Measuring Ranked List Robustness for Query Performance Prediction

Yun Zhou and W. Bruce Croft
Department of Computer Science
University of Massachusetts, Amherst
{yzhou, croft}@cs.umass.edu

ABSTRACT

We introduce the notion of ranking robustness, which refers to a property of a ranked list of documents that indicates how stable the ranking is in the presence of uncertainty in the ranked documents. We propose a statistical measure called the robustness score to quantify this notion. Our initial motivation for measuring ranking robustness is to predict topic difficulty for content-based queries in the ad-hoc retrieval task. Our results demonstrate that the robustness score is positively and consistently correlation with average precision of content-based queries across a variety of TREC test collections. Though our focus is on prediction under the ad-hoc retrieval task, we observe an interesting negative correlation with query performance when our technique is applied to named-page finding queries which are a fundamentally different kind of queries. A side effect of this different behavior of the robustness score between the two types of queries is that the robustness score is also found to be a good feature for query classification.

Received: Nov 11,2006
Revised: Mar 19,2007
Accepted: Jun 10,2007

General Terms

Algorithms, Experimentation, Theory

Keywords

Ranking robustness, query performance prediction, query classification, named-page finding, ad-hoc retrieval

1. INTRODUCTION

In a typical retrieval system, a user forms a query according to his information need and a number of documents are presented to the user by the retrieval system in response to the query. Query performance prediction refers to the process of estimating the quality of the output of a retrieval system in response to a user's query *without* any relevance information. Compared to the long history of developing sophisticated retrieval models for improving performance in IR, research on predicting query performance is still in its early stage. However, researchers have started to realize the importance of this problem and a number of new methods have been proposed for prediction recently [27]. The ability to predict query performance has the potential of a fundamental impact both on the user and the retrieval system.

From the perspective of a user, performance prediction provides valuable feedback that can be used to direct a search. For example, when the retrieved documents are estimated to be of low quality, the user may rephrase his query or be more willing to cooperate with the system to improve retrieval effectiveness, such as providing relevance feedback. With the help of prediction, the user can quickly form a good query to acquire satisfying results for his information need. Otherwise, the user must spend time reading the returned documents to rewrite the query when the results for the initial query are not satisfactory.

On the other hand, from the perspective of a retrieval system, performance prediction is the first step at solving the crucial problem of retrieval consistency. Current retrieval systems are evaluated by the *average* effectiveness on a fixed set of queries. Although failures on a small number of queries may not have a significant effect on average performance, users who are interested in these queries are unlikely to be tolerant of this kind of deficiency. A reliable system that always produces acceptable retrieval performance is more preferred by users than another system that works extremely well on a number of queries but occasionally makes terrible mistakes. To improve the consistency of retrieval systems, we first need to distinguish poorly-performing queries by performance prediction techniques. The important role of performance prediction in improving retrieval consistency has been recognized by the IR community. For example, in 2003, the Robust Track [2,22] was proposed by TREC which addresses the problem of enhancing the retrieval of poorly-performing queries. As the first footprint in finding a solution to this problem, the Robust Track requires systems to rank the queries by predicted effectiveness to investigate the capabilities of systems to detect hard queries [27].

However, accurate performance prediction with zero-judgment is not an easy task. The major difficulty of performance prediction comes from the fact that many factors, such as the query, the ranking function and the collection, have an impact on retrieval performance. Each factor affects performance to a different degree and the overall effect is hard to predict accurately. Therefore, it is not surprising to notice that simple features, such as the frequency of query terms in the collection [10] and the average IDF of query terms [25], do not predict well. In fact, most of the successful techniques are based on measuring some characteristics of the retrieved document set (usually in the form of a ranked list) to estimate performance. For example, the clarity score [4] measures the coherence of a list of documents by the KL-divergence between the query model and the collection model.

In this paper, we investigate another property of a ranked list of documents called ranking robustness which refers to how stable the ranking is in the presence of uncertainty in the ranked documents. This method was first introduced in [30]. The idea of predicting retrieval performance by measuring ranking robustness is inspired by a general observation in noisy data retrieval that the degree of ranking robustness against noise is correlated with retrieval performance. Regular documents also contain "noise" if we interpret noise as uncertainty. We propose a statistical measure called the robustness score to quantify the notion of ranking robustness. For content-based queries in the traditional ad-hoc retrieval task, we demonstrate that the robustness score significantly and positively correlates with query performance in a variety of TREC test collections. In comparison to the clarity score method, our experimental results show that the robustness score performs better than or at least as good as the clarity score. Although the robustness score is initially designed for estimating topic difficulty, we also explore the relation between the robustness score and retrieval performance of named-page finding (NP) queries. An interesting negative correlation with NP-query performance is observed, suggesting the fundamental difference in the retrieval processes for the two types of queries. Considering the opposite behavior of the robustness score between these two types of queries, our further investigation reveals that the robustness score is a good feature to distinguish between the two types.

The rest of this paper is organized as follows. Section 2 describes related work. In section 3, we propose a statistical measure called the robustness score to quantify the notion of ranking robustness. In section 4, we present our evaluations that show the effectiveness of our approach. In section 5, we summarize the main conclusions of this paper.

2. RELATED WORK

2.1 Query Performance Prediction

Prediction of query performance has long been of interest in information retrieval and has been investigated under different names such as query-difficulty or query-ambiguity. Query prediction is a challenging task as shown in [27] and [21]. Some of the first success at addressing this task was demonstrated by the clarity score method proposed in [4].

Recently, a number of prediction methods have been tried since the introduction of the TREC Robust Track in 2003. In the Robust Track systems are required to rank the queries by predicted performance, with the goal of utilizing the prediction capability to do query-specific processing. One thing we want to point out is that most study on performance prediction focuses on content-based queries in the ad-hoc task. At the time of writing this paper, we know of no published work that explicitly addresses other types of queries such as named-page finding queries.

Generally speaking, current prediction methods extract features of retrieval and compute the performance score for each query by using the features to estimate the query performance. One way to measure the quality of the performance prediction methods is to compare the rankings of queries based on their actual precision (such as MAP) with the rankings of the same queries ranked by their performance scores (that is, predicted precision).

Some researchers have used IDF-related (inverse document frequency) features as predictors. For example, Tomlinson et al. [25] adopted the weighted average IDF of the query terms for predicting. He and Ounis [10] proposed a predictor based on the standard deviation of the IDF of the query terms. Plachouras [20] represented the quality of a query term by Kwok's inverse collection term frequency. The above IDF-based predictors showed some moderate correlation with query performance. These predictors are easy to compute but they do not take the retrieval algorithms into account and thus are unlikely to predict query performance well.

Inspired by the success of the clarity score, some researcher have proposed methods that are related to the ideas in the clarity score technique. Amati [1] proposed to use the KL-divergence between a query term's frequency in the top retrieved documents and the frequency in the whole collection, which is very similar to the definition of the clarity score. He and Ounis [10] proposed a simplified version of the clarity score where the query model is estimated by the term frequency in the query. Motivated by the observation that the clarity score indicates the specificity of a query, they [10] also proposed the notion of the query scope, which is quantified as the percentage of documents that contain at least one query term in the collection. Diaz and Jones [6] extended clarity scores to include time features. They showed that using these time features together with clarity scores improves prediction.

Realizing that the retrieved document set provides valuable information for estimating retrieval performance, a few researchers have focused on investigating properties of the search results that may relate to search performance. Our approach and the clarity method fall into this category. Another example is that Carmel et al. [5] found that the distance measured by the Jensen-Shannon divergence between the retrieved document set and the collection is significantly correlated to average precision. Vinay et al.[26] propose four measures to capture the geometry of the top retrieved documents for prediction. The most effective measure they found is the sensitivity to document perturbation, an idea somewhat similar to our idea. Generally speaking, techniques that make use of the search results for prediction are more accurate that those that do not.

Other researchers have applied machine learning techniques for prediction. For example, Elad Yom-Tov et al. [28] proposed a histogram-based predictor and a decision tree based predictor. The features used in their models were the document frequency of query terms and the overlap of top retrieval results between using the full query and the individual query terms. Their idea was that well-performing queries tend to agree on most of the retrieved documents. Kwok et al. [15,16] built a query predictor using support vector regression. For features, they chose the best three terms in each query and used their log document frequency and their corresponding frequencies in the query. They also included the number of top retrieved documents that contain some or all query terms as a feature. They observed a small correlation between predicted and actual query performance. Using visual features, such as titles and snippets, from a surrogate document representation of retrieved documents, Jensen et al. [7] trained a regression model with manually labeled queries to predict precision at the top 10 documents (P@10) in the Web search. They reported moderate correlation with P@10.

2.2 Information Retrieval on Noisy Data

With regard to text document collections in information retrieval, it is often convenient to assume that the contents of the collections are clean and free of errors. With the advent of large collections of multimedia documents (such as audio or image document), techniques such as OCR (optical character recognition) or ASR (automatic speech recognition) have been widely used to extract text from multimedia archives. In the following description, the text output of a recognition process applied to multimedia documents is *noisy data* or *corrupted data* since the recognition process is error prone and brings significant levels of noise to the data. The recognition process that produces corrupted data is *data corruption*.

One of the core problems in the field of information retrieval on corrupted data is to explore the impact of data corruption on retrieval effectiveness in order to design a ranking function that is robust to unexpected errors in corrupted data. Here a robust retrieval model means that some changes in document or collection statistics caused by data corruption do not alter the retrieval results much compared to retrieval on perfect documents (that is, the results of a recognition process with 100% accuracy).

A general observation about experiments on investigating the effects of data corruption is that as retrieval effectiveness improves, the ranking function becomes more robust against data corruption. For example, Lopresti and Zhou [11] explored the effectiveness of three retrieval functions on simulated OCR noisy data. They found that the ranking of the three functions with respect to retrieval effectiveness is the same as their ranking with respect to their ability to deal with simulated noise. Another example is that Singhal, Salton and Buckley [23] proposed a new robust length normalization method to alleviate the problem that the regular cosine normalization is sensitive to OCR errors. Although the original motivation for this technique was to deal with OCR data corruption, surprisingly they found that the new normalization scheme also brought significant improvements on correct text collections in comparison to the original cosine normalization. Moreover, Mittendorf [17] studied data corruption effects on retrieval and presented a theorem on ranking robustness that partially explained the phenomenon that retrieval performance on corrupted data is often correlated with the degree of resilience against noise.

The above work reveals the interesting relationship between ranking robustness and retrieval performance. Although this work was done in the context of retrieval on noisy data, clean documents in regular retrieval also contain "noise" if we interpret noise as uncertainty. In the remaining of this paper, we will propose a framework to quantify ranking robustness and show its correlation with query performance.

3. MEASURE RANKING ROBUSTNESS

The notion of ranking robustness originates in the field of noisy data retrieval, where retrieval is performed on the output of a recognition process that exacts text from multimedia archives. Ranking robustness in noisy data retrieval refers to a property of a ranked list of documents that indicates how stable the ranking is in the presence of noise brought by the recognition process. Note that clean documents also contain "noise" if we generalize the notion of noise from recognition errors to uncertainty in text documents. For example, the meaning of a document may remain the same even after adding or deleting some words. Synonymy and homonymy are another two popular examples that can bring uncertainty to clean text documents. Therefore, we can extend the notion of ranking robustness to regular ad-hoc document retrieval. In essence, ranking robustness reflects the ability of a retrieval system to handle uncertainty.

The idea of predicting retrieval performance by measuring ranking robustness is inspired by a general observation in noisy data retrieval that the degree of ranking robustness against noise is positively correlated with retrieval performance. We hypothesize that when it comes to regular ad-hoc retrieval, the positive correlation between robustness and performance still holds. Our hypothesis will be thoroughly examined in the next section.

Next we describe our way of measuring ranking robustness in regular retrieval. We begin by considering how to calculate ranking robustness in noisy data retrieval. If we can acquire a clean version of the corrupted data, one straightforward way is to compare a ranked document list from the corrupted collection to the corresponding ranked list from the perfect collection using the same query and ranking function. With regard to regular document retrieval, usually documents are assumed to be free of corruption. To simulate data corruption, we assume that there exists a noisy channel which is analogous to the recognition process in noisy data retrieval. Documents are corrupted after going thought the channel. One way to implement the noisy channel is to design a document model for each document (Document models are distributions over words or other linguistic units). One corrupted version of the original document is one random sample from the corresponding document model.

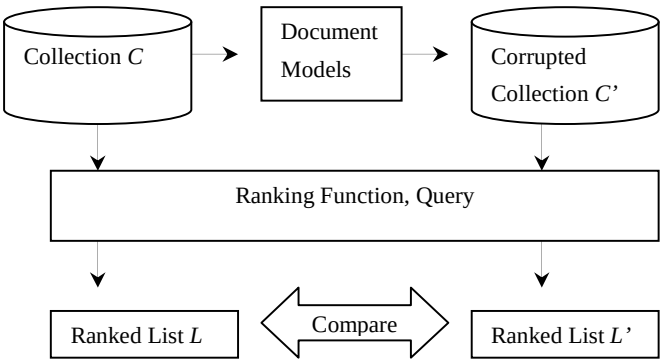


Figure 1: Robustness Score Calculation

Specifically, suppose we have query Q, ranking function G and collection C. We generate corrupted collection C' by sampling from the document models of the documents in C. Then we perform retrieval on both C and C' and two ranked list L and L' are returned respectively. Finally we compute the similarity between the two rankings. Note that L is a fixed ranked list while L' is a random variable. We call the expected similarity between L and L' the robustness score and use it to measure ranking robustness. This process is illustrated in Figure 1.

Let us formally define the robustness score. Consider query Q and a document collection of M documents $C=(D_1,D_2,\ldots D_M)$. Let V

denote the size of vocabulary, both query Q and the documents are represented as vectors of indexed term counts, that is,

$Q=(q_1,q_2,\ldots q_V)\in N^V$

$D_k=(D_{k,1},D_{k,2},\ldots D_{k,V}) \in N^V$

where $D_{k,i}$ is the number of times that term i appears in document D_k and q_j is the number of times that term j appears in query Q. N denotes nonnegative integer and N^V denotes a V-dimension vector space of nonnegative integer. Under our representation, collection C is a $M\times V$ matrix with nonnegative integer entries, that is, $C\in S(M\times V)$, where $S(M\times V)$ denotes the set of a $M\times V$ matrix with nonnegative integer entries . The rows of matrix C can be viewed as a set of documents represented by V-dimension vectors.

We introduce a few definitions before we show the computation of the robustness score.

***Definition 1*:** Retrieval Function $G(D,Q)$

retrieval function $G(D,Q)$ maps query Q and document D into a real number, that is , $G(D,Q)\in R, D\in N^V, Q\in N^V$

***Definition 2*:** Ranked List $L(Q,G,C)$

Let S_M denote the set of permutation of $\{1,2..M\}$. Ranked list $L(Q,G,C)\in S_M$ is a permutation of the documents in collection C that describes the ordering of documents by decreasing $G(D,Q)$ where $D\in C$

***Definition 3*:** Document Model X_k and Probability Mass Function (pmf) $f_{X_k}(x)$

We assume that document D_k, k$\in[1,M]$, corresponds to document model X_k which is a V-dimension multivariate distribution and can be represented by a random vector $X_k=(X_{k,1},X_{k,2},\ldots X_{k,i},..X_{k,V})\in N^V$, where random variable $X_{k,i}$ denotes the number of times term i occurs. The joint pmf of X_k is the function defined by $f_{X_k}(x)=f_{X_k}(x_1,\ldots,x_V)=\Pr(X_{k,1}=x_1,\ldots,X_{k,V}=x_V)$

where $x=(x_1,\ldots,x_V)\in N^V$.

***Definition 4*:** Ranking Similarity $SimRank(L_1,L_2)$

Given two ranked list $L_1(Q,G,C_1)$ and $L_2(Q,G,C_2)$, function $SimRank(L_1,L_2)$ returns a real number that measures the similarity between the two ranked lists.(we assume that the documents in C_1 have one-to-one correspondence to the documents in C_1). Moreover, $SimRank(L_1,L_2)$ should be bounded.

***Definition 5*:** Random Collection X

Given document model $X_1,\ldots X_M$, where X_k (k$\in[1,M]$) is a V-dimension random vector, we define random collection $X=(X_1,X_2,\ldots X_M)$,that is, X is a $M\times V$ matrix whose entries consist of random nonnegative integers from some distributions. The pmf of X is the function defined by $f_X(T)=f_X(t_1,\ldots,t_M)=\Pr(X_1=t_1,\ldots,X_M=t_M)$, where X_k denotes the k-th row of X and $t_k\in N^V$, k$\in[1,M]$.

With the above definitions, we give the definition of the robustness score.

Given query $Q\in N^V$, retrieval function G, collection $C=(D_1,D_2,\ldots D_M)\in S(M\times V)$ and random collection $X=(X_1,X_2,\ldots X_M)$, the robustness score is defined as the expected value of random variable $SimRank(L(Q,G,C),L(Q,G,X))$:

$$\begin{aligned}&Robustness\ Score(Q,G,C,X)=E\{SimRank(L(Q,G,C),L(Q,G,X))\}\\&=\sum_{T\in S(M\times V)} SimRank(L(Q,G,C),L(Q,G,T))f_X(T) \qquad (1)\end{aligned}$$

To make Equation 1 feasible to calculate, we further make the following five assumptions:

(1) We assume independence between any two document models X_i and X_j, that is,

$$f_X(T)=f_X(t_1,t_2,\ldots t_M)=\prod_{k=1}^{M}\Pr(X_k=t_k)=\prod_{k=1}^{M}f_{X_k}(t_k) \qquad (2)$$

(2) Instead of the whole collection, only the top J retrieved documents in $L(Q,G,C)$ and the corresponding J documents in $L(Q,G,X)$ are used to compute the similarity between the two ranked lists. For the purpose of rank comparison, the corresponding J documents in $L(Q,G,X)$ will shift up in rank and form a new ranked list of length J.

(3) The Spearman rank correlation coefficient [12] is adopted to compute the value of function $SimRank(L_1,L_2)$ in Equation 1. The coefficient ranges from -1 to 1. A value close to 1 means a perfect positive correlation between the two rankings and a value close to -1 means a perfect negative correlation. If the two rankings have almost no correlation, the correlation coefficient will be close to zero.

(4) For each document model, we assume independence between any terms. We also assume the term frequencies in the sampled document follow Poisson distributions with the means equal to the corresponding term frequencies in the original document. Modeling term frequencies by Poisson distributions has been widely adopted by other researchers [3] [8]. Furthermore, many retrieval models, such as the query likelihood model, only take query terms into account when ranking documents. In this case, we can simplify Equation 2 by assuming that the frequencies of non-query terms are constant in the sampled document. Formally speaking, given document $D_k=(D_{k,1},D_{k,2},\ldots D_{k,V})$ and query $Q=(q_1,q_2,\ldots q_V)$, probability mass function f_{X_k} of document model $X_k=(X_{k,1},X_{k,2},\ldots X_{k,V})$ is estimated as follows:

$$f_{X_k}(x_1,x_2,\ldots x_V)=\prod_{j=1}^{V}f_{X_{k,j}}(x_j) \quad (3)$$

where $f_{X_{k,j}}(x)$ is given by :

$$if\ (q_j>0)\ AND\ (D_{k,j}>0)$$

$$f_{X_{k,j}}(x)=\Pr(X_{k,j}=x)=\frac{e^{-\lambda}\lambda^x}{x!},\ x\in N,\ \lambda=D_{k,j}$$

$$else$$

$$f_{X_{k,j}}(x)=\Pr(X_{k,j}=x)=\begin{cases}1, if\ x=D_{k,j}\\0, else\end{cases}$$

For better understanding, we give a toy example to show how to generate a simulated document given the original document based on the above assumptions.

Given vocabulary $V=\{a,b,c\}$, query $Q=\{a\}$ and document $D_1=\{a,a,b,b,b\}$, Q and D_1 are represented by 3-dimension vector [1,0,0] and [2,3,0] respectively. Let N(D_1) denotes a simulated

document generated from X_1 ,that is, the document mode of D_1 . Since term *c* does not occur in D_1 , it will not occur in N(D_1). Since term *b* is a non-query term and it occurs three times in D_1, it will occur exactly three times in N(D_1). The occurrence frequency of term *a* in N(D_1) is a random number determined by Poisson distribution P(λ) with λ=2 because term *a* occurs twice in D_1. For example, {*a,a,a,b,b,b*} and {*a,b,b,b*} are two possibilities of N(D_1).

(5) The expectation in Equation 1 is very hard to evaluate directly. Instead, we independently draw *K* samples *T(1),T(2),..T(K)* from $f_X(T)$ to approximate the expectation, that is, Equation 1 is estimated as:

$$Robustness\ Score(Q,G,C,X) \cong \frac{1}{K}\sum_{i=1}^{K} SimRank(L(Q,G,C), L(Q,G,T(i))) \qquad (4)$$

where *T(i)* is a sample independently drawn from $f_X(T)$ which is determined by Equation 2 and 3.

The error of this estimation is proportional to the reciprocal of the square root of *K* [13]. According to our experiments, we find that a relatively small value of *K* is good and stable enough for query performance prediction.

In summary, evaluating robustness takes the following steps. First, we perform retrieval with query *Q* and retrieval function *G*. Then we generate *J* simulated documents using the document models of the top *J* documents retrieved and rank the simulated documents with the same query and retrieval function. The similarity between the two ranked lists is computed using the Spearman rank correlation coefficient. We repeat this *K* times and the average of the Spearman correlation coefficient is the robustness score.

We briefly explain why the robustness score defined above gives us useful information on retrieval performance. A low robustness score means that after document perturbation the ranking function provides a very different ranking compared to the original ranking. The low robustness score suggests that the degree of correlation between documents in the ranked list is low and the original ranking is more like a random ranking. In other words, the low robustness score is likely to correspond to a poorly-performing retrieval that returns a ranked list of loosely related topic covering many topics.

In the above discussion, we assume that the retrieval task is the traditional ad-hoc retrieval based on topic relevance. We will show later on that the use of the robustness score to predict retrieval performance is particularly appropriate for content-based queries. However, with regard to named-page finding queries that often have only a single relevant document, the expected positive correlation with query performance may not exist any more. This is largely due to the fact that top ranked documents in the ranked list in response to a named-page finding (NP) query are not necessarily related while those documents are connected more or less by topic in the case of ad-hoc retrieval.

4. EXPERIEMNTAL RESULTS

Our evaluation focuses on performance prediction within the context of ad-hoc retrieval at which the robustness method primarily aims. In addition, we investigate the effect of this technique on named-page finding queries. Namely, we consider the issue of predicting query performance for the two types of queries: content-based and Named-Page (NP) finding queries, corresponding to the ad-hoc retrieval task and the Named-Page finding task respectively. Other than performance prediction, we also investigate the possibility of utilizing the robustness score for query classification, motivated by the different behavior of the robustness score in the two types of queries observed in our prediction experiments.

4.1 Prediction for Content-based Queries

In this section, we present the results of predicting query performance by the robustness score within the context of the ad-hoc retrieval task. We adopt the clarity method as our baseline. Query performance is measured by average precision.

First, we study the correlation with average precision. Our results show that robustness scores have statistically significant correlation with average precision across a variety of TREC collections. We note that the clarity score is barely correlated with query performance on the GOV2 collection while the correlation between the robustness score and query performance remains significant. We also observe that a combination of the two usually performs better than either one when used in isolation.

Second, we perform a linear regression analysis to evaluate the ability to directly predict the value of average precision. This analysis reveals that the robustness score predicts the value of average precision better than the clarity score. Again, we observe further improvements with a combination of the two.

Our experiments use a variety of TREC collections and the web collection GOV2. All queries used in our experiments are titles of TREC topics. Table 1 gives the summary of these test collections.

Table 1 Summary of test collections for Content-based Queries

TREC	Collection	Topic Number	Number of Document
1+2+3	Disk 1+2+3	51-150	1,078,166
4	Disk 2+3	201-250	567,529
5	Disk 2+4	251-300	524,929
Robust 2004	Disk 4+5 minus CR	301-450; 601-700[1]	528,155
Terabyte 2004 (ad-hoc task)	GOV2	701-750	25,205,197
Terabyte 2005 (ad-hoc task)	GOV2	751-800	25,205,197

With regard to the calculation of the robustness score, we use the query likelihood model [24] with Dirichlet smoothing as the ranking function (Dirichlet prior μ is set to 1000). We set

[1] Topic 672 is removed because of no relevant documents.

parameter *K* in Equation 4 to 100 and choose top 50 documents to compute the rank similarity in Equation 4. We tried different values of *K* ranging from 10 to 500000 and found that the results change very little starting from 100. This means we do not have to require a large number of samples to compute robustness scores.

For computing the clarity score, we use the equations defined in [4] .The document model is estimated by using Dirichlet smoothing with Dirichlet prior μ =1000. Relevance models are mixed from Jelinek-Mercer smoothed document models with λ =0.6.

To obtain average precision, all document retrieval is done by using the query-likelihood model and the results are evaluated by the trec_eval program. Again, Dirichlet smoothing with Dirichlet prior μ =1000 is used for smoothing.

4.1.1 Correlation with Average Precision

We measure the correlation with average precision by both the Kendall's rank correlation test [12] and the Pearson's correlation test [14]. Kendall's rank correlation is a non-parametric test since it does not assume any distributions of both variables. In our experiments, Kendall's rank correlation is used to compare the ranking of queries by average precision to the ranking by the clarity scores or the robustness scores of these queries. Pearson's correlation reflects the degree of linear relationship between the two variables[2].The values of both kinds of correlation range between -1.0 and 1.0 where -1.0 means perfect negative correlation and 1.0 means perfect positive correlation.

Table 2 Pearson's correlation coefficient for correlation with average precision, for robustness score, clarity score and a linear combination of the two features. Bold cases mean the results are statistically significant at the 0.05 level.

TREC	Robustness Score	Clarity Score	Robustness +Clarity
TREC123	**0.434**	**0.335**	**0.469**
TREC4	**0.613**	**0.430**	**0.582**
TREC5	**0.454**	**0.366**	**0.507**
Robust 04	**0.550**	**0.507**	**0.613**
Terabyte04	**0.341**	**0.305**	**0.374**
Terabyte05	**0.301**	0.206	**0.362**

Table 3 Kendall's rank correlation coefficient for correlation with average precision, for robustness score, clarity score and a linear combination of the two features. Bold cases mean the results are statistically significant at the 0.05 level.

TREC	Robustness Score	Clarity Score	Robustness +Clarity
TREC123	**0.329**	**0.331**	**0.370**
TREC4	**0.548**	**0.353**	**0.499**
TREC5	**0.328**	**0.311**	**0.345**
Robust 04	**0.392**	**0.412**	**0.460**
Terabyte04	**0.213**	0.134	**0.226**
Terabyte05	**0.208**	0.171	**0.252**

The results for correlation with average precision are presented in table 2 and 3. When we combine the clarity score and the robustness score, we adopt a simple linear combination, that is, (1-α)×clarity score+α×robustness score. For the collections other than TREC 123, we use the α that yields the highest value of Pearson's coefficient on TREC123. For TREC123, we use the best α on Robust 2004. In fact, we find that the optimal linear combination weight changes little across our test collections. Note that when using linear regression to combine the two, we essentially apply learning to our method. But we have only one parameter and we find the regression generalizes well.

From these results, we first observe statistically significant correlation between the robustness scores and the average precision over all test collections no matter which metric is adopted. The extent of the correlation in the Robust 2004 Track is visible in Figure 2 as a linear trend for average precision of queries to increase as their robustness score increases.

Second, we see that the linear combination of the two features usually performs better than either one when used in isolation. This is within our expectation since clarity scores and robustness scores measure two different properties of a ranked document list.[3] Note that the only exception occurs in TREC 4 because the robustness scores correlate with the average precision much better than the clarity scores.

Third, the robustness score shows a stronger linear relationship with average precision compared to the clarity score. The linear regression analysis performed in the next section will further confirm this observation.

We observe that the performance of the clarity score drops greatly on the GOV2 collection. We speculate that this is due to the fact that there are a relatively large number of low quality documents in this collection. Moreover, it seems that this characteristic has a more negative impact on clarity scores than on robustness scores. To understand this, let us recall that the clarity score measure the degree of dissimilarity between the language usage associated with the query and the generic language of the collection. The

[2] Here the two variables refer to the actual query performance (measured by average precision) and the predictor.

[3] We also examine the correlation between the clarity score and the robustness score. We observe the correlations measured by Pearson's coefficient range from 0.27 to 0.63 on the four TREC collections. We find almost no correlation on the two Web collections. We see that there are relations between the two measures, but they are not very similar to each other. Otherwise, a combination of the two would not lead to further improvement.

ability of clarity scores to predict query performance is based on the following assumption: a query whose highly ranked documents contain many relevant documents (high query performance) is likely to receive a high clarity score because these highly ranked documents tend to be about a single topic and therefore have unusual word usage. However, when it comes to large web collections, the low quality documents retrieved in respond to a query are likely to have unusual word distributions[29], resulting in high clarity scores. In other words, the clarity score method can not distinguish whether a high clarity score is caused by a small number of topic terms in the query language model or by the noise from the low quality documents retrieved.

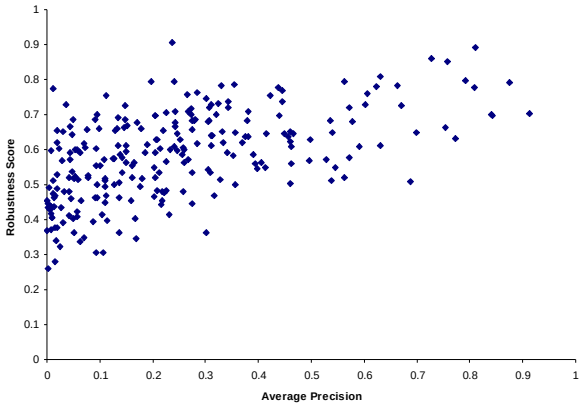


Figure 2: Average precision versus robustness score for the 249 title queries from the Robust 2004 Track.

4.1.2 Linear Regression Analysis

Both Kendall's rank correlation and Pearson's correlation are not capable of directly predicting average precision scores. To address this problem, we adopt the linear regression technique which yields an equation that predicts the values of average precision from predictors. Although there are fancier non-linear models, linear regression models often perform better in situations with sparse data or highly noisy data . Moreover, the linear regression analysis provides an adequate and interpretable description of how the predictors affect the dependent variable. In this section, we first evaluate the linear prediction quality of the clarity score and the robustness score. Then we investigate the relative importance of each predictor in terms of prediction power.

Table 4 Coefficient of determination (R-square) from linear regression: the dependent variable is average precision. The predictor (independent variable) is either the robustness score or the clarity score or a combination of the two.

TREC	Robustness Score only	Clarity Score only	Robustness +Clarity
TREC123	0.188	0.112	0.220
TREC4	0.376	0.185	0.339
TREC5	0.206	0.134	0.257
Robust 04	0.302	0.257	0.376
Terabyte04	0.116	0.093	0.140
Terabyte05	0.091	0.042	0.131

One common way to measure how well a linear regression model fits data is the so-called coefficient of determination or R-square. The range of R-square is between 0 and 1 and a high value means fitting well. Here we perform simple linear regression and the predictor is either the robustness score or the clarity score or the linear combination of the two. Table 4 shows the results which are consistent to what we have observed in Table 2 and 3. For example, we see that the robustness scores fit the average precision much better than the clarity scores on all collections. The goodness-of-fit is low on the GOV2 collection. Again, we observe that the linear combination of the two predictors often boost the quality of linear regression. The effect of linear regression between average precision and robustness score for the 50 title queries from the TREC4 collection is shown in Figure 3.

To identify the predictor that bestows the greatest impact on the dependent variable, we compare the regression coefficients of the two predictors. However, the values of the original regression coefficients depend on both the importance of each predictor and the variance of that predictor. To make a fair comparison, we adopt the standardized regression coefficient called Beta that eliminates the influence of variance. The standardized coefficient is what the regression coefficient would be if the model were fitted to standardized data, that is, if from each observation we subtracted the sample mean and then divided by the sample deviation. Hence, the magnitudes of these Beta values represent the importance of each predictor. Table 5 shows the results for standardized regression coefficients. We used the SPSS software to compute the standardized regression coefficients. We observe the similar trends as in Table 4. Based on the results from table 4 and 5, our results suggest that when using linear regression robustness scores predict average precision better than clarity scores.

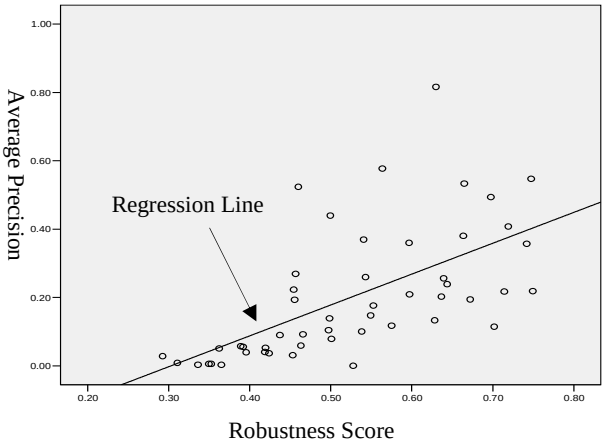


Figure 3: Linear regression between average precision and robustness score for the 50 title queries from the TREC4

Table 5 standardized regression coefficients (Beta) from multiple linear regression: the dependent variable is average precision. The two predictors are the robustness score and the clarity score.

Collection	Robustness Score	Clarity Score
TREC123	0.357	0.195
TREC4	0.568	0.071
TREC5	0.376	0.246
Robust 04	0.396	0.311
Terabyte 04	0.270	0.216
Terabyte 05	0.314	0.224

4.2 Prediction for Named-Page Finding Queries

In the previous section, we have demonstrated that the robustness score consistently correlate with topic difficulty. In this section, our goal is to examine whether there is any relationship between the robustness score and the performance of name-page finding (NP) queries.

The data sets used for evaluation come from the Named-Page finding topics of the Terabyte Tracks of 2005 and 2006 and we name them TB05-NP and TB06-NP respectively. Table 6 gives more details on the two data sets. We adopt the mixed language model [18][19] for our named-page finding retrieval. Retrieval parameters are the same as in [18]. Retrieval performance of individual NP queries is measured by the reciprocal rank of the first correct answer. We use the correlation with the reciprocal ranks measured by the Pearson's correlation test to evaluate prediction quality. The results are presented in Table 7. Again, our baseline is the clarity score.

For the clarity score, we tried different parameters and found that using the first ranked document to build the query model yields the best prediction accuracy. This makes sense because NP-query performance heavily depends on the relevance of the first ranked document. From Table 7, we can see that the correlation with query performance on both test sets is low, suggesting that measuring ranked list coherence is not effective for NP queries.

Regarding the robustness score, we observe an interesting and surprising negative correlation with reciprocal ranks. We explain this finding briefly. A high robustness score means that a number of top ranked documents in the original ranked list are still highly ranked after perturbing the documents. The existence of such documents is a good sign of high performance for content-based queries as these queries usually contain a number of relevant documents. However, with regard to NP queries, one fundamental difference is that there is only one relevant document for each query. The existence of such documents can confuse the ranking function and lead to low retrieval performance. Although the negative correlation with retrieval performance exists, it can still be used for prediction and the strength of this correlation is stronger compared to the clarity method as shown in Table 7.

On the other hand, in comparison to the results shown in the previous section for content-based queries, we notice that prediction by the robustness method for named-page finding queries is less accuracy and consistent on average, suggesting that the robustness score is more appropriate for predicting topic difficulty.

Table 7: Pearson's correlation coefficients for correlation with reciprocal ranks on the Terabyte Tracks (named-page finding task) for clarity score and robustness score. Bold cases mean the results are statistically significant at the 0.01 level.

Methods	Clarity	Robust.
TB05-NP	**0.150**	**-0.370**
TB06-NP	0.112	-0.160

Table 6: Data sets used for the named-page fining evaluation

Name	Collection	Topic Number	Query Type
TB05-NP	GOV2	NP601-NP872	NP
TB06-NP	GOV2	NP901-NP1081	NP

4.3 Query Classification

In this section, we show that the robustness score, though originally proposed for performance prediction, is also a good indicator of query types. The use of the robustness score for query classification is motivated by the observation obtained from our prediction experiments that the robustness score behaves very differently between these two types of queries : named-page finding and content-based. In the following experiments, we create a set of content-based queries consisting of all of the 150 ad-hoc title queries from Terabyte Track 2004-2006 and a set of NP queries consisting of 252 NP queries from Terabyte Track 2005.

We first investigate the distributions of robustness scores for NP and content-based queries respectively. Since we do not know what distribution the scores actually follow, we adopt Kernel

density estimation [9] which does not assume any specific form of distribution on the features we want to estimate. Kernel density estimator belongs to a class of estimators called non-parametric density estimators that have no fixed structure and depend upon all data points to reach an estimate. Specifically, for a query set (in our case , the set is either the NP query set or the content-based query set) of size N, we calculate robustness scores x_1, x_2, …x_N for each query and the probability density function f(x) of robustness score on the set is estimated by :

$$\hat{f}(x) = \frac{1}{N\lambda}\sum_{i=1}^{N} K_{\lambda}(x, x_i)$$

where λ is the bandwidth and K_{λ} is a Kernel function.

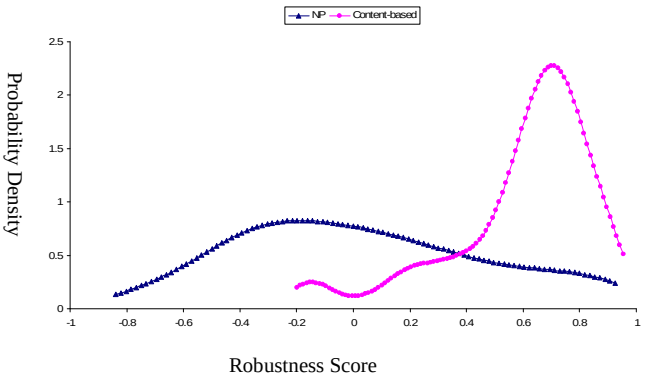


Figure 4: Distribution of robustness scores for named-page finding (NP) and content-based (CB) queries. The NP queries are the 252 NP topics from the 2005 Terabyte Track. The content-based queries are the 150 ad-hoc title from the Terabyte Tracks 2004-2006. The probability distributions are estimated by the Kernel density estimation method.

In this paper we use the Gaussian Kernel. There is a standard way to select the bandwidth (λ) based on minimizing the expected square error between the estimated density and the original density [9]. In this paper, we adopt this method to calculate λ.

The results are shown in figure 4. As we can see, on average content-based queries have a much higher robustness score than NP queries.

Next we test the accuracy of query classification by the robustness score. To this end, we combine the two query sets mentioned above into one query pool. That is, the query pool consists of 150 content-based (CB) queries and 252 NP queries. Each time we pick one query that has not been selected before from the query set and all other queries are used as training data. Our strategy for predicting the type of the selected query is simple: the robustness score classifier will attach a NP (CB) label to the query if the robustness score for the query is below (above) a threshold trained from training data. Table 8 shows the results. For example, the value 25 at the intersection between the second row and the third column means 25 (out of 150) content-based queries are incorrectly labeled as the NP type. That is, the chance of correctly classifying a content-based query is about 83%. From the results in Table 8 we can see that our classifier reaches fairly good classification accuracy.

Table 8 Query Classification Results

	Content(labeled)	NP(labeled)
Content(actual)	125	25
NP(actual)	52	200

5. CONCLUSIONS

In this paper, we introduce the notion of ranking robustness and propose a statistical measure called the robustness score to quantify ranking robustness. The robustness score is an effective tool for predicting retrieval performance of content-based queries. We demonstrate across a variety of test collections that there is a strong positive correlation between the robustness score of a content-based query and the performance of that query. We also apply this technique to predict performance of another fundamentally different kind of queries: named-page finding. An interesting negative correlation between ranking robustness and retrieval performance is observed. However, our experiments show that predicting using robustness scores for named-page queries is less effective compared to content-based queries, suggesting that the robustness technique is more appropriate for predicting topic difficulty. In addition, the opposite behavior of the robustness score between the two types of queries motivates us to investigate the possibility of the use of the robustness score for query classification. Our results show that the robustness score is a good feature for distinguishing between the two query types. The results reported in this paper give fresh insight into our understanding of principles underlying different retrieval tasks and open up possibilities for exploring other applications of ranking robustness.

6. ACKNOWLEGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, and in part by an award from Google. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] Amati G, Carpineto C, Romano G(2004) Query difficulty ,robustness and selective application of query expansion In the proceedings of 26th European Conference on Information Retrieval (ECIR) ,pp 127-137

[2] Bernstein Y, Billerbeck B, Garcia S, et al (2005) RMIT University at TREC 2005: Terabyte and Robust Track. In the Online Proceedings of 2005 Text REtrieval Conference

[3] Bookstein A, Swanson D (1974) Probabilistic models for automatic indexing. Journal of the American Society for Information Science 25,5(1974), pp 312-319

[4] Cronen-Townsend S, Zhou Y, Croft W.B (2002) Predicting query performance. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp 299-306

[5] Carmel D, Yom-Tov E, Darlow A,et al (2006), What Makes a Query Difficult? In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp 390-397

[6] Diaz F, Jones R (2004) Using temporal profiles of queries for precision prediction. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp 18-24

[7] Jensen E.C, Beitzel S.M, Chowdhury A, et al (2005), "Predicting Query Difficulty on the Web by Learning Visual Clues", In Proceedings of the 2005 ACM Conference on Research and Development in Information Retrieval , pp. 615-616

[8] Harter S.P (1975) A probabilistic approach to automatic keyword indexing. Journal of the American Society for Information Science 26,4 and 5(1975), Part I:197-206; Part II:280-289

[9] Hastie T, Tibshirani R, Friedman J.H (2001) .The Elements of Statistical Learning, Chapter 6, Kernel Method Springer press

[10] He B , Ounis I (2004) Inferring query performance using pre-retrieval predictors. In proceedings of the SPIRE 2004. pp 43-54

[11] Lopresti D,Zhou J(2006) Retrieval Strategy for Noisy Text, In symposium on document analysis and information retrieval, pp1-16

[12] Gibbons J.D , Chakraborty S (1992), Nonparametric statistical inference, Marcel Dekker, New York

[13] Kalos M.H, Whitlock P.A (1996) Monte carlo methods, John Wiley & Sons,Inc.

[14] Kreyszig E (1997), Advanced Engineering Mathematics ,chapter 23.10, JONH WILEY&SONS, INC.

[15] Kwok K.L, Grunfeld L, Sun H.L, et al(2004) TREC 2004 Robust Track Experiments Using PIRCS. In the Online Proceedings of 2004 Text REtrieval Conference

[16] Kwok K.L, Grunfeld L, Dinstl, et al (2005) TREC 2005 Robust Track Experiments Using PIRCS. In the Online Proceedings of 2005 Text REtrieval Conference

[17] Mittendorf E, (1998) Data corruption and information retrieval, PhD Thesis, Department of Computer Science, the Katholieke Universiteit Leuven

[18] Metzler D, Strohman T,Zhou Y,et al (2005) Indri at TREC 2005: Terabyte Track, In the Online Proceedings of 2005 TREC

[19] Ogilvie P , Callan J (2003) Combining document representations for known-item search, In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp 143-150

[20] Plachouras V, He B, Ounis I (2004) University of Glasgow at TREC2004: Experiments in Web, Robust, and Terabyte Tracks with Terrier. In the Online Proceedings of 2004 Text REtrieval Conference

[21] Predicting Query Difficulty. SIGIR workshop 2005 http://www.haifa.ibm.com/sigir05-qp/index.html

[22] Robust Track http://trec.nist.gov/tracks.html.

[23] Singhal A,Salton G, Buckley C(1996) Length normalization in degraded text collections. In symposium on document analysis and information retrieval, pp149-162

[24] Song F , Croft W.B (1999), A general language model for information retrieval. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp 270-280

[25] Tomlinson S (2004) Robust, Web and Terabyte Retrieval with Hummingbird SearchServer at TREC 2004. In the Online Proceedings of 2004 Text REtrieval Conference

[26] Vinay V, Cox I. J, Mill-Frayling N, et al (2006), On Ranking the Effectiveness of Searcher, In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp 398-404

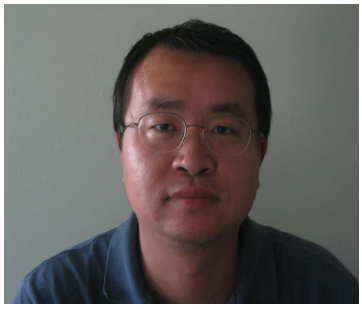
[27] Voorhees E.M. (2004) Overview of the TREC 2004 Robust Track. In the Online Proceedings of 2004 Text REtrieval Conference

[28] Yom-Tov E, Fine S, Carmel D et al (2005) Learning to Estimate Query Difficulty with Applications to Missing Content Detection and Distributed Information Retrieval, In the proceedings of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval , pp. 512-219

[29] Zhou Y, Croft W.B (2005) Document Quality Models for Web Ad Hoc Retrieval, a poster presentation, in the Proceedings of the ACM 14th Conference on Information and Knowledge Management (CIKM 2005), pp. 331-334

[30] Zhou Y, Croft W.B (2006) Ranking Robustness: A Novel Framework to Predict Query Performance, in the Proceedings of the ACM 15th Conference on Information and Knowledge Management (CIKM 2006), pp. 567-574

Author Biographies



Yun Zhou is currently a Ph.D. candidate in the Center for Intelligent Information Retrieval of the Department of Computer Science at the University of Massachusetts Amherst . He will finish his Ph.D. in September of 2007 and join Google Inc. after graduation. His general research interests lie in Information Retrieval (IR) and other closely related fields such as machine learning and natural language processing. His research experience has focused on applying statistical techniques to facilitate information access. His thesis work, under the guidance of Professor W. Bruce Croft, concentrates on predicting retrieval effectiveness without any relevance information. This work offers a pioneering and comprehensive study of retrieval performance prediction in different search environments.

W. Bruce Croft is a Distinguished Professor in the Department of Computer Science at the University of Massachusetts, Amherst, which he joined in 1979. In 1992, he became the Director of the NSF State/Industry/University Collaborative Research Center for Intelligent Information Retrieval (CIIR), which combines basic research with technology transfer to a variety of government and industry partners. Croft's research interests are in formal models of retrieval for complex, text-based objects, text representation techniques, the design and implementation of text retrieval and routing systems, and user interfaces. He has published more than 100 articles on these subjects. This research is also being used in a number of operational retrieval systems.