Ranking Robustness: A Novel Framework to Predict Query Performance

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

ABSTRACT

In this paper, 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. We demonstrate that the robustness score significantly and consistently correlates with query performance in a variety of TREC test collections including the GOV2 collection. We compare the robustness score with the clarity score method which is the state-of-the-art technique for query performance prediction. Our experimental results show that the robustness score performs better than or at least as good as the clarity score. We find 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 notice that a combination of the two usually results in more prediction power.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation, Theory

Keywords

Ranking robustness, query performance prediction

1. INTRODUCTION

In a typical retrieval system, a user forms a query according to his information need and a number of documents (usually in the form of a ranked list) 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 [1]. 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] 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 [1].

In this paper, we develop a method for predicting query performance by computing 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. 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. 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. We demonstrate that the robustness score significantly and consistently correlates with query performance in a variety of TREC test collections including the GOV2 collection. We compare the robustness score with the clarity score method which is the state-of-the-art technique for query performance prediction.

CIKM'06, November 5–11, 2006, Arlington, Virginia, USA.

Our experimental results show that the robustness score performs better than or at least as good as the clarity score.

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 [1] and [3]. Some of the first success at addressing this task was demonstrated by the clarity score method proposed in [4]. Since then, the clarity measure has been the state-of-the-art technique. At the time of writing this paper, we know of no published work that has claimed to achieve the prediction accuracy comparable to or better than the clarity score across a variety of test collections.

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. Generally speaking, these 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). Based on whether training data are needed when computing the performance scores, these methods can be classified into two groups: one that does not need training data and one that does. Our approach that will be introduced in Section 3 is in the first group.

Category I: Does Not Need Training Data

In this category, no training data are required when predicting query performance. Our method that will be introduced in section 3 belongs to this category.

Some researchers have used IDF-related (inverse document frequency) features as predictors. For example, Tomlinson et al. [5] adopted the weighted average IDF of the query terms for predicting. He and Ounis [6] proposed a predictor based on the standard deviation of the IDF of the query terms. Plachouras [7] 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 [8] 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 [6] 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 [6] 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 [9] extended clarity scores to include time features. They showed that using these time features together with clarity scores improves prediction.

Kwok et al. [10] suggests predicting query performance by retrieved document similarity. The basic idea is that when relevant documents occupy the top ranking positions, the similarity between top retrieved documents should be high, based on the assumption that relevant documents are similar to each other. While this idea is interesting, preliminary results are not very promising.

Bernstein et al. [11] estimate the prior probability of each document that will be retrieved by the retrieval system. For a given query, they compare the ranking of documents based on the prior probabilities to the ranking of documents returned from the retrieval system. They hypothesize that if the two ranking are similar, the query will be difficult since the query does not have strong discriminating power. Their results show some limited indication of query performance.

Category II: Needs Training Data

Elad Yom-Tov et al. [12] 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. They reported promising prediction results and showed that their methods were more precise than those used in [13][7][5].

Kwok et al. [13] 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. [14] 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.

We point out that there kinds of predictors may highly depend on the amount and characteristics of available training data if the prediction methods do not generalize well and have to be retrained often.

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 [15] 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 [16] 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 retrieval, the 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.

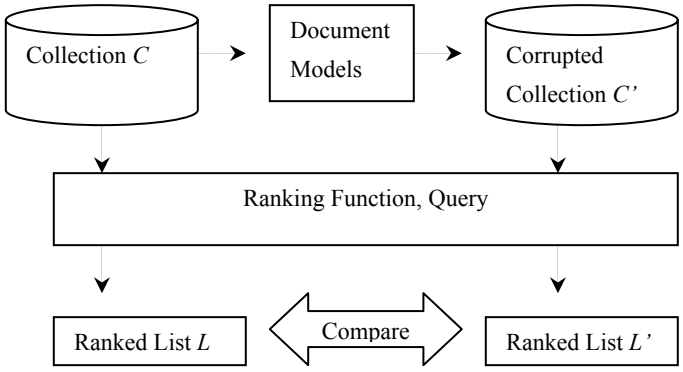

Figure 1: Robustness Score Calculation

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 $C=(D_1,D_2,...D_M)$. Let V denote the size of vocabulary, both query Q and the documents are represented as vectors of indexed term counts, that is,

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

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

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 $X_k = (X_{k,1}, X_{k,2}, ...X_{k,i}, ...X_{k,V}) \in N^V$, 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 $f_{X_k}(x) = f_{X_k}(x_1, ..., x_V) = \Pr(X_{k,1} = x_1, ..., X_{k,V} = x_V)$

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

***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 $X_1,...X_M$, where X_k ($k \in [1,M]$) is a V-dimension random vector, we define random collection $X=(X_1,X_2,...X_M)$,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 $f_X(T) = f_X(t_1, ..., t_M) = \Pr(X_1 = t_1, ..., X_M = t_M)$, 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 $C=(D_1,D_2,...D_M) \in S(M \times V)$ and random collection $X=(X_1,X_2,...X_M)$, the robustness score is defined as the expected value of random variable $SimRank(L(Q,G,C),L(Q,G,X))$:

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

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,

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

(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 [18] 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 [19] [20]. 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 $D_k=(D_{k,1},D_{k,2},...D_{k,V})$ and query $Q=(q_1,q_2,...q_V)$, probability mass function f_{X_k} of document model $X_k=(X_{k,1},X_{k,2},...X_{k,V})$ is estimated as follows:

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

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 $P(\lambda)$ with $\lambda=2$ 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 [21]. 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 the ranking function provides a very different ranking in the presence of simulated noise compared to the perfect ranking(The ranking on the corresponding clean documents without any simulated noise). We assume that the perfect ranking is optimal, that is, performance of any ranking in the presence of simulated noise can not exceed that of the ranking without simulated noise. Under this assumption, a large deviation between the noisy ranking and the perfect ranking indicates that the noisy ranking is ineffective. If the retrieval performance on the documents with simulated noise is low, we have reasons to believe that the performance on the actual collection may also be low.

4. EVALUATION

In this section, we present the results of predicting query performance by the robustness score. 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.

4.1 Experimental setup

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

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	GOV2	701-750	25,205,197
Terabyte 2005	GOV2	751-800	25,205,197

With regard to the calculation of the robustness score, we use the query likelihood model [22] with Dirichlet smoothing as the ranking function (Dirichlet prior μ is set to 1000). We set 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.

4.2 Correlation with Average Precision

We measure the correlation with average precision by both the Kendall's rank correlation test [18] and the Pearson's correlation test [23]. 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.

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

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

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 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[24], 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.

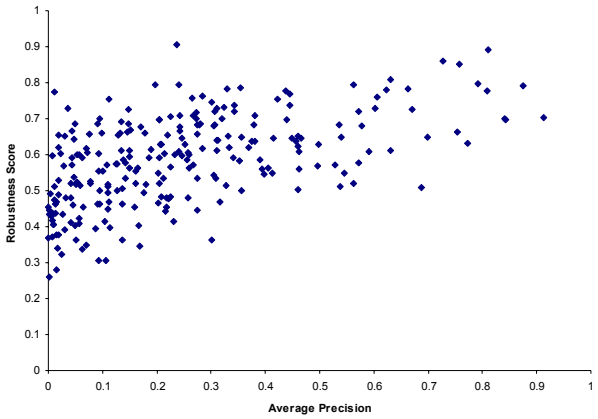

Figure 2: Average precision versus robustness score for the 249 title queries from the Robust 2004 Track.

4.3 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

[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.

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 [25]. 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.

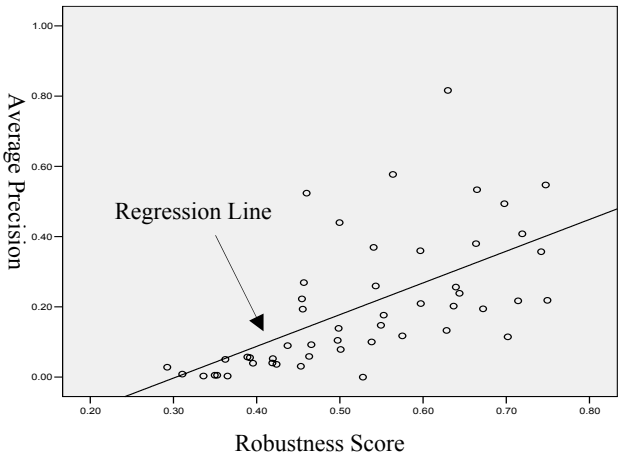

Figure 3: Linear regression between average precision and robustness score for the 50 title queries from the TREC4

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

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. We demonstrate that there is a strong correlation between the robustness score of a test query and the performance of that query. We compare the robustness score with the clarity score method which is the state-of-the-art technique for query performance prediction. Our experimental results show that the robustness score performs better than or at least as good as the clarity score. We observe that the robustness score shows a stronger linear relationship with query performance compared to the clarity score. Therefore, the robustness score can predict the values of average precision more accurately than the clarity score when using a linear regression model. We find 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 notice that a combination of the two usually results in more prediction power. These results give fresh insight into our understanding of principles underlying retrieval and opens up possibilities for developing new techniques in the direction of ranking robustness for predicting or improving retrieval effectiveness.

6. ACKNOWLEGEMENTS

We thank Vanessa Murdock, Ben Carterette and Jiwoon Jeon for their helpful comments on this work. This work was supported by the Center for Intelligent Information Retrieval.

Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

7. REFERENCES

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

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

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

[4] Steve Cronen-Townsend, Yun Zhou, W. Bruce Croft. Predicting query performance. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.(SIGIR 2002)

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

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

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

[8] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. Query difficulty ,robustness and selective application of query expansion In ECIR 2004 ,pp 127-137

[9] F.Diaz and R.Jones. 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.(SIGIR 2004)

[10] K.L. Kwok, L. Grunfeld, N. Dinstl, P. Deng. TREC 2005 Robust Track Experiments Using PIRCS. In the Online Proceedings of 2005 Text REtrieval Conference (TREC 2005)

[11] Y. Bernstein, B. Billerbeck, S. Garcia, N. Lester, F. Scholer, J. Zobe. RMIT University at TREC 2005: Terabyte and Robust Track. In the Online Proceedings of 2005 Text REtrieval Conference (TREC 2005)

[12] E. Yom-Tov, S. Fine, D. Carmel, A. Darlow (2005) "Learning to Estimate Query Difficulty with Applications to Missing Content Detection and Distributed Information Retrieval", 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005), Salvador, Brazil. pp. 512-219

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

[14] Eric C. Jensen, Steven M. Beitzel, Abdur Chowdhury, Ophir Frieder, David Grossman, "Predicting Query Difficulty on the Web by Learning Visual Clues", In Proceedings of the 2005 ACM Conference on Research and Development in Information Retrieval (SIGIR 2005), pp. 615

[15] D.Lopresti and J.Zhou, Retrieval Strategy for Noisy Text, In symposium on document analysis and information retrieval, pp1-16,1996

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

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

[18] J.D. Gibbons and S.Chakraborty, Nonparametric statistical inference, Marcel Dekker, New York,1992

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

[20] Harter,S.P. 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

[21] M.H. Kalos and P.A. Whitlock, Monte carlo methods, John Wiley & Sons,Inc. 1996

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

[23] E. Kreyszig, Advanced Enineering Mathematics ,chapter 23.10, JONH WILEY&SONS, INC. 1997

[24] Yun Zhou, W. Bruce Croft, Document Quality Models for Web Ad Hoc Retrieval, a poster presentation, in the Proceedings of CIKM 2005, pp. 331-334

[25] T. Hastie, R. Tibshirani, J. H. Friedman .The Elements of Statistical Learning, Springer press, 2001