovetz. Viewing morphology as an inference process. In *Proc. of the 16th Annual ACM SIGIR Conference*, pages 191–202, June–July 1993.
- [12] K. L. Kwok. A new method of weighting query terms for ad-hoc retrieval. In *Proc. of the 19th Annual ACM SIGIR Conference*, pages 187–195, 1996.
- [13] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proc. of the 24th Annual ACM SIGIR Conference*, pages 120–127, September 2001.
- [14] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999.
- [15] A. Pirkola and K. Jarvelin. Employing the resolution power of search keys. *Journal of the American Society for Information Science and Technology*, 52(7):575–583, 2001.
- [16] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. of the 21st Annual ACM SIGIR Conference*, pages 275–281, 1998.
- [17] P. Resnik. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61:127–159, 1996.
- [18] M. Rorvig. A new method of measurement for question difficulty. In *Proceedings of the 2000 Annual Meeting of the American Society for Information Science, Knowledge Innovations*, volume 37, pages 372–378, 2000.
- [19] F. Song and W. B. Croft. A general language model for information retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference*, pages 279–280, 1999.
- [20] T. Sullivan. Locating question difficulty through explorations in question space. In *Proceedings of the 1st ACM/IEEE Joint Conference on Digital Libraries*, pages 251–252, 2001.
- [21] S. K. M. Wong and Y. Y. Yao. An information-theoretic measure of term specificity. *Journal of the American Society for Information Science*, 43(1):54–61, 1992.