Query Performance Prediction in Web Search Environments

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

ABSTRACT

Current prediction techniques, which are generally designed for content-based queries and are typically evaluated on relatively homogenous test collections of small sizes, face serious challenges in web search environments where collections are significantly more heterogeneous and different types of retrieval tasks exist. In this paper, we present three techniques to address these challenges. We focus on performance prediction for two types of queries in web search environments: content-based and Named-Page finding. Our evaluation is mainly performed on the GOV2 collection. In addition to evaluating our models for the two types of queries separately, we consider a more challenging and realistic situation that the two types of queries are mixed together without prior information on query types. To assist prediction under the mixed-query situation, a novel query classifier is adopted. Results show that our prediction of web query performance is substantially more accurate than the current state-of-the-art prediction techniques. Consequently, our paper provides a practical approach to performance prediction in real-world web settings.

1. INTRODUCTION

Query performance prediction has many applications in a variety of information retrieval (IR) areas such as improving retrieval consistency, query refinement, and distributed IR. The importance of this problem has been recognized by IR researchers and a number of new methods have been proposed for prediction recently [1, 2, 17].

Most work on prediction has focused on the traditional "ad-hoc" retrieval task where query performance is measured according to topical relevance. These prediction models are evaluated on TREC document collections which typically consist of no more than one million relatively homogenous newswire articles. With the popularity and influence of the Web, prediction techniques that will work well for web-style queries are highly preferable. However, web search environments pose significant challenges to current prediction models that are mainly designed for traditional TREC settings. Here we outline some of these challenges.

First, web collections, which are much larger than conventional TREC collections, include a variety of documents that are different in many aspects such as quality and style. Current prediction techniques can be vulnerable to these characteristics of web collections. For example, the reported prediction accuracy of the ranking robustness technique and the clarity technique on the GOV2 collection (a large web collection) is significantly worse compared to the other TREC collections [1]. Similar prediction accuracy on the GOV2 collection using another technique is reported in [2], confirming the difficult of predicting query performance on a large web collection.

Furthermore, web search goes beyond the scope of the ad-hoc retrieval task based on topical relevance. For example, the Named-Page (NP) finding task, which is a navigational task, is also popular in web retrieval. Query performance prediction for the NP task is still necessary since NP retrieval performance is far from perfect. In fact, according to the report on the NP task of the 2005 Terabyte Track [3], about 40% of the test queries perform poorly (no correct answer in the first 10 search results) even in the best run from the top group. To our knowledge, little research has explicitly addressed the problem of NP-query performance prediction. Current prediction models devised for content-based queries will be less effective for NP queries considering the fundamental differences between the two.

Third, in real-world web search environments, user queries are usually a mixture of different types and prior knowledge about the type of each query is generally unavailable. The mixed-query situation raises new problems for query performance prediction. For instance, we may need to incorporate a query classifier into prediction models. Despite these problems, the ability to handle this situation is a crucial step towards turning query performance prediction from an interesting research topic into a practical tool for web retrieval.

In this paper, we present three techniques to address the above challenges that current prediction models face in Web search environments. Our work focuses on query performance prediction for the content-based (ad-hoc) retrieval task and the name-page finding task in the context of web retrieval. Our first technique, called *weighted information gain* (WIG), makes use of both single term and term proximity features to estimate the quality of top retrieved documents for prediction. We find that WIG offers consistent prediction accuracy across various test collections and query types. Moreover, we demonstrate that good prediction accuracy can be achieved for the mixed-query situation by using WIG with the help of a query type classifier. *Query feedback* and *first rank change*, which are our second and third prediction techniques, perform well for content-base queries and NP queries respectively.

Our main contributions include: (1) considerably improved prediction accuracy for web content-based queries over several state-of-the-art techniques. (2) new techniques for successfully predicting NP-query performance. (3) a practical and fully automatic solution to predicting mixed-query performance. In addition, one minor contribution is that we find that the robustness score [1], which was originally proposed for performance prediction, is helpful for query classification.

Related work is discussed in Section 2. We detail our prediction models in Section 3. Experimental results are presented in Section 4 and Section 5 concludes the paper.

2. RELATED WORK

As we mentioned in the introduction, a number of prediction techniques have been proposed recently that focus on content-based queries in the topical relevance (ad-hoc) task. We know of no published work that addresses other types of queries such as NP queries, let alone a mixture of query types. Next we review some representative models.

The major difficulty of performance prediction comes from the fact that many factors 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 [4] and the average IDF of query terms [5], do not predict well. In fact, most of the successful techniques are based on measuring some characteristics of the retrieved document set to estimate topic difficulty. For example, the clarity score [6] measures the coherence of a list of documents by the KL-divergence between the query model and the collection model. The robustness score [1] quantifies another property of a ranked list: the robustness of the ranking in the presence of uncertainty. Carmel et al. [2] 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.[7] propose four measures to capture the geometry of the top retrieved documents for prediction. The most effective measure is the sensitivity to document perturbation, an idea somewhat similar to the robustness score. Unfortunately, their way of measuring the sensitivity does not perform equally well for short queries and prediction accuracy drops considerably when a state-of-the-art retrieval technique (like Okapi or a language modeling approach) is adopted for retrieval instead of the tf-idf weighting used in their paper [16].

The difficulties of applying these models in web search environments have already been mentioned. In this paper, we mainly adopt the clarity score and the robustness score as our baselines. We experimentally show that the baselines, even after being carefully tuned, are inadequate for the web environment.

One of our prediction models, WIG, is related to the Markov random field (MRF) model for information retrieval [8]. The MRF model directly models term dependence and is found be to highly effective across a variety of test collections (particularly web collections) and retrieval tasks. This model is used to estimate the joint probability distribution over documents and queries, an important part of WIG. The superiority of WIG over other prediction techniques based on unigram features, which will be demonstrated later in our paper, coincides with that of MRF for retrieval. In other word, it is interesting to note that term dependence, when being modeled appropriately, can be helpful for both improving and predicting retrieval performance.

3. PREDICTION MODELS

3.1 Weighted Information Gain (WIG)

This section introduces a weighted information gain approach that incorporates both single term and proximity features for predicting performance for both content-based and Named-Page (NP) finding queries.

Given a set of queries Q={Q_s} (s=1,2,..N) which includes all possible user queries and a set of documents D={D_t} (t=1,2…M), we assume that each query-document pair (Q_s,D_t) is manually judged and will be put in a relevance list if Q_s is found to be relevant to D_t. The joint probability P(Q_s,D_t) over queries Q and documents D denotes the probability that pair (Q_s,D_t) will be in the relevance list. Such assumptions are similar to those used in [8]. Assuming that the user issues query $Q_i \in$Q and the retrieval results in response to Q_i is a ranked list L of documents, we calculate the amount of information contained in P(Q_s,D_t) with respect to Q_i and L by Eq.1 which is a variant of entropy called the weighted entropy[13]. The weights in Eq.1 are solely determined by Q_i and L.

$$H_{Q_i,L}(Q_s,D_t) = -\sum_{s,t} weight(Q_s,D_t)\log P(Q_s,D_t) \qquad (1)$$

In this paper, we choose the weights as follows:

$$weight(Q_s,D_t) = \begin{cases} 1/K, if \ s=i \ and \ D_t \in T_K(L) \\ 0, otherwise \end{cases} \qquad (2)$$

$$where\ T_K(L)\ contains\ the\ top\ K\ documents\ in\ L$$

The cutoff rank K is a parameter in our model that will be discussed later. Accordingly, Eq.1 can be simplified as follows:

$$H_{Q_i,L}(Q_s,D_t) = -\frac{1}{K}\sum_{D_t\in T_K(L)} \log P(Q_i,D_t) \qquad (3)$$

Unfortunately, weighted entropy $H_{Q_i,L}(Q_s,D_t)$ computed by Eq.3, which represents the amount of information about how likely the top ranked documents in L would be relevant to query Q_i on average, cannot be compared across different queries, making it inappropriate for directly predicting query performance. To mitigate this problem, we come up with a background distribution P(Q_s,C) over Q and D by imagining that every document in D is replaced by the same special document C which represents average language usage. In this paper, C is created by concatenating every document in D. Roughly speaking, C is the collection (the document set) {D_t} without document boundaries. Similarly, weighted entropy $H_{Q_i,L}(Q_s,C)$ calculated by Eq.3 represents the amount of information about how likely an average document (represented by the whole collection) would be relevant to query Q_i.

Now we introduce our performance predictor *WIG* which is the weighted information gain [13] computed as the difference between $H_{Q_i,L}(Q_s,D_t)$ and $H_{Q_i,L}(Q_s,C)$.Specifically, given query Q_i, collection C and ranked list L of documents, WIG is calculated as follows:

$$\begin{aligned} WIG(Q_i,C,L) &= H_{Q_i,L}(Q_s,C) - H_{Q_i,L}(Q_s,D_t) \\ &= \sum_{s,t} weight(Q_s,D_t)\log\frac{P(Q_s,D_t)}{P(Q_s,C)} = \frac{1}{K}\sum_{D_t\in T_K(L)} \log\frac{P(Q_i,D_t)}{P(Q_i,C)} \qquad (4) \end{aligned}$$

WIG computed by Eq.4 measures the change in information about the quality of retrieval (in response to query Q_i) from an imaginary state that only an average document is retrieved to a posterior state that the actual search results are observed. We hypothesize that WIG is positively correlated with retrieval effectiveness because high quality retrieval should be much more effective than just returning the average document.

The heart of this technique is how to estimate the joint distribution $P(Q_s,D_t)$. In the language modeling approach to IR, a variety of models can be applied readily to estimate this distribution. Although most of these models are based on the bag-of-words assumption, recent work on modeling term dependence under the language modeling framework have shown consistent and significant improvements in retrieval effectiveness over bag-of-words models. Inspired by the success of incorporating term proximity features into language models, we decide to adopt a good dependence model to estimate the probability $P(Q_s,D_t)$. The model we chose for this paper is Metzler and Croft's Markov Random Field (MRF) model, which has already demonstrated superiority over a number of collections and different retrieval tasks [8,9].

According to the MRF model, $\log P(Q_{i,} D_t)$ can be written as

$$\log P(Q_i, D_t) = -\log Z_1 + \sum_{\xi \in F(Q_i)} \lambda_\xi \log P(\xi \mid D_t) \qquad (5)$$

where Z_1 is a constant that ensures that $P(Q_{i,} D_t)$ sums up to 1. $F(Q_i)$ consists of a set of features expanded from the original query Q_i . For example, assuming that query Q_i is "talented student program", $F(Q_i)$ includes features like "program" and "talented student". We consider two kinds of features: single term features T and proximity features P. Proximity features include exact phrase (#1) and unordered window (#uwN) features as described in [8]. Note that $F(Q_i)$ is the union of $T(Q_i)$ and $P(Q_i)$. For more details on $F(Q_i)$ such as how to expand the original query Q_i to $F(Q_i)$, we refer the reader to [8] and [9]. $P(\xi|D_t)$ denotes the probability that feature ξ will occur in D_t. More details on $P(\xi|D_t)$ will be provided later in this section. The choice of λ_ξ is somewhat different from that used in [8] since λ_ξ plays a dual role in our model. The first role, which is the same as in [8], is to weight between single term and proximity features. The other role, which is specific to our prediction task, is to normalize the size of $F(Q_i)$.We found that the following weight strategy for λ_ξ satisfies the above two roles and generalizes well on a variety of collections and query types.

$$\lambda_\xi = \begin{cases} \dfrac{\lambda_T}{\sqrt{|T(Q_i)|}}, \xi \in T(Q_i) \\ \dfrac{1-\lambda_T}{\sqrt{|P(Q_i)|}}, \xi \in P(Q_i) \end{cases} \qquad (6)$$

where $|T(Q_i)|$ and $|P(Q_i)|$ denote the number of single term and proximity features in $F(Q_i)$ respectively. The reason for choosing the square root function in the denominator of λ_ξ is to penalize a feature set of large size appropriately, making WIG more comparable across queries of various lengths. λ_T is a fixed parameter and set to 0.8 according to [8] throughout this paper.

Similarly, $\log P(Q_i,C)$ can be written as:

$$\log P(Q_i, C) = -\log Z_2 + \sum_{\xi \in F(Q_i)} \lambda_\xi \log P(\xi \mid C) \qquad (7)$$

When constant Z_1 and Z_2 are dropped, WIG computed in Eq.4 can be rewritten as follows by plugging in Eq.5 and Eq.7 :

$$WIG(Q_i, C, L) = \frac{1}{K} \sum_{D_t \in T_K(L)} \sum_{\xi \in F(Q_i)} \lambda_\xi \log \frac{P(\xi \mid D_t)}{P(\xi \mid C)} \qquad (8)$$

One of the advantages of WIG over other techniques is that it can handle well both content-based and NP queries. Based on the type (or the predicted type) of Q_i, the calculation of WIG in Eq. 8 differs in two aspects: (1) how to estimate $P(\xi|D_t)$ and $P(\xi|C)$, and (2) how to choose K.

For content-based queries, $P(\xi|C)$ is estimated by the relative frequency of feature ξ in collection C as a whole. The estimation of $P(\xi|D_t)$ is the same as in [8]. Namely, we estimate $P(\xi|D_t)$ by the relative frequency of feature ξ in D_t linearly smoothed with collection frequency $P(\xi|C)$. K in Eq.8 is treated as a free parameter. Note that K is the only free parameter in the computation of WIG for content-based queries because all parameters involved in $P(\xi|D_t)$ are assumed to be fixed by taking the suggested values in [8].

Regarding NP queries, we make use of document structure to estimate $P(\xi|D_t)$ and $P(\xi|C)$ by the so-called *mixture of language models* proposed in [10] and incorporated into the MRF model for Named-Page finding retrieval in [9]. The basic idea is that a document (collection) is divided into several fields such as the title field, the main-body field and the heading field. $P(\xi|D_t)$ and $P(\xi|C)$ are estimated by a linear combination of the language models from each field. Due to space constraints, we refer the reader to [9] for details. We adopt the exact same set of parameters as used in [9] for estimation. With regard to K in Eq.8, we set K to 1 because the Named-Page finding task heavily focuses on the first ranked document. Consequently, there are no free parameters in the computation of WIG for NP queries.

3.2 Query Feedback

In this section, we introduce another technique called *query feedback* (QF) for prediction. Suppose that a user issues query Q to a retrieval system and a ranked list L of documents is returned. We view the retrieval system as a noisy channel. Specifically, we assume that the output of the channel is L and the input is Q. After going through the channel, Q becomes corrupted and is transformed to ranked list L.

By thinking about the retrieval process this way, the problem of predicting retrieval effectiveness turns to the task of evaluating the quality of the channel. In other words, prediction becomes finding a way to measure the degree of corruption that arises when Q is transformed to L. As directly computing the degree of the corruption is difficult, we tackle this problem by approximation. Our main idea is that we measure to what extent information on Q can be recovered from L on the assumption that only L is observed. Specifically, we design a decoder that can accurately translate L back into new query Q' and the similarity S between the original query Q and the new query Q' is adopted as a performance predictor. This is a sketch of how the QF technique predicts query performance. Before filling in more details, we briefly discuss why this method would work.

There is a relation between the similarity S defined above and retrieval performance. On the one hand, if the retrieval has strayed from the original sense of the query Q, the new query Q' extracted from ranked list L in response to Q would be very different from the original query Q. On the other hand, a query distilled from a ranked list containing many relevant documents is likely to be similar to the original query. Further examples in support of the relation will be provided later.

Next we detail how to build the decoder and how to measure the similarity S.

In essence, the goal of the decoder is to compress ranked list L into a few informative terms that should represent the content of the top ranked documents in L. Our approach to this goal is to represent ranked list L by a language model (distribution over terms). Then terms are ranked by their contribution to the language model's KL (Kullback-Leibler) divergence from the background collection model. Top ranked terms will be chosen to form the new query Q'. This approach is similar to that used in Section 4.1 of [11].

Specifically, we take three steps to compress ranked list L into query Q' without referring to the original query.

1. We adopt the *ranked list language model* [14], to estimate a language model based on ranked list L. The model can be written as:

$$P(w\,|\,L)=\sum_{D\in L}P(w\,|\,D)P(D\,|\,L) \qquad (9)$$

where w is any term, D is a document. P(D|L) is estimated by a linearly decreasing function of the rank of document D.

2. Each term in P(w|L) is ranked by the following KL-divergence contribution:

$$P(w\,|\,L)\log\frac{P(w\,|\,L)}{P(w\,|\,C)} \qquad (10)$$

where P(w|C) is the collection model estimated by the relative frequency of term w in collection C as a whole.

3. The top N ranked terms by Eq.10 form a weighted query Q'={(w_i,t_i)} i=1,N. where w_i denotes the i-th ranked term and weight t_i is the KL-divergence contribution of w_i in Eq. 10.

Two representative examples, one for a poorly performing query "Cruise ship damage sea life" (TREC topic 719; average precision: 0.08) and the other for a high performing query "prostate cancer treatments"(TREC topic 710; average precision: 0.49), are shown in Table 1 and 2 respectively. These examples indicate how the similarity between the original and the new query correlates with retrieval performance. The parameter N in step 3 is set to 20 empirically and choosing a larger value of N is unnecessary since the weights after the top 20 are usually too small to make any difference.

Term	cruise	ship	vessel	sea	passenger
KL contribution	0.050	0.040	0.012	0.010	0.009

Table 1: top 5 terms compressed from the ranked list in response to query "Cruise ship damage sea life"

Term	prostate	cancer	treatment	men	therapy
KL contribution	0.177	0.140	0.028	0.025	0.020

Table 2: top 5 terms compressed from the ranked list in response to query "prostate cancer treatments"

To measure the similarity between original query Q and new query Q', we first use Q' to do retrieval on the same collection. A variant of the query likelihood model [15] is adopted for retrieval. Namely, documents are ranked by:

$$P(Q'\,|\,D)=\sum_{(w_i,t_i)\in Q'}P(w_i\,|\,D)^{t_i} \qquad (11)$$

where w_i is a term in Q' and t_i is the associated weight. D is a document.

Let L' denote the new ranked list returned from the above retrieval. The similarity is measured by the overlap of documents in L and L'. Specifically, the percentage of documents in the top K documents of L that are also present in the top K documents in L'. the cutoff K is treated as a free parameter.

We summarize here how the QF technique predicts performance given a query Q and the associated ranked list L. We first obtain a weighted query Q' compressed from L by the above three steps. Then we use Q' to perform retrieval and the new ranked list is L'. The overlap of documents in L and L' is used for prediction.

3.3 First Rank Change (FRC)

In this section, we propose a method called the *first rank change* (FRC) for performance prediction for NP queries. This method is derived from the ranking robustness technique [1] that is mainly designed for content-based queries. When directly applied to NP queries, the robustness technique will be less effective because it takes the top ranked documents as a whole into account while NP queries usually have only one single relevant document. Instead, our technique focuses on the first rank document while the main idea of the robustness method remains. Specifically, the pseudo-code for computing FRC is shown in figure 1.

```
Input: (1) ranked list L={$D_i$} where i=1,100. $D_i$ denotes the i-th ranked document. (2) query Q
1 initialize: (1) set the number of trials J=100000 (2) counter c=0;
2 **for** i=1 to J
3 Perturb every document in L, let the outcome be a set F={$D_i$'} where $D_i$' denotes the perturbed version of $D_i$.
4   Do retrieval with query Q on set F
5   c=c+1 if and only if $D_1$' is ranked first in step 4
6   end of **for**
7   return the ratio c/J
```

Figure 1: pseudo-code for computing FRC

FRC approximates the probability that the first ranked document in the original list L will remain ranked first even after the documents are perturbed. The higher the probability is, the more confidence we have in the first ranked document. On the other hand, in the extreme case of a random ranking, the probability would be as low as 0.5. We expect that FRC has a positive association with NP query performance. We adopt [1] to implement the document perturbation step (step 4 in Fig.1) using Poisson distributions. For more details, we refer the reader to [1].

4. EVALUATION

We now present the results of predicting query performance by our models. Three state-of-the-art techniques are adopted as our baselines. We evaluate our techniques across a variety of Web retrieval settings. As mentioned before, we consider two types of queries, that is, content-based (CB) queries and Named-Page(NP) finding queries.

First, suppose that the query types are known. We investigate the correlation between the predicted retrieval performance and the actual performance for both types of queries separately. Results show that our methods yield considerable improvements over the baselines.

We then consider a more challenging scenario where no prior information on query types is available. Two sub-cases are considered. In the first one, there exists only one type of query but the actual type is unknown. We assume a mixture of the two query types in the second case. We demonstrate that our models achieve good accuracy under this demanding scenario, making prediction practical in a real-world Web search environment.

4.1 Experimental Setup

Our evaluation focuses on the GOV2 collection which contains about 25 million documents crawled from web sites in the .gov domain during 2004 [3]. We create two kinds of data set for CB queries and NP queries respectively. For the CB type, we use the ad-hoc topics of the Terabyte Tracks of 2004, 2005 and 2006 and name them TB04-adhoc, TB05-adhoc and TB06-adhoc respectively. In addition, we also use the ad-hoc topics of the 2004 Robust Track (RT04) to test the adaptability of our techniques to a non-Web environment. For NP queries, we use the Named-Page finding topics of the Terabyte Tracks of 2005 and 2006 and we name them TB05-NP and TB06-NP respectively. All queries used in our experiments are titles of TREC topics as we center on web retrieval. Table 3 summarizes the above data sets.

Name	Collection	Topic Number	Query Type
TB04-adhoc	GOV2	701-750	CB
TB05-adhoc	GOV2	751-800	CB
TB06-adhoc	GOV2	801-850	CB
RT04	Disk 4+5 (minus CR)	301-450;601-700	CB
TB05-NP	GOV2	NP601-NP872	NP
TB06-NP	GOV2	NP901-NP1081	NP

Table 3: Summary of test collections and topics

Retrieval performance of individual content-based and NP queries is measured by the average precision and reciprocal rank of the first correct answer respectively. We make use of the Markov Random field model for both ad-hoc and Named-Page finding retrieval. We adopt the same setting of retrieval parameters used in [8,9]. The Indri search engine [12] is used for all of our experiments. Though not reported here, we also tried the query likelihood model for ad-hoc retrieval and found that the results change little because of the very high correlation between the query performances obtained by the two retrieval models (0.96 measured by Pearson's coefficient).

4.2 Known Query Types

Suppose that query types are known. We treat each type of query separately and measure the correlation with average precision (or the reciprocal rank in the case of NP queries). We adopt the Pearson's correlation test which reflects the degree of linear relationship between the predicted and the actual retrieval performance.

4.2.1 *Content-based Queries*

Methods	Clarity	Robust	JSD	WIG	QF	WIG +QF
TB04+05 adhoc	**0.333**	**0.317**	**0.362**	**0.574**	**0.480**	**0.637**
TB06 adhoc	0.076	0.294	N/A	**0.464**	**0.422**	**0.511**

Table 4: Pearson's correlation coefficients for correlation with average precision on the Terabyte Tracks (ad-hoc) for clarity score, robustness score, the JSD-based method(we directly cites the score reported in [2]), WIG, query feedback(QF) and a linear combination of WIG and QF. Bold cases mean the results are statistically significant at the 0.01 level.

Table 4 shows the correlation with average precision on two data sets: one is a combination of TB04-adhoc and TB05-adhoc(100 topics in total) and the other is TB06-adhoc (50 topics). Our baselines are the clarity score (clarity) [6],the robustness score (robust)[1] and the JSD-based method (JSD) [2]. For the clarity and robustness score, we have tried different parameter settings and report the highest correlation coefficients we have found. We directly cite the result of the JSD-based method reported in [2]. The table also shows the results for the Weighted Information Gain (WIG) method and the Query Feedback (QF) method for predicting content-based queries. As we described in the previous section, both WIG and QF have one free parameter to set, that is, the cutoff rank K. We train the parameter on one dataset and test on the other. When combining WIG and QF, a simple linear combination is used and the combination weight is learned from the training data set.

From these results, we can see that our methods are considerably more accurate compared to the baselines. As an example, Figure 2 depicts the strength of the correlation with average precision for WIG on the first data set. It shows a clear linear relationship between the predictor and the actual retrieval performance. We also observe that further improvements are obtained from the combination of WIG and QF, suggesting that they measure different properties of the retrieval process that relate to performance.

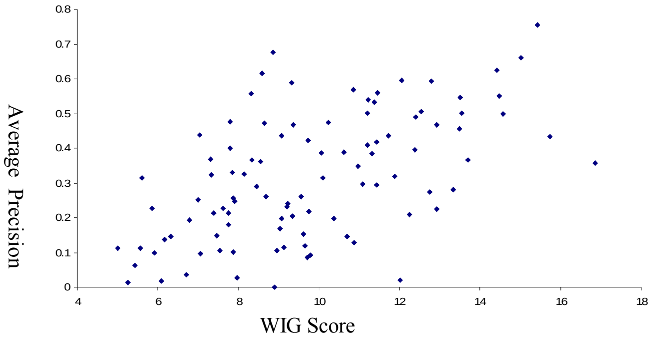

Figure 2: WIG score versus average precision for the 100 title ad-hoc queries from the Terabyte Tracks of 2004 and 2005

We also observe that our methods generalize well on TB06-adhoc while the correlation for the clarity score with retrieval performance on this data set is considerably worse. Further investigation shows that the mean average precision of TB06-ad-hoc is 0.342 and is about 10% better than that of the first data set. While the other three methods typically consider the top 100 or less documents given a ranked list, the clarity method usually needs the top 500 or more documents to adequately measure the coherence of a ranked list. Higher mean average precision makes ranked lists retrieved by different queries more similar in terms of coherence at the level of top 500 documents. We believe that this is the main reason for the low accuracy of the clarity score on the second data set.

Though this paper focuses on a Web search environment, it is desirable that our techniques will work consistently well in other situations. To this end, we examine the effectiveness of our techniques on the Robust 2004 Track. For our methods, we evenly divide all of the test queries into five groups and perform five-fold cross validation. Each time we use one group for training and the remaining four groups for testing. We make use of all of the queries for our two baselines, that is, the clarity score and the robustness score. The parameters for our baselines are the same as those used in [1].The results shown in Table 5 demonstrate that the prediction accuracy of our methods is on a par with that of the two strong baselines.

Clarity	Robust	WIG	QF
0.464	**0.539**	**0.468**	**0.464**

Table 5: Comparison of Pearson's correlation coefficients on the 2004 Robust Track for clarity score, robustness score, WIG and query feedback (QF). Bold cases mean the results are statistically significant at the 0.01 level.

Furthermore, we examine the prediction sensitivity of our methods to the cutoff rank K. With respect to WIG, it is quite robust to K on the Terabyte Tracks (2004-2006) while it prefers a small value of K like 5 on the 2004 Robust Track. In other words, a small value of K is a nearly-optimal choice for both kinds of tracks. Considering the fact that all other parameters involved in WIG are fixed and consequently the same for the two cases, this means WIG can achieve nearly-optimal prediction accuracy in two considerably different situations with exactly the same parameter setting. Regarding QF, it prefers a larger value of K such as 100 on the Terabyte Tracks and a smaller value of K such as 25 on the 2004 Robust Track.

4.2.2 NP Queries

We adopt WIG and first rank change (FRC) for predicting NP-query performance. We also try a linear combination of the two as in the previous section. The combination weight is obtained from the other data set. 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 6. Again, our baselines are the clarity score and the robustness score.

To make a fair comparison, we tune the clarity score in different ways. We found that using the first ranked document to build the query model yields the best prediction accuracy. We also attempted to utilize document structure by using the *mixture of language* models mentioned in section 3.1. Little improvement was obtained. The correlation coefficients for the clarity score reported in Table 6 are the best we have found. As we can see, our methods considerably outperform the clarity score technique on both of the runs. This confirms our intuition that the use of a coherence-based measure like the clarity score is inappropriate for NP queries.

Methods	Clarity	Robust.	WIG	FRC	WIG+FRC
TB05-NP	**0.150**	**-0.370**	**0.458**	**0.440**	**0.525**
TB06-NP	0.112	-0.160	**0.478**	**0.386**	**0.515**

Table 6: Pearson's correlation coefficients for correlation with reciprocal ranks on the Terabyte Tracks (NP) for clarity score, robustness score, WIG, the first rank change (FRC) and a linear combination of WIG and FRC. Bold cases mean the results are statistically significant at the 0.01 level.

Regarding the robustness score, we also tune the parameters and report the best we have found. 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 [1]. 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, the strength of the correlation is weaker and less consistent compared to our methods as shown in Table 6.

Based on the above analysis, we can see that current prediction techniques like clarity score and robustness score that are mainly designed for content-based queries face significant challenges and are inadequate to deal with NP queries. Our two techniques proposed for NP queries consistently demonstrate good prediction accuracy, displaying initial success in solving the problem of predicting performance for NP queries. Another point we want to stress is that the WIG method works well for both types of queries, a desirable property that most prediction techniques lack.

4.3 Unknown Query Types

In this section, we run two kinds of experiments without access to query type labels. First, we assume that only one type of query exists but the type is unknown. Second, we experiment on a mixture of content-based and NP queries. The following two subsections will report results for the two conditions respectively.

4.3.1 Only One Type exists

We assume that all queries are of the same type, that is, they are either NP queries or content-based queries. We choose WIG to deal with this case because it shows good prediction accuracy for both types of queries in the previous section. We consider two cases: (1) CB: all 150 title queries from the ad-hoc task of the Terabyte Tracks 2004-2006 (2)NP: all 433 NP queries from the named page finding task of the Terabyte Tracks 2005 and 2006.

We take a simple strategy by labeling all of the queries in each case as the same type (either NP or CB) regardless of their actual type. The computation of WIG will be based on the labeled query type instead of the actual type. There are four possibilities with respect to the relation between the actual type and the labeled type. The correlation with retrieval performance under the four possibilities is presented in Table 7. For example, the value 0.445 at the intersection between the second row and the third column shows the Pearson's correlation coefficient for correlation with average precision when the content-based queries are incorrectly labeled as the NP type.

	CB (labeled)	NP (labeled)
CB (actual)	**0.536**	**0.445**
NP (actual)	**0.174**	**0.467**

Table 7: Comparison of Pearson's correlation coefficients for correlation with retrieval performance under four possibilities on the Terabyte Tracks (NP). Bold cases mean the results are statistically significant at the 0.01 level.

Based on these results, we recommend treating all queries as the NP type when only one query type exists and accurate query classification is not feasible, considering the risk that a large loss of accuracy will occur if NP queries are incorrectly labeled as content-based queries. These results also demonstrate the strong adaptability of WIG to different query types.

4.3.2 A mixture of contented-based and NP queries

A mixture of the two types of queries is a more realistic situation that a Web search engine will meet. We evaluate prediction accuracy by how accurately poorly-performing queries can be identified by the prediction method assuming that actual query types are unknown (but we can predict query types). This is a challenging task because both the predicted and actual performance for one type of query can be incomparable to that for the other type.

Next we discuss how to implement our evaluation. We create a query pool which consists of all of the 150 ad-hoc title queries from Terabyte Track 2004-2006 and all of the 433 NP queries from Terabyte Track 2005&2006. We divide the queries in the pool into classes: "good" (better than 50% of the queries of the same type in terms of retrieval performance) and "bad" (otherwise). According to these standards, a NP query with the reciprocal rank above 0.2 or a content-based query with the average precision above 0.315 will be considered as good.

Then, each time we randomly select one query Q from the pool with probability p that Q is contented-based. The remaining queries are used as training data. We first decide the type of query Q according to a query classifier. Namely, the query classifier tells us whether query Q is NP or content-based. Based on the predicted query type and the score computed for query Q by a prediction technique, a binary decision is made about whether query Q is good or bad by comparing to the score threshold of the predicted query type obtained from the training data. Prediction accuracy is measured by the accuracy of the binary decision. In our implementation, we repeatedly take a test query from the query pool and prediction accuracy is computed as the percentage of correct decisions, that is, a good(bad) query is predicted to be good (bad). It is obvious that random guessing will lead to 50% accuracy.

Let us take the WIG method for example to illustrate the process. Two WIG thresholds (one for NP queries and the other for content-based queries) are trained by maximizing the prediction accuracy on the training data. When a test query is labeled as the NP (CB) type by the query type classifier, it will be predicted to be good if and only if the WIG score for this query is above the NP (CB) threshold. Similar procedures will be taken for other prediction techniques.

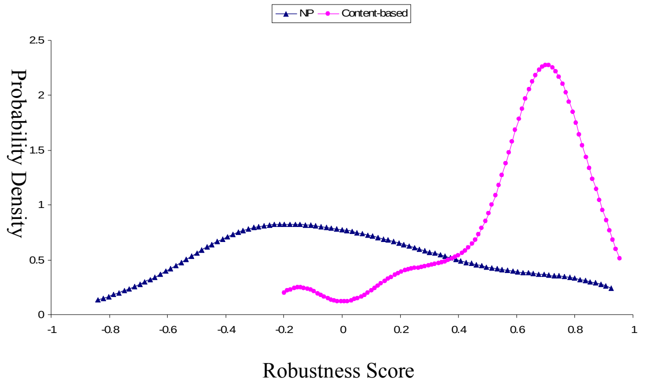

Figure 3: Distribution of robustness scores for NP and CB queries. The NP queries are the 252 NP topics from the 2005 Terabyte Track. The content-based queries are the 150 ad-hoc title from the Terabyte Tracks 2004-2006. The probability distributions are estimated by the Kernel density estimation method.

Now we briefly introduce the automatic query type classifier used in this paper. We find that the robustness score, though originally proposed for performance prediction, is a good indicator of query types. We find that on average content-based queries have a much higher robustness score than NP queries. For example, Figure 3 shows the distributions of robustness scores for NP and content-based queries. According to this finding, 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.

Strategies	Robust	WIG-1	WIG-2	WIG-3	Optimal
p=0.6	0.565	0.624	0.665	0.684	0.701
P=0.4	0.567	0.633	0.654	0.673	0.696

Table 8: Comparison of prediction accuracy for five strategies in the mixed-query situation. Two ways to sample a query from the pool: (1) the sampled query is content-based with the probability p=0.6. (that is, the query is NP with probability 0.4) (2) set the probability p=0.4.

We consider five strategies in our experiments. In the first strategy (denoted by "*robust*"), we use the robustness score for query performance prediction with the help of a perfect query classifier that always correctly map a query into one of the two categories (that is, NP or CB). This strategy represents the level of prediction accuracy that current prediction techniques can achieve in an ideal condition that query types are known. In the next following three strategies, the WIG method is adopted for performance prediction. The difference among the three is that three different query classifiers are used for each strategy: (1) the classifier always classifies a query into the NP type. (2) the classifier is the robust score classifier mentioned above. (3) the classifier is a perfect one.

These three strategies are denoted by *WIG-1, WIG-2* and *WIG-3* respectively. The reason we are interested in WIG-1 is based on the results from section 4.3.1. In the last strategy (denoted by "*Optimal*") which serves as an upper bound on how well we can do so far, we fully make use of our prediction techniques for each query type assuming a perfect query classifier is available. Specifically, we linearly combine WIG and QF for content-based queries and WIG and FRC for NP queries.

The results for the five strategies are shown in Table 8. For each strategy, we try two ways to sample a query from the pool: (1) the sampled query is CB with probability p=0.6. (the query is NP with probability 0.4) (2) set the probability p=0.4. From Table 8 We can see that in terms of prediction accuracy WIG-2 (the WIG method with the automatic query classifier) is not only better than the first two cases, but also is close to WIG-3 where a perfect classifier is assumed. Some further improvements over WIG-3 are observed when combined with other prediction techniques. The merit of WIG-2 is that it provides a practical solution to automatically identifying poorly performing queries in a Web search environment with mixed query types, which poses considerable obstacles to traditional prediction techniques.

5. CONCLUSIONS AND FUTURE WORK

To our knowledge, our paper is the first to thoroughly explore prediction of query performance in web search environments. We demonstrated that our models resulted in higher prediction accuracy than previously published techniques not specially devised for web search scenarios. In this paper, we focus on two types of queries in web search: content-based and Named-Page (NP) finding queries, corresponding to the ad-hoc retrieval task and the Named-Page finding task respectively.

For content-based queries, our two models (WIG and query feedback) are substantially more accurate than the current state-of-the-art techniques that do not scale well to large web collections. Furthermore, we tested the robustness of the two models by conducting experiments on a traditional non-Web TREC collection where the current state-of-the-art techniques are known to be effective. Results showed that the two models resulted in consistent prediction accuracy comparable to that obtained by the state-of-the-art techniques.

Regarding NP queries, our two models (WIG and the first rank change) clearly outperformed the baselines originally designed for content-based queries even though we carefully tuned the parameters of the baselines.

We considered a more realistic case that no prior information on query types is available. We first assumed that there exists only one type of query. Our experiments showed that good prediction accuracy can be obtained by WIG if we treat all queries as the NP type regardless of their actual type. We then considered a much more challenging situation that a mixture of query types exists. We showed that good prediction accuracy can be obtained by using WIG with the help of an effective query type classifier found by us. By this strategy, we provide a practical solution to performance prediction in real-world web environments which is largely beyond the capabilities of current techniques.

Considering the adaptability of WIG to a range of collections and query types, our future plan is to apply this method to predict user preference of search results on realistic data collected from a commercial search engine.

6. ACKNOWLEGMENTS

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 author and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] Y. Zhou ,W. B. Croft ,Ranking Robustness: A Novel Framework to Predict Query Performance, in Proceedings of CIKM 2006.

[2] D.Carmel, E.Yom-Tov, A.Darlow,D.Pelleg, What Makes a Query Difficult?, in Proceedings of SIGIR 2006.

[3] C.L.A. Clarke, F. Scholer, I.Soboroff, The TREC 2005 Terabyte Track, In the Online Proceedings of 2005 TREC.

[4] B. He and I.Ounis. Inferring query performance using pre-retrieval predictors. In proceedings of the SPIRE 2004.

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

[6] S. Cronen-Townsend, Y. Zhou, W. B. Croft, Predicting Query Performance, in Proceedings of SIGIR 2002.

[7] V.Vinay, I.J.Cox, N.Mill-Frayling,K.Wood, On Ranking the Effectiveness of Searcher, in Proceedings of SIGIR 2006.

[8] D.Metzler, W.B.Croft, A Markov Random Filed Model for Term Dependencies, in Proceedings of SIGIR 2005.

[9] D.Metzler, T.Strohman,Y.Zhou,W.B.Croft, Indri at TREC 2005: Terabyte Track, In the Online Proceedings of 2004 TREC.

[10] P. Ogilvie and J. Callan, Combining document representations for known-item search, in Proceedings of SIGIR 2003.

[11] A.Berger, J.Lafferty, Information retrieval as statistical translation, in Proceedings of SIGIR 1999.

[12] Indri search engine : http://www.lemurproject.org/indri/

[13] I.J. Taneja: On Generalized Information Measures and Their Applications, Advances in Electronics and Electron Physics, Academic Press (USA), 76, 1989, 327-413.

[14] S.Cronen-Townsend, Y. Zhou and Croft, W. B. , "A Framework for Selective Query Expansion," in Proceedings of CIKM 2004.

[15] F.Song, W.B.Croft, A general language model for information retrieval, in Proceedings of SIGIR 1999.

[16] Personal email contact with Vishwa Vinay and our own experiments

[17] E.Yom-Tov, S.Fine, D.Carmel, A.Darlow, Learning to Estimate Query Difficulty Including Applications to Missing

Content Detection and Distributed Information retrieval, in Proceedings of SIGIR 2005