Predicting Query Performance by Query-Drift Estimation

ANNA SHTOK and OREN KURLAND, Technion – Israel Institute of Technology
DAVID CARMEL, IBM Haifa Research Labs
FIANA RAIBER and GAD MARKOVITS, Technion – Israel Institute of Technology

Predicting *query performance*, that is, the effectiveness of a search performed in response to a query, is a highly important and challenging problem. We present a novel approach to this task that is based on measuring the standard deviation of retrieval scores in the result list of the documents most highly ranked. We argue that for retrieval methods that are based on document-query surface-level similarities, the standard deviation can serve as a surrogate for estimating the presumed amount of *query drift* in the result list, that is, the presence (and dominance) of aspects or topics not related to the query in documents in the list. Empirical evaluation demonstrates the prediction effectiveness of our approach for several retrieval models. Specifically, the prediction quality often transcends that of current state-of-the-art prediction methods.

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

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Query-performance prediction, query drift, score distribution

ACM Reference Format:
Shtok, A., Kurland, O., Carmel, D., Raiber, F., and Markovits, G. 2012. Predicting query performance by query-drift estimation. ACM Trans. Inf. Syst. 30, 2, Article 11 (May 2012), 35 pages.
DOI = 10.1145/2180868.2180873 http://doi.acm.org/10.1145/2180868.2180873

1. INTRODUCTION

Many Information Retrieval (IR) systems suffer a radical variance in performance when responding to users' queries. Even for systems that succeed very well on average, the quality of results returned for some of the queries is poor [Harman and Buckley 2004; Voorhees 2004]. Thus, it is desirable that IR systems will be able to identify "difficult" queries in order to handle them properly.

Previous work has focused on identifying features that can serve as indicators for performance quality. These features can be classified into two main categories: preretrieval and postretrieval features. Preretrieval features [Hauff et al. 2008a; He and Ounis 2004; Mothe and Tanguy 2005; Scholer et al. 2004; Zhao et al. 2008] are extracted directly from the query expression, using either linguistic [Mothe and Tanguy

Portions of the work reported here were previously presented in Shtok et al. [2009].
The article is based on work supported in part by the Israel Science Foundation under grant no. 557/09, by Google's, IBM's, and Yahoo!'s faculty research awards, by IBM's SUR award, and by IBM's Ph.D. fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsoring institutions.
Authors' addresses: A. Shtok, O. Kurland (corresponding author), F. Raiber, and G. Markovits, Faculty of Industrial Engineering and Management, Technion – Israel Institute of Technology, Haifa 32000, Israel; email: kurland@ie.technion.ac.il; D. Carmel, IBM Research – Haifa Labs, Haifa 31905, Israel.

DOI 10.1145/2180868.2180873 http://doi.acm.org/10.1145/2180868.2180873

2005] or statistical [Hauff et al. 2008a; He and Ounis 2004; Scholer et al. 2004; Zhao et al. 2008] analysis of query terms. Postretrieval approaches analyze the *result list*: the list of documents most highly ranked in response to the query. The *query clarity* method [Cronen-Townsend et al. 2002], for example, measures the divergence of a language model induced from the result list from that induced from the corpus.

We present a novel approach to query performance prediction. Our approach is based on measuring the standard deviation of retrieval scores in the result list. We argue that for retrieval methods based on surface-level similarities between documents and the query, the standard deviation can serve as a surrogate in estimating the presumed amount of *query drift* in the result list, that is, the presence and dominance of nonquery-related aspects or topics manifested in documents in the list.

To substantiate our argument, we leverage insights from work on pseudo-feedback-based query expansion in which query drift plays an important role [Mitra et al. 1998]. One such insight is that a *centroid* representation of the result list, that is, a centroid of the models of documents in the list (e.g., the centroid of the tf.idf vectors that represent the documents [Salton et al. 1975]), which often serves as a basis for an expanded query form, manifests query drift [Abdul-Jaleel et al. 2004; Harman 1992; Zhai and Lafferty 2001a]. Thus, for retrieval methods for which retrieval scores reflect document-query similarity, the retrieval score of a centroid represents insufficient *query commitment*, that is, insufficient emphasis of query terms. This retrieval score can thereby serve as a reference comparison for estimating query drift.

We show that for several retrieval methods, a centroid need not even be computed because the *mean* retrieval score of documents in the result list corresponds to the retrieval score of some centroid. Consequently, the mean can serve as a reference comparison in query-drift estimation. Accordingly, we argue that high divergence of retrieval scores from this reference comparison (as measured, for example, by the standard deviation) correlates with low levels of query drift and, therefore, with improved retrieval effectiveness.

To evaluate the quality of query performance prediction, we use evaluation paradigms that address, among others, the variability of query performance prediction quality with respect to values of free parameters incorporated by the predictors. Evaluation performed using several TREC collections attests to the high-quality query performance prediction of our predictor for several retrieval models. Specifically, the prediction quality of our approach often transcends that of state-of-the-art—preretrieval and postretrieval predictors.

Through an additional exploration we study factors that affect prediction quality (e.g., the size of the result list). Furthermore, we demonstrate the merits of using standard deviation for measuring retrieval scores dispersion as a means for predicting query performance through a comparison with an alternative approach for quantifying dispersion.

2. RELATED WORK

Predicting query performance is a significant challenge due to the numerous factors that impact retrieval performance. Preretrieval prediction methods aim to identify characteristics of query difficulty by analyzing the query expression [Hauff et al. 2008a; He and Ounis 2004; Mothe and Tanguy 2005; Scholer et al. 2004; Zhao et al. 2008]. For instance, linguistic features [Mothe and Tanguy 2005] and statistical properties of query terms (e.g., IDF values) [He and Ounis 2004; Scholer et al. 2004; Zhao et al. 2008] were suggested as query performance predictors. Among these predictors there are some [Zhao et al. 2008] that rely on computing the variance of a query term's TF.IDF value across all documents in the corpus. This is conceptually

reminiscent of our *postretrieval* prediction approach that is based on the standard deviation of retrieval scores of the documents most highly ranked in response to the query. In Section 4 we further discuss the connection between, and compare the prediction quality of, the predictors. Furthermore, we present an in-depth prediction quality comparison of our prediction approach with a suite of preretrieval prediction methods.

The (short) query alone is often not expressive enough for reliable prediction [Hauff et al. 2008a] as we will demonstrate using several experimental settings in Section 4. Thus, many effective prediction approaches, as the prediction method we propose here, employ postretrieval analysis of the *result list*, that is, the list of documents most highly ranked in response to the query. In what follows we discuss three such prominent paradigms.[1]

Clarity. The clarity approach [Cronen-Townsend et al. 2002] is based on measuring the "focus" (clarity) of the result list with respect to the corpus. The conjecture is that a language model induced from the result list for an "easy" query will be distinct (e.g., in terms of KL divergence [Cronen-Townsend et al. 2002]) from that induced from the entire corpus. Consequently, different forms of clarity estimation were proposed. For example, measuring the divergence between query terms' frequency in the result list and that in the entire corpus was suggested in the *divergence from randomness framework* [Amati et al. 2004]. The Jensen-Shannon divergence between models of the result list, the query, and that of the entire corpus was also shown to indicate query performance [Carmel et al. 2006]. Using a weighted KL-divergence measure [Cronen-Townsend et al. 2004; Hauff et al. 2008b], and/or utilizing only documents that contain all query terms [Hauff et al. 2008b], were proposed for improving clarity estimation. In Section 4 we show that the prediction quality of our suggested predictor often transcends that of the clarity measure [Cronen-Townsend et al. 2002] and is also better than that of an improved clarity measure [Hauff et al. 2008b, 2010] for a noisy, large-scale Web collection (namely, ClueWeb).

Robustness. Another effective performance prediction paradigm is based on estimating different notions of the robustness of the result list. The hypothesis is that the more robust the result list is, the less "difficult" the query is. Robustness with respect to query perturbations was measured by the overlap between the list retrieved in response to the entire query and those retrieved in response to individual query terms [Yom-Tov et al. 2005]. The effect of document perturbations on the resultant retrieved list is another form of robustness estimation [Vinay et al. 2006; Zhou and Croft 2006]. Measuring the agreement between lists retrieved in response to the query using various retrieval functions was also suggested as a robustness estimate [Aslam and Pavlu 2007]. The *Query Feedback* (QF) prediction method is based on comparing the original result list with that induced by using a query generated from this list [Zhou and Croft 2007]. The idea is that the more similar the result lists are, the less query-related "noise" there is in the corpus, and, hence, the more effective the retrieval is presumed to be. We use this approach as a reference comparison in the evaluation presented in Section 4.

Our proposed query performance predictor, which measures the diversity of retrieval scores in the result list, can be thought of as a surrogate for estimating

[1]For a comprehensive recent survey of methods for query performance prediction see Carmel and Yom-Tov [2010].

robustness with respect to document perturbations [Vinay et al. 2006; Zhou and Croft 2006]. We further discuss this point in Section 3.2.

The cohesion of the result list (another aspect of result list robustness) was shown to indicate query performance [Vinay et al. 2006]. In contrast, our proposed predictor focuses on the way documents are related to the query (as measured by their retrieval scores), rather than to each other.

Analyzing retrieval scores. In many retrieval models, the similarity of documents to a query is reflected by their retrieval scores. Hence, the distribution of retrieval scores can potentially help to predict query performance, as we show in this article. For example, the highest retrieval score was shown to be a relatively successful query performance predictor [Tomlinson 2004]. The difference between retrieval scores produced in a query-independent manner and those produced in a query-dependent way, which reflects the "discriminative power" of the query, was also shown to be an indicator for query performance [Bernstein et al. 2005]. Query performance was also demonstrated to be correlated with the extent to which the result list "respects" the *cluster hypothesis* [Diaz 2007]; that is, the extent to which similar documents receive similar retrieval scores. This form of score distribution analysis is complementary to ours, and we leave their integration for future work.

A recently proposed state-of-the-art predictor is the *Weighted Information Gain* (WIG) measure [Zhou and Croft 2007]. WIG essentially measures the divergence between the mean retrieval score of top-ranked documents and that of the entire corpus. The hypothesis is that the more similar these documents are to the query, with respect to the query similarity exhibited by a general nonrelevant document (i.e., the corpus), the more effective the retrieval is. In contrast, as described in Section 3, our predictor computes the divergence between the retrieval scores of top-ranked documents and that of a pseudo nonrelevant document that exhibits relatively high query similarity. We show in Section 4 that the performance prediction quality of our predictor often transcends that of WIG. However, the integration of the two predictors is an interesting venue for future work.

Very recently, we have proposed a query performance prediction framework that is based on statistical decision theory [Shtok et al. 2010]. Both the query performance predictor that we propose in this article and the predictors that serve for reference comparisons in the evaluation study, presented in Section 4, were shown to be quite effective when incorporated in the framework; specifically, in estimating the extent to which a relevance model constructed from a language-model-based result list represents the information need underlying the query.

We use the standard deviation of retrieval scores in the list, which is a simple statistics of the retrieval scores distribution, for predicting query performance. There is a large body of work on retrieval scores distribution fitting [Arampatzis and Robertson 2011; Arampatzis et al. 2009; Dai et al. 2011; Kanoulas et al. 2010; Manmatha et al. 2001; Robertson 2007]; specifically, for tasks such as normalizing retrieval scores for fusion (metasearch) [Manmatha et al. 2001] and for truncating retrieved lists [Arampatzis et al. 2009]. Applying such methods to the query performance prediction task is an interesting future venue.

Following the introduction of the standard deviation of retrieval scores as a basis for query performance prediction [Pérez-Iglesias and Araujo 2009; Shtok et al. 2009], there has been some work on automatically setting the result list size, at which the deviation is computed, on a per-query basis, and on normalizing the deviation [Cummins et al. 2011a; Pérez-Iglesias and Araujo 2010]. While these approaches were shown to improve prediction quality, we focus here on the initially introduced predictor [Shtok et al. 2009] that uses a fixed list size for all queries and a basic corpus-based

normalization. We note that the result list size is set for all postretrieval predictors examined in this article, either by optimization of prediction quality as measured over all queries per collection, or by using cross-validation. (Section 4 discusses the technical details.) It is important to note that our goal is not to optimize our suggested predictor in the best possible way, but rather study the factors that affect its quality, analyze the relative prediction quality of the components it is composed of, and gain a better understanding of the settings in which it is effective. Moreover, this practice, which results in conservative estimates of the prediction quality of our predictor, enables a fair comparison with other postretrieval predictors that use a fixed result list size for all queries as well; these predictors could also potentially be improved by a per-query result-list size setting, but this is out of the scope of this article. Finally, we note that, as is, the predictor we present is shown in Section 4 to have prediction quality that favorably compares with that of state-of-the-art predictors.

3. PREDICTION FRAMEWORK

We assume that the following have been fixed: a query q, a corpus of documents $\mathcal{D}$, and a retrieval method $\mathcal{M}$ that is used to rank the documents in $\mathcal{D}$ in response to q. We use $Score(d)$ to denote the retrieval score assigned by $\mathcal{M}$ to document d in response to q. Our goal is to devise an estimate (predictor) for the effectiveness of the ranking induced by $\mathcal{M}$ in the absence of relevance judgment information. The estimated effectiveness is the query performance we attribute to $\mathcal{M}$ with respect to q. To that end, the methods we present utilize the result list $\mathcal{D}_q^{[k]}$ of the k documents that are the most highly ranked; k is a free parameter that is fixed to some value prior to retrieval (and prediction) time.

Documents in $\mathcal{D}_q^{[k]}$ are presumed by $\mathcal{M}$ to be the most relevant to the information need expressed by q among all documents in the corpus. For many retrieval models (e.g., vector space model [Salton et al. 1975], probabilistic models [Fuhr 1992], the inference network model [Turtle and Croft 1990], and the language model framework [Ponte and Croft 1998; Croft and Lafferty 2003]), this assumption is based on the fact that the documents in $\mathcal{D}_q^{[k]}$ exhibit the highest surface-level similarity to q. Hereinafter, we assume that $\mathcal{M}$ is one such retrieval method; we also assume that $\mathcal{D}_q^{[k]}$ is composed of documents that contain at least one query term (i.e., exhibit a nonzero surface-level query similarity).

3.1. Empirical Observation

As noted before, previous work has demonstrated some connections between the characteristics of retrieval scores in $\mathcal{D}_q^{[k]}$ and query performance. In Section 4 we present the following novel finding about one such connection.

> The larger the standard deviation of retrieval scores in $\mathcal{D}_q^{[k]}$ (modulo some technical details), the better the query performance of $\mathcal{M}$ with respect to q.

This observation, as we show in Section 4, holds in a consistent manner for several retrieval methods, and across various TREC datasets. Our goal in what follows is to provide a potential explanation to why this observation holds.

3.2. Preliminary Discussion

At first glance, the observation just stated might seem to be contradicting the result-list cohesion principle [Vinay et al. 2006]. That is, the more similar documents in

$\mathcal{D}_q^{[k]}$ are to each other, the better query performance should be. Closer examination, however, reveals that there is no real contradiction: the standard deviation of retrieval scores reflects the connection between documents in $\mathcal{D}_q^{[k]}$ and the query, rather than between the documents themselves.

A potential explanation to the effectiveness of the standard deviation of retrieval scores as a query performance predictor can be drawn based on result-list robustness arguments. It was argued, and empirically shown, that a result list that is robust, that is, does not change much both in terms of the documents that compose it and in their relative ranking, indicates good query performance. Robustness can be measured with respect to perturbations of documents [Vinay et al. 2006; Zhou and Croft 2006], perturbations of the query and/or the query model [Yom-Tov et al. 2005; Zhou and Croft 2007], and changes in the retrieval function applied [Aslam and Pavlu 2007].

Here, we consider document perturbations [Vinay et al. 2006; Zhou and Croft 2006]. Perturbations of documents that do not substantially affect the frequency of query terms in documents are unlikely to significantly change retrieval scores. Thus, if the retrieval scores in the result list $\mathcal{D}_q^{[k]}$ are quite spread, as measured, for example, by their standard deviation, then these perturbations are unlikely to result in significant changes to $\mathcal{D}_q^{[k]}$. Consequently, $\mathcal{D}_q^{[k]}$ could be considered robust with respect to document perturbations.

Along the same line, recent work [Pérez-Iglesias and Araujo 2009] that was published at the same time the conference version of this article was published [Shtok et al. 2009] presented a list-robustness-based explanation for the effectiveness of standard deviation of retrieval scores as a query performance predictor.

> "If a ranking list has a high value of dispersion among the document scores, it could be a sign that the ranking function has been able to discriminate between relevant and not relevant documents. On the other hand if a low level of dispersion appears, because the ranking function has assigned similar weights, it can be interpreted as it was not able to distinguish between relevant and not relevant documents."

However, an important property of the standard deviation as a means for measuring dispersion is not accounted for in the list-robustness arguments discussed earlier, that is, the fact that dispersion is measured with respect to the mean retrieval score. Indeed, in Section 4.2.4 we demonstrate the potential importance of this specific way of measuring retrieval scores dispersion. We do so by showing that while measuring dispersion by the distance of the retrieval scores function from a constant function yields effective query performance prediction, prediction effectiveness is still consistently lower than that attained by using the standard deviation.

In the rest of this section we present an informal discussion that argues for the potential connection between the standard deviation of retrieval scores, specifically, the importance of using the mean retrieval score for score dispersion computation, and query performance.

3.3. The Standard Deviation of Retrieval Scores as a Query Performance Predictor

It is often the case that there are documents not pertaining to the information need expressed by q that exhibit high similarity to q. Hence, these documents can be in $\mathcal{D}_q^{[k]}$: the list of most highly ranked documents in response to q. Significant presence of such documents in $\mathcal{D}_q^{[k]}$ results in degraded retrieval performance. We refer to these documents as *misleaders* because they "mislead" the retrieval method into "believing" that

they are relevant as they exhibit relatively high query similarity. Usually, misleaders are deemed nonrelevant because they mainly discuss aspects or topics not related (or only partially related) to the information need expressed by the query, and these aspects dominate the (incidental) occurrence of query terms [Buckley 2004]. Thus, misleaders can be thought of as expressing *query drift* [Buckley et al. 1994], since they represent aspects that "drift away" from those represented by the query. In other words, if we were to distill a query from a misleader, the information need expressed by the distilled query would have drifted away from that represented by the original query q, because the misleader is not relevant to q.

Thus, it seems we should opt to devise a measure that quantifies the presumed amount of query drift in $\mathcal{D}_q^{[k]}$, that is, presence and dominance of nonrelated query aspects. More specifically, we should potentially estimate how many documents in $\mathcal{D}_q^{[k]}$ abstain from exhibiting query drift, and to what extent, as these are less likely to be misleaders. While doing so might seem to be replacing the challenge of assessing relevance with that of assessing query drift, it turns out that there is a potential way of estimating presumed query drift. In fact, we argue in the next section that the standard deviation of retrieval scores in $\mathcal{D}_q^{[k]}$ can be thought of as a proxy for the amount of query drift (or more precisely, lack thereof) in $\mathcal{D}_q^{[k]}$. To that end, we leverage insights gained in work on pseudo-feedback-based retrieval [Buckley et al. 1994; Mitra et al. 1998].

3.3.1. Estimating Query Drift. Query drift is a recurring issue in pseudo-feedback-based query expansion retrieval methods [Buckley et al. 1994]. Such methods are usually based on constructing a query model using the documents in $\mathcal{D}_q^{[k]}$; the query model is then used to rank the entire corpus [Buckley et al. 1994]. Often, the constructed model is based on a *centroid* representation, $Cent(\mathcal{D}_q^{[k]})$, of the list $\mathcal{D}_q^{[k]}$. For example, in Rocchio's model [Rocchio 1971], which operates in the vector space, and in the *relevance model* [Lavrenko and Croft 2001], which operates in the language model simplex, $Cent(\mathcal{D}_q^{[k]})$ is the (weighted) mean of the models of documents in $\mathcal{D}_q^{[k]}$: $\frac{1}{k}\sum_{d\in\mathcal{D}_q^{[k]}} w(d)M_d$; M_d is d's model, and $w(d)$ is an importance weight associated with d. However, it is a well-established empirical fact that using only a centroid of $\mathcal{D}_q^{[k]}$ (e.g., as the one just mentioned) can yield poor retrieval performance for many retrieval methods [Abdul-Jaleel et al. 2004; Harman 1992; Rocchio 1971; Zhai and Lafferty 2001a]. We note that this observation was shown to hold even in some cases wherein all documents in $\mathcal{D}_q^{[k]}$ are relevant to q [Harman 1992; Raiber and Kurland 2010; Terra and Warren 2005].

Thus, the centroid is often anchored to the original query q using interpolation [Abdul-Jaleel et al. 2004; Harman 1992; Zhai and Lafferty 2001a]; this is the case whether using pseudo feedback or true feedback (i.e., only the relevant document) for query expansion. In Rocchio's model, and in the relevance model approach, this translates to using $\alpha M_q + \beta Cent(\mathcal{D}_q^{[k]})$ for retrieval; M_q is q's original model (representation) and α and β are free parameters[2]. The resultant retrieval performance is indeed significantly improved whether using pseudo or true feedback. Consequently, researchers concluded that the centroid often manifests query drift, that is, it can be, and often is, dominated by aspects not related to the query [Harman 1992; Mitra et al. 1998].

[2]The relevance model that uses only the centroid is often termed "RM1", while the one that uses interpolation with a model of the original query is termed "RM3" [Abdul-Jaleel et al. 2004].

Given the observations just stated, $Cent(\mathcal{D}_q^{[k]})$ could be viewed as a prototypical misleader. It attributes (some) importance to the query terms by virtue of the way it is constructed (from documents in $\mathcal{D}_q^{[k]}$); however, this importance is dominated by nonquery-related aspects that can lead to query drift. The importance attributed to q's terms by $Cent(\mathcal{D}_q^{[k]})$ can be measured, for example, by its retrieval score with respect to q, $\mu \stackrel{def}{=} Score(Cent(\mathcal{D}_q^{[k]}))$. In fact, we need not even directly compute μ. As it turns out, the mean retrieval score of documents in $\mathcal{D}_q^{[k]}$,

$$\hat{\mu} \stackrel{def}{=} \frac{1}{k} \sum_{d \in \mathcal{D}_q^{[k]}} Score(d),$$

corresponds in several retrieval methods to the retrieval score, μ, of some centroid-based representation of $\mathcal{D}_q^{[k]}$. (We show that in Section 3.5.) Thus, $\hat{\mu}$ can serve as a reference comparison for query-drift estimation, as it represents the insufficient *Query Commitment* (QC) of a prototypical misleader; that is, the insufficient emphasis of query terms with respect to non-query-related terms.

3.3.2. Estimates of Retrieval Effectiveness. In light of the preceding, one way of predicting the effectiveness of the results retrieved by $\mathcal{M}$ is to estimate the potential volume of misleaders in $\mathcal{D}_q^{[k]}$. The smaller this volume, the higher our estimate of effectiveness should be.

Looking for query committed documents. The first estimate we explore considers documents that are potentially query committed to a good extent; that is, that are less likely to express query drift. Following the previous discussion, we note that predicting whether document d in $\mathcal{D}_q^{[k]}$ is sufficiently query committed, or conversely, a misleader, can be based to a certain extent on the following idea. We need to assess whether the aspects pertaining to q that are manifested in d dominate the aspects that do not pertain to q. Now, given that $\hat{\mu}$ represents the insufficient query commitment of a prototypical misleader (the centroid), the following principle emerges. The higher d's retrieval score is with respect to $\hat{\mu}$, the less likely d is to be dominated by nonquery-related aspects and therefore the lower its chances of exhibiting query drift and being a misleader. Naturally, this principle applies only to documents with retrieval scores higher than $\hat{\mu}$. It is important to note that the conceptual premise underlying this principle is that substantial occurrence of query terms in a document—which is measured here with respect to that in the centroid—is not likely to be incidental, and hence, can potentially imply to a significant presence of query-related aspects. This premise echoes some of the underlying assumptions made in axiomatic approaches to retrieval [Fang and Zhai 2005].[3]

One way to quantify the principle just stated is by computing the overall divergence of retrieval scores higher than $\hat{\mu}$ from $\hat{\mu}$. However, retrieval scores are query dependent and consequently, so might be the divergence. Thus, to ensure inter-query compatibility, we normalize the divergence with respect to the retrieval score of a general prototypical nonrelevant document, namely, the corpus. (We assume that the corpus can be represented as a single pseudo document, for example, by using a centroid

[3]In adversarial retrieval settings, wherein keyword stuffing can take place, for example, this premise naturally does not hold. However, this is out of the scope of this article.

representation.) The resultant Normalized Query Commitment (NQC) estimate, which is based on documents with presumably positive ("+") query commitment, is[4]

$$NQC_+(q, \mathcal{M}) \stackrel{def}{=} \frac{\sqrt{\frac{1}{k}\sum_{d\in\mathcal{D}_q^{[k]}:Score(d)>\hat{\mu}}(Score(d)-\hat{\mu})^2}}{|Score(\mathcal{D})|}. \quad (1)$$

We use the absolute value of the corpus retrieval scores so as to address cases wherein retrieval scores are negative. This is the case when using logs of probabilities of relevance, as in the language modeling framework, for example, that we use in Section 3.5.

Misleader population. The second estimate that we consider addresses the potential size of the misleader population. Specifically, let $\mathcal{R}$ and $\mathcal{N}$ be the sets of relevant and nonrelevant documents in the corpus, respectively. We use $\mathcal{R}_q$ and $\mathcal{N}_q$ to denote the sets of relevant and nonrelevant documents that exhibit a "reasonable" surface-level query similarity, respectively. By definition, $\mathcal{D}_q^{[k]} \subset \mathcal{R}_q \cup \mathcal{N}_q$, as $\mathcal{D}_q^{[k]}$ is the list of documents most similar to the query; also, note that misleaders in $\mathcal{D}_q^{[k]}$ come from $\mathcal{N}_q$. Now, if $\mathcal{N}_q$ is an empty set, then the retrieval is highly effective, since $\mathcal{D}_q^{[k]}$ is populated only with documents from $\mathcal{R}_q$, and these are relevant. On the other hand, the more documents there are in $\mathcal{N}_q$—the source for misleaders—the more chances there are for misleaders to populate $\mathcal{D}_q^{[k]}$, and, consequently, to degrade retrieval effectiveness. Thus, we hypothesize that $|\mathcal{N}_q|$ correlates with retrieval effectiveness. Since it is a hard task to estimate $|\mathcal{N}_q|$ in the absence of relevance judgments, we make the assumption that the smaller $|\mathcal{R}_q \cup \mathcal{N}_q|$, the higher the chances that $\mathcal{N}_q$ is of a relatively small size. This assumption echoes the "pigeonhole principle": given a fixed-size set $\mathcal{R}_q$, the smaller $\mathcal{R}_q \cup \mathcal{N}_q$, the smaller $\mathcal{N}_q$ is. Or put in other words, given that the result list size, k, is fixed, and that the number of relevant documents exhibiting "reasonable" query similarity is also fixed, the less nonrelevant documents there are in the corpus that exhibit "reasonable" query similarity, the more documents from $\mathcal{R}_q$ are likely to populate $\mathcal{D}_q^{[k]}$, and hence, retrieval performance increases.

To summarize the arguments presented before, the smaller the number of documents in the corpus that exhibit "reasonable" query similarity, the less likely that there is a large population of potential misleaders, and the less are the chances that these "dominate" the result list.

Naturally, the challenge is to quantify "reasonable" query similarity. To that end, we use $\hat{\mu}$, which represents insufficient query commitment, as a reference comparison. Specifically, if the retrieval scores that are lower than $\hat{\mu}$ are lower by a substantial amount (e.g., close to zero in the extreme case), then potentially, the overall number of documents with reasonable query similarity ($|\mathcal{R}_q \cup \mathcal{N}_q|$) is small. In such a case, the potential of misleaders entering $\mathcal{D}_q^{[k]}$ declines, as stated earlier. Conversely, a scenario where $\hat{\mu}$ results from having many documents with retrieval scores slightly higher and slightly lower than $\hat{\mu}$ implies an increased $|\mathcal{R}_q \cup \mathcal{N}_q|$ and consequently an increased potential for misleaders entering $\mathcal{D}_q^{[k]}$. Thus, we hypothesize that retrieval effectiveness is correlated with increased (normalized) negative ("-") query commitment of

[4]Recall that k should be set to the same value for all queries prior to prediction (retrieval) time. But, for a query for which the number of documents containing at least one query term is less than that determined a priori, we set k to this number; hence, the need for normalization by $\frac{1}{k}$.

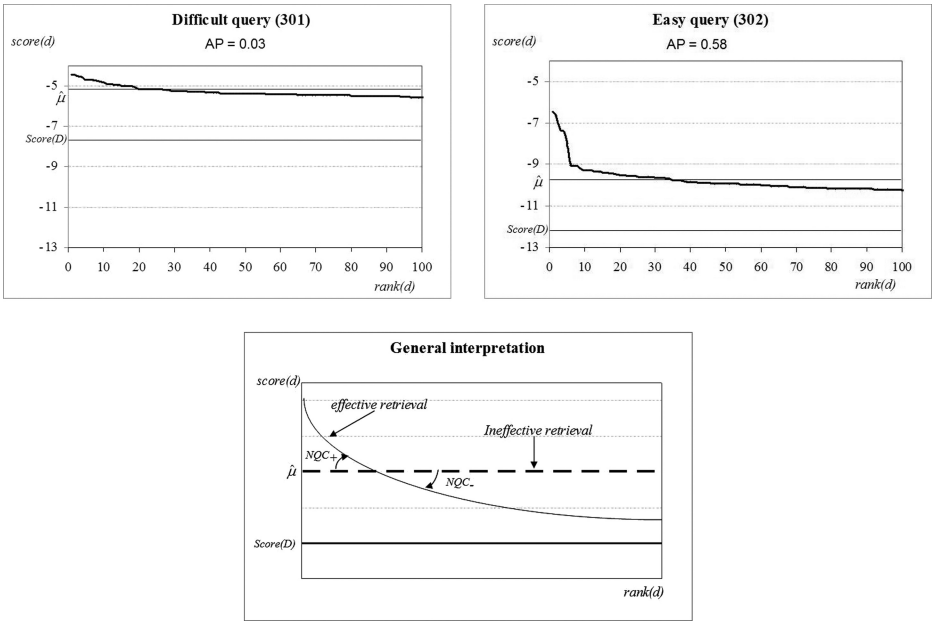


Fig. 1. Graphical vizualization of NQC. The two graphs in the first row present retrieval scores curves for "difficult" and "easy" queries, respectively. The queries are chosen from the ROBUST benchmark according to Average Precision (AP) performance of a language-model-based retrieval approach. The shift between these two scenarios, which is generalized in the graph on the second row, is quantified by NQC.

documents with retrieval scores lower than $\hat{\mu}$. The resultant estimate for retrieval effectiveness is then

$$NQC_{-}(q, \mathcal{M}) \stackrel{def}{=} \frac{\sqrt{\frac{1}{k}\sum_{d \in \mathcal{D}_q^{[k]}: Score(d) < \hat{\mu}} (Score(d) - \hat{\mu})^2}}{|Score(\mathcal{D})|}. \tag{2}$$

Integrating estimates. We integrate the estimates NQC_{+} and NQC_{-} by

$$\begin{aligned} NQC(q, \mathcal{M}) &\stackrel{def}{=} \sqrt{NQC_{+}(q, \mathcal{M})^2 + NQC_{-}(q, \mathcal{M})^2} \\ &= \frac{1}{|Score(\mathcal{D})|}\sqrt{\frac{1}{k}\sum_{d \in \mathcal{D}_q^{[k]}} (Score(d) - \hat{\mu})^2}. \end{aligned} \tag{3}$$

This NQC measure, which is our main predictor for retrieval effectiveness (i.e., query performance), is simply the standard deviation of the retrieval scores of documents in $\mathcal{D}_q^{[k]}$, normalized with respect to the retrieval score of the corpus.

In Figure 1 we exemplify how NQC can help differentiate between "difficult" and "easy" queries. The left graph on the first row presents the retrieval scores curve of an ineffective language-model-based (query likelihood) retrieval [Song and Croft 1999] performed in response to a "difficult" query. The right graph on the first row presents the scores curve for an "easy" query, for which the retrieval is highly effective. Both queries were selected based on the average precision obtained for them. As we can see, for the effective retrieval there are more documents with retrieval scores that

are higher than that of the mean retrieval score, than for the ineffective retrieval. Furthermore, the retrieval scores are much higher than the mean for the effective retrieval, while they are only slightly higher than the mean for the ineffective retrieval. For retrieval scores lower than the mean, the descent can be somewhat larger for the effective retrieval than for the ineffective retrieval. In Appendix A we present a few more examples of retrieval scores curves for easy and difficult queries, which further demonstrate the patterns just discussed.

To further exemplify the principle underlying NQC, we examine the graph on the second row in Figure 1. Suppose that all documents in $\mathcal{D}_q^{[k]}$ exhibit the same query similarity as that of the centroid, a scenario represented by the horizontal retrieval scores line that corresponds to the value $\hat{\mu}$. Then, the retrieval is of very low effectiveness, since the centroid is a misleader. On the other hand, if $\mathcal{D}_q^{[k]}$ is composed of documents with retrieval scores much higher than $\hat{\mu}$ (which corresponds to high NQC_+), and documents with much lower retrieval scores than $\hat{\mu}$ (which corresponds to high NQC_-), then the retrieval is supposedly much more effective. The extent of the shift between these two scenarios is measured by NQC.

3.4. Discussion Summary

Say that retrieval scores represent surface-level document-query similarities. The discussion presented before, then, relies on two basic arguments. First, a very high retrieval score with respect to that of the centroid, which often manifests query drift represented by lack of sufficient emphasis of query terms, implies less chances for non-query-related aspects to dominate the document. Second, low retrieval scores with respect to that of the centroid imply to a small overall population of documents in the corpus that exhibit "reasonable" query similarity; assuming that there are a few relevant documents in the corpus exhibiting reasonable query similarity, then the population of nonrelevant documents with "reasonable" query similarity in this case is small, and the less chances there are that they are part of the result list.

Naturally, there are cases not captured by the discussion presented earlier, and in which the standard deviation of retrieval scores might not serve as an effective basis for query performance prediction. Yet, we note that many other postretrieval predictors may be ineffective as well in these cases. For example, consider the ambiguous query "Jaguar". Given no additional information regarding the underlying information need, one can only aim at diversifying the result list so that relevant documents that discuss the car "Jaguar" and the cat "Jaguar" are part of the result list. High standard deviation of retrieval scores need not necessarily imply diversity following the preceding discussion. That is, the result list could be populated by documents discussing only one interpretation of the query, and still exhibit high standard deviation of retrieval scores. The same holds, for example, for the: (i) Clarity predictor [Cronen-Townsend et al. 2002] that can assign a very high prediction value to a query with a result list that is focused only on one possible interpretation of the query, (ii) the WIG predictor [Zhou and Croft 2007] that will assign a high prediction score to a query if the top-ranked documents are assigned with very high retrieval scores even if all these documents discuss only one interpretation of the query, (iii) the query-feedback predictor [Zhou and Croft 2007] that can assign a very high prediction score to a query with a result list that is focused on a single interpretation; and (iv) autocorrelation-based predictors [Diaz 2007] that as long as similar documents receive the same retrieval scores, then the assigned prediction value is high. Thus, for both the NQC predictor, and the other postretrieval predictors just mentioned, the assigned prediction value in the case of ambiguous queries can potentially reflect coverage of only one possible interpretation of the query.

3.5. Use Case: Language Modeling Framework

The effectiveness estimates we proposed can be employed with various retrieval methods that estimate relevance based on the surface-level similarity between documents and a query. Here, we focus on the language model retrieval approach [Croft and Lafferty 2003; Ponte and Croft 1998], which is a prominent retrieval paradigm.

Let $p(w|d)$ be the probability assigned to term w by a (smoothed) unigram language model induced from document d. The commonly used *Query Likelihood* (QL) retrieval method [Song and Croft 1999] scores document d in response to query $q = \{q_i\}$ by $\prod_{q_i} p(q_i|d)$. Using log transformation, the scoring function becomes

$$Score_{QL}(d) = \sum_{q_i} \log p(q_i|d). \tag{4}$$

Note that the QL-retrieval score of d for short queries is, in general, higher than that for longer queries. (The summation in Eq. (4) is of negative values.) This length bias has no effect when scoring documents in response to a given query. However, effectiveness prediction measures that rely on retrieval scores assigned in response to different queries might be affected. Nevertheless, we hasten to point out that since our measures essentially normalize the retrieval score of a document with respect to that of the corpus, they are not affected by this length bias[5].

The corpus retrieval score ($Score_{QL}(\mathcal{D})$). We treat the corpus as the big document that results from concatenating all documents; the order of concatenation has no effect, since we use unigram language models that assume term independence. Naturally, no smoothing is employed when inducing a language model from the corpus (i.e., we use a maximum likelihood estimate) [Croft and Lafferty 2003].

The centroid. We stated in Section 3.3.1 that the mean retrieval score ($\hat{\mu}$) of documents in $\mathcal{D}_q^{[k]}$ corresponds to the retrieval score of a centroid-based representation of $\mathcal{D}_q^{[k]}$. We now demonstrate this correspondence for the query likelihood model.

PROPOSITION 1. *The mean QL-retrieval score of documents in $\mathcal{D}_q^{[k]}$ is the QL score of a geometric centroid language-model-based representation of $\mathcal{D}_q^{[k]}$.*

PROOF. Let $\hat{\mu} = \frac{1}{k}\sum_{d\in\mathcal{D}_q^{[k]}} Score_{QL}(d)$. By definition, $\hat{\mu} = \frac{1}{k}\sum_{d\in\mathcal{D}_q^{[k]}}\sum_{q_i}\log p(q_i|d)$. We can rearrange the summation and write $\hat{\mu} = \sum_{q_i}\frac{1}{k}\sum_{d\in\mathcal{D}_q^{[k]}}\log p(q_i|d) = \sum_{q_i}\log\sqrt[k]{\prod_{d\in\mathcal{D}_q^{[k]}} p(q_i|d)}$. We define $p\big(w|Cent\big(\mathcal{D}_q^{[k]}\big)\big) \stackrel{def}{=} \sqrt[k]{\prod_{d\in\mathcal{D}_q^{[k]}} p(w|d)}$—a language model (modulo normalization details) that corresponds to the geometric centroid of language models of documents in $\mathcal{D}_q^{[k]}$; a similar centroid was used in recent work on cluster-based retrieval [Liu and Croft 2008; Seo and Croft 2010]. By Eq. (4), $Score_{QL}\big(Cent\big(\mathcal{D}_q^{[k]}\big)\big) = \hat{\mu}$. □

We note that the connection between the mean retrieval score of documents in $\mathcal{D}_q^{[k]}$ and the retrieval score of a centroid of $\mathcal{D}_q^{[k]}$ holds for other linear-feature-based retrieval

[5]Alternatively, we could have used, for example, query-length normalized QL scores $\left(\frac{1}{|q|}Score_{QL}(d)\right)$. These correspond to using the cross-entropy measure under certain conditions [Lafferty and Zhai 2001]. However, the query-length normalization cancels out in our estimates, because we use the corpus score as a normalizer.

Table I. Data Used for Experiments

Collection	Data	Num Docs	Topics	Rels/topic
TREC1-3	Disks 1&2	741,856	51–200	253.78
TREC4	Disks 2&3	567,529	201–250	130.06
TREC5	Disks 2&4	524,929	251–300	110.48
WT10G	WT10g	1,692,096	451–550	61.14
ROBUST	Disk 4&5-CR	528,155	301–450, 601–700	69.92
GOV2	GOV2	25,205,179	701–850	181.79

Numbers in the last column are the average number of relevant documents per topic.

functions [Metzler and Croft 2007] as well. For example, if $\vec{x}$ is the vector-space representation of text x [Salton et al. 1975], and the inner product is used as a retrieval function, then $\hat{\mu} \stackrel{def}{=} \frac{1}{k}\sum_{d\in\mathcal{D}_q^{[k]}} < \vec{q}, \vec{d} > = < \vec{q}, \frac{1}{k}\sum_{d\in\mathcal{D}_q^{[k]}} \vec{d} >$; $\frac{1}{k}\sum_{d\in\mathcal{D}_q^{[k]}} \vec{d}$ is a centroid of $\mathcal{D}_q^{[k]}$.

4. EVALUATION

4.1. Experimental Setup

We conducted our main experiments with the TREC collections specified in Table I. These collections were used in several query performance prediction studies [Diaz 2007; Yom-Tov et al. 2005; Zhou 2007; Zhou and Croft 2006]. The size of these collections enables to perform an in-depth analysis of the prediction quality of our NQC predictor and that of the reference comparisons, and to study various factors that affect these predictors. In Section 4.2.7 we present an additional evaluation performed with the large-scale ClueWeb collection.

We use the titles of TREC topics for queries, except for the TREC4 case, where no titles are provided; hence, topic descriptions are used for this collection. We applied tokenization, Porter stemming, and stopword removal (using the INQUERY list) to all data via the Lemur/Indri toolkit[6], which was also used for retrieval.

The Query Likelihood model (QL) [Song and Croft 1999] described in Section 3.5, employed with unigram language models, serves as the retrieval model, unless otherwise specified. We use Dirichlet smoothing for document language models with the smoothing parameter set to 1000 following previous recommendations [Zhai and Lafferty 2001b]. Thus, the main goal of the evaluation to follow is to predict the success (or lack thereof) of the different predictors in predicting the query performance of a standard language-model-based retrieval. In Section 4.2.5 we study the effectiveness of NQC in predicting the performance of additional retrieval methods, and in Section 4.2.8 we study its effectiveness in predicting the performance of *runs* submitted to some of TREC's tracks.

4.1.1. Reference Comparisons. We compare the prediction quality of NQC with that of three effective postretrieval predictors, namely, Clarity [Cronen-Townsend et al. 2002], Query Feedback (QF) [Zhou and Croft 2007], and WIG [Zhou and Croft 2007]. These predictors are representatives of the three categories of postretrieval methods presented in Section 2: the clarity approach (Clarity), the result list robustness approach (QF), and the retrieval scores distribution analysis approach (WIG). We note that QF and WIG were shown to post state-of-the-art prediction quality [Zhou and Croft 2007],

[6]http://www.lemurproject.org/

while Clarity is the first proposed postretrieval predictor. In Sections 4.2.6, 4.2.7, and 4.2.8, we also present a comparison of NQC with a variety of preretrieval predictors.

Clarity. The Clarity method is based on measuring the focus of the result list $\mathcal{D}_q^{[k]}$ with respect to the corpus.

Let $R_{\mathcal{D}_q^{[k]}}$ be a relevance model [Lavrenko and Croft 2001] induced from $\mathcal{D}_q^{[k]}$ as follows.

$$p\left(w|R_{\mathcal{D}_q^{[k]}}\right) \stackrel{def}{=} \sum_{d\in\mathcal{D}_q^{[k]}} p(w|d)p(d|q); \tag{5}$$

$p(d|q)$ is d's normalized query likelihood: $p(d|q) \stackrel{def}{=} \frac{p(q|d)}{\sum_{d'\in\mathcal{D}_q^{[k]}} p(q|d')} = \frac{\prod_{q_i\in q} p(q_i|d)}{\sum_{d'\in\mathcal{D}_q^{[k]}} \prod_{q_i\in q} p(q_i|d')}$. Note that the (exponent of) query likelihood scores $\left(Score_{QL}(d) = \log\prod_{q_i} p(q_i|d)\right)$ serve to differentially weigh the documents in $\mathcal{D}_q^{[k]}$ when constructing $R_{\mathcal{D}_q^{[k]}}$.

Then, the KL divergence between $R_{\mathcal{D}_q^{[k]}}$ and a maximum-likelihood-estimate-based language model induced from the corpus[7], $p(\cdot|\mathcal{D})$, serves as a measure for the focus of the result list.

$$Clarity(q;QL) \stackrel{def}{=} \sum_w p\left(w|R_{\mathcal{D}_q^{[k]}}\right)\log\frac{p\left(w|R_{\mathcal{D}_q^{[k]}}\right)}{p(w|\mathcal{D})}.$$

Higher values of Clarity, that is, higher values of the "distance" between the (relevance) model of the result list and that of the corpus, are presumed correlated with improved query performance.

We used Lemur's Clarity implementation. We found that not smoothing the document language model ($p(w|d)$) when constructing $R_{\mathcal{D}_q^{[k]}}$ in Eq. (5), and clipping $R_{\mathcal{D}_q^{[k]}}$ so as to use only the 100 terms to which it assigns the highest probabilities [Abdul-Jaleel et al. 2004], results in highly effective performance prediction. Specifically, this setting yields much better prediction performance than that of a setting suggested in work on Clarity optimization [Cronen-Townsend et al. 2006], wherein no term clipping is applied.

Query feedback. The Query Feedback (QF) measure [Zhou and Croft 2007] models the retrieval as a communication channel problem. The input is the query, the channel is the search system, and the set of results is the noisy output of the channel. A relevance model $R_{\mathcal{D}_q^{[k]}}$, constructed from the result list $\mathcal{D}_q^{[k]}$, serves as a "query"[8] for ranking the corpus. Specifically, the KL divergence between a Dirichlet smoothed document language model and $R_{\mathcal{D}_q^{[k]}}$ is the ranking criterion. Then, the overlap at cutoff m_{QF} (m_{QF} is a free parameter) between the retrieved document list and the result list $\mathcal{D}_q^{[k]}$ is used for performance prediction.

We note that QF could conceptually be thought of, although not explicitly stated as such [Zhou and Croft 2007], as a measure for quantifying the potential amount of query drift in $\mathcal{D}_q^{[k]}$. That is, the higher the overlap between the rankings induced by the

[7]The maximum likelihood estimate of term w with respect to text x is $\frac{\text{tf}(w\in x)}{\sum_{w'}\text{tf}(w'\in x)}$, where $\text{tf}(w \in x)$ is the number of occurrences of w in x. As noted before, the corpus is represented by the big document that results from concatenating the documents it contains.

[8]To maintain consistency with Clarity computation, we use the relevance model defined in Eq. (5) for ranking the corpus. We note that this relevance model is somewhat different than the query model originally used in Zhou and Croft [2007], but still results in highly effective query performance prediction.

query and by the relevance model, the less query drift manifested in $\mathcal{D}_q^{[k]}$ is presumed to be. Hence, in spirit, QF is connected with the query-drift quantification idea that NQC is based on.

WIG. The WIG predictor measures the divergence of retrieval scores of top-ranked documents from that of the corpus. WIG was originally proposed and employed in the Markov Random Field (MRF) retrieval framework [Metzler and Croft 2005]. However, if no term dependencies are considered, that is, a bag-of-terms representation is used, then MRF reduces to using the query likelihood model with unigram language models. Indeed, it was noted that WIG is very effective with this implementation [Zhou 2007]. In this case, WIG can be computed by

$$WIG(q, QL) \stackrel{def}{=} \frac{1}{k} \sum_{d_i \in \mathcal{D}_q^{[k]}} \frac{1}{\sqrt{|q|}} (Score_{QL}(d_i) - Score_{QL}(\mathcal{D})). \quad (6)$$

$|q| \stackrel{def}{=} \#\{q_i \in q\}$ is q's length. Refer back to the discussion in Section 3.5 regarding the importance of query-length normalization.

4.1.2. Evaluation Paradigms. The goal of all the query performance predictors presented earlier is to estimate the retrieval effectiveness (query performance) of a *given* retrieval method with respect to a *given* query.

There are two evaluation measures that are commonly used in the query performance prediction framework for evaluating prediction performance. The first is Pearson's correlation coefficient (ρ), henceforth denoted P-ρ. The correlation is computed between the *actual* average precision (AP at a cutoff of 1000) values for queries in a given query set (as measured by using relevance judgments), and the values assigned to these queries by a prediction measure [Carmel et al. 2006]. The second evaluation measure is Kendall's$-\tau$ correlation coefficient, K-τ in short. The correlation is computed between a ranking of the queries that corresponds to the actual AP attained for them and a ranking induced by the assigned values of the prediction measure [Voorhees 2004]. For both evaluation measures, higher correlation values indicate increased prediction performance.

The prediction performance of our NQC measure, and that of the three reference comparisons, can vary with respect to the number of documents, k, in the result list. A case in point, WIG was claimed to be most effective when computed over very short retrieved lists of about 5 documents [Zhou 2007]. Effective Clarity-based prediction, on the other hand, calls for using longer result lists. Furthermore, the prediction performance of QF depends not only on the result list size (k), but also on the cutoff m_{QF} used for computing document overlap. Most previous work on comparing the performance of query performance predictors has used for each predictor fixed values of k that were effective across corpora. Naturally, and as we show shortly, the optimal value of k can vary with respect to the corpus used.

Thus, we employ two different evaluation paradigms that can help address the impact of k (and that of m_{QF} for QF) on the resultant prediction performance. The first is based on using for each predictor, per each corpus, values of free parameters that yield optimal prediction performance. The prediction performance is measured using either Pearson's correlation or Kendall's$-\tau$ as computed over all queries per corpus. This evaluation approach, which we refer to as *Optimal*, is intended to contrast the potential effectiveness of the predictors while ameliorating free-parameter values effects. Later on we study the effect of varying the values of free parameters on prediction performance.

The second paradigm that we employ is based on learning free-parameter values. For each corpus, we randomly split the queries into two equal sized sets that are used in a cross-validation procedure. That is, we use one set, the "train set", to find the free-parameter values of a predictor that yield optimal prediction performance (as measured by Pearson's correlation or Kendall's$-\tau$) with respect to this set. Then, the predictor is employed with these optimal free-parameter values so as to predict performance over the second set of queries: the "test set". We then flip the roles of the two sets of queries. Since using a single split of the queries might result in a biased evaluation of prediction performance, we use 40 random splits and report the average, and standard deviation of, prediction performance over the test sets. Accordingly, we refer to this evaluation paradigm as *Cross-Validation*.

It is also important to note that the prediction performance numbers attained by a predictor using the Optimal and Cross-Validation evaluation paradigms are not necessarily comparable to each other. That is, while the Optimal paradigm is based on measuring correlation with true performance over *all* queries per corpus, the Cross-Validation paradigm is based on measuring average correlation over (many) splits of the queries. Thus, the prediction performance of different predictors is compared with respect to each paradigm separately: the Optimal paradigm serves to compare the potential prediction performance of predictors, and the Cross-Validation paradigm serves to compare prediction performance when values of free parameters are learned using a held-out query set.

The values of the free parameters of the predictors are set as follows. The result list size, k, is set to values in {5, 10, 50, 100, 150, 200, 300, 500, 700, 1000, 2000, 3000, 4000, 5000} for our NQC measure, as well as for the Clarity, WIG, and QF methods. Experiments show (see Section 4.2.2) that the prediction performance of QF is much more sensitive with respect to the value of m_{QF} (the cutoff used for computing document overlap) than to the value of k, which affects relevance-model construction; the value of m_{QF} is selected from {5, 10, 50, 100, 150, 200, 300, 500, 700, 1000}.

4.2. Experimental Results

4.2.1. Main Result. Our first order of business is comparing the prediction quality of NQC with that of the reference comparisons, Clarity, QF, and WIG. Table II presents the prediction-quality numbers when using the Optimal and Cross-Validation evaluation paradigms.

Examination of Table (a) of Table II, which presents prediction-quality numbers measured using the Optimal paradigm, reveals the following.

Our NQC measure outperforms Clarity and WIG in a vast majority of the *relevant comparisons* (6 corpora $\times$ 2 evaluation measures—Pearson's coefficient and Kendall's-τ). Specifically, NQC outperforms Clarity in 11 out of the 12 relevant comparisons, and WIG in 10 out of the 12 comparisons. On the other hand, QF posts the best prediction quality in Table II for 7 out of the 12 relevant comparisons. We hasten to point out, however, that this might be a result of the fact that QF relies on two free parameters (the result list size (k) and the cutoff parameter (m_{QF})), while NQC, Clarity, and WIG rely only on a single free parameter (k). Indeed, when learning free parameters' values, NQC outperforms QF in a majority of relevant comparisons, as will be discussed shortly.

We next examine Table (b) of Table II, which is based on the Cross-Validation paradigm. We can see that NQC outperforms each of the three reference comparisons (Clarity, QF, and WIG) in a majority of the relevant comparisons (6 corpora $\times$ 2 evaluation measures). Furthermore, each of NQC and QF posts the best prediction quality in 4 out of the 12 relevant comparisons, while Clarity and WIG do so in only 2 out of the 12

Table II. Main Result

(a) Evaluation using the Optimal paradigm.

	TREC1-3		TREC4		TREC5		ROBUST		WT10G		GOV2	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	.472	.360	.476	.370	.431	.318	.522	.408	.432	.368	.456	.315
QF	.691	**.501**	**.651**	**.511**	.447	**.468**	.500	.398	.483	**.372**	**.566**	**.413**
WIG	.683	.459	.554	.502	.297	.258	.550	.386	.376	.301	.486	.340
NQC	**.700**	.465	.641	.494	**.502**	.340	**.566**	**.419**	**.527**	.331	.462	.362

(b) Evaluation using the Cross-Validation paradigm. (Numbers in parentheses indicate the standard deviation of prediction quality.)

	TREC1-3		TREC4		TREC5	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	.444(.071)	.347(.054)	.458(.122)	.351(.095)	**.399**(.121)	.288(.088)
QF	.664(.050)	**.481**(.045)	.581(.142)	.440(.106)	.275(.134)	**.312**(.121)
WIG	.677(.050)	.453(.042)	.538(.097)	**.482**(.068)	.268(.151)	.221(.109)
NQC	**.690**(.040)	.455(.044)	**.586**(.120)	.456(.086)	.365(.167)	.272(.094)
	ROBUST		WT10G		GOV2	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	.516(.062)	.396(.037)	.429(.073)	**.361**(.068)	.418(.076)	.290(.060)
QF	.465(.056)	.362(.048)	.387(.091)	.291(.063)	**.510**(.082)	**.372**(.063)
WIG	**.540**(.055)	.382(.038)	.368(.101)	.293(.066)	.483(.071)	.336(.050)
NQC	.535(.055)	**.409**(.037)	**.487**(.119)	.286(.085)	.416(.066)	.343(.047)

Comparison of NQC with Clarity, QF, and WIG using the Optimal and Cross-Validation evaluation paradigms. Boldface marks the best result in a column.

relevant comparisons. Yet, it is important to note that the standard deviation of NQC's prediction quality is smaller than that of QF in 9 out of the 12 relevant comparisons. This finding could be attributed to the fact that QF incorporates two free parameters while NQC incorporates only one. The standard deviation of NQC's prediction quality is also: (i) often smaller than that of Clarity, and (ii) sometimes larger and sometimes smaller than that of WIG, that is, no clear dominance is observed.

All in all, the results presented in Table II show that NQC is a highly effective predictor that yields prediction quality that is quite robust (in terms of standard deviation) with respect to that of the other three predictors considered.

4.2.2. Analysis of the Effect of Free-Parameter Values. In Figure 2 we present the effect of varying the value of k, the result list size, on the prediction quality (measured using Pearson's correlation) of NQC and that of the reference comparisons. Recall that k is the only free parameter incorporated by NQC, Clarity, and WIG; for QF, we set m_{QF}, its second free parameter, to 50, which yields effective prediction quality on average.

Our first observation based on Figure 2 is that the prediction quality of Clarity and QF is quite robust with respect to the value of k for most collections. This is due to the fact that the relevance model constructed from the initial list weighs documents by their QL (query likelihood) retrieval scores. (Refer back to Eq. (5)). Hence, low ranked documents have little effect on the constructed relevance model, and consequently, on the resultant Clarity and QF assigned prediction values. (A notable exception for the prediction-quality robustness of QF is the GOV2 collection.) In general, using values

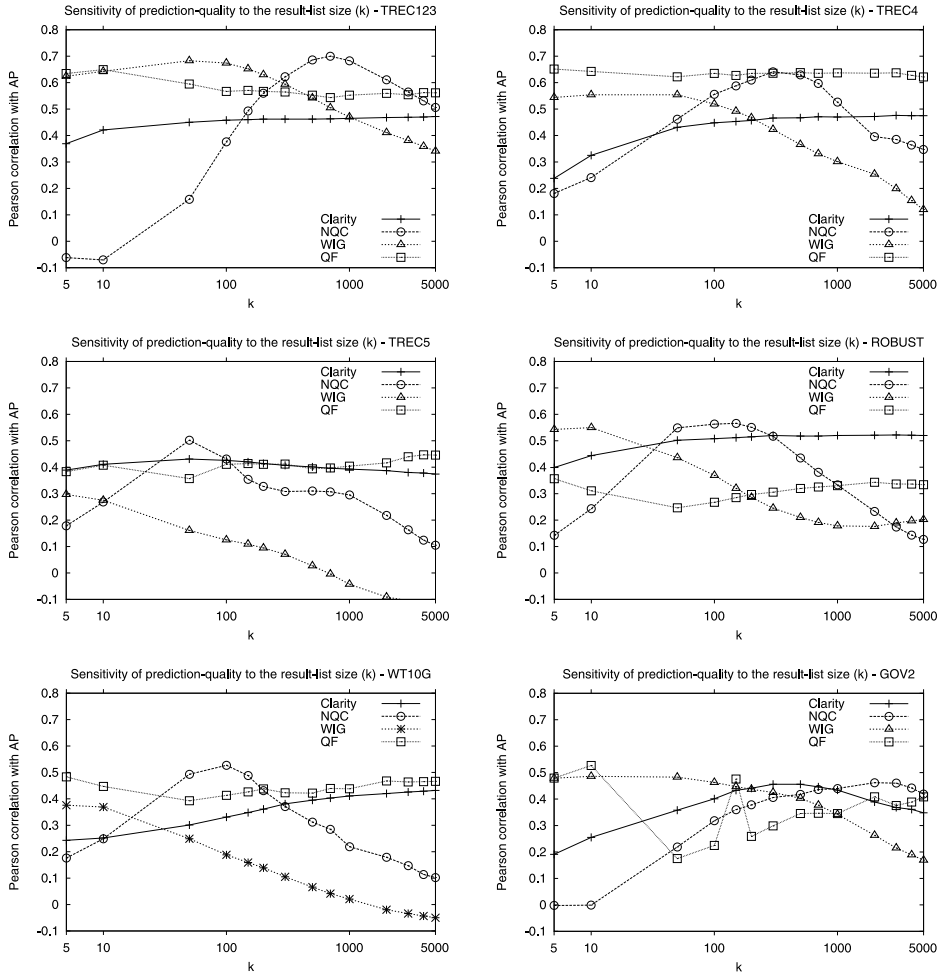


Fig. 2. Prediction quality (measured by Pearson's coefficient) of Clarity, QF, WIG, and NQC as a function of the result list size (k).

of k that are above 500 yields effective prediction quality for both Clarity and QF over all collections.

We can also see in Figure 2 that the prediction performance of WIG is optimal for low values of k; often, for $k \in \{5, 10\}$. This finding is in accordance with previous reports [Zhou and Croft 2007]. Furthermore, WIG's prediction quality degrades when increasing the value of k.

The prediction quality curve of NQC has a relatively similar shape across collections; that is, with increasing values of k there is a prediction-quality increase up to a peak that is obtained at values of k that are often above 100; then, further increasing the value of k results in a prediction-quality decrease. While the value of k at which NQC attains optimal prediction quality can vary across collections, we note that NQC is highly effective when $k \in \{100, 150\}$ for most collections.

Another interesting observation that we make based on Figure 2 is the relatively high value of k for which NQC attains its best prediction quality for TREC1-3 and GOV2 with respect to that for the other collections (which is around 150); the

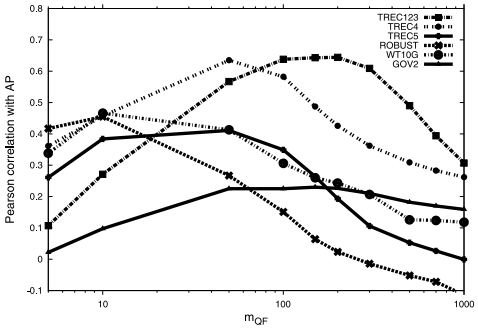


Fig. 3. Prediction quality (measured by Pearson's coefficient) of QF as a function of the overlap cutoff parameter (m_{QF}); the result list size, k, is set to 100.

observation with regard to TREC1-3 also holds for WIG, which attains the best prediction quality for $k = 50$, while for the other collections $k \in \{5, 10\}$ yields the best WIG prediction quality. These findings could potentially be attributed to the fact that for TREC1-3 and GOV2 the average number of relevant documents per query is much larger than that for the other collections. (See Table I.) Indeed, for the centroid of the result list to be a prototypical misleader (refer to the discussion in Section 3.3.1), an assumption on which NQC is based, the result list considered should contain a "descent" amount of nonquery-pertaining aspects; hence, examining a relatively long result list when there are many relevant documents that can be ranked quite high is important as is the case for TREC1-3 and GOV2.

Finally, we hasten to point out that no single predictor dominates the other predictors over the entire tested range of k across the collections.

A note on QF. The findings presented before demonstrate that QF is a highly effective predictor; and, that when fixing the value of m_{QF} (specifically, to 50), the overlap cutoff parameter on which QF depends, QF posts relatively stable prediction quality with respect to the size of the result list (k). In Figure 3 we present the effect of varying the value of m_{QF} on QF's prediction performance; k is set to 100; using larger values of k has little effect on the constructed relevance model that QF uses as noted earlier. We can see in Figure 3 that the optimal value of m_{QF} can vary across collections. Thus, while the prediction quality of QF is quite robust with respect to the value of k, this is not the case with respect to the value of m_{QF}.

4.2.3. NQC Subcomponents. Table III presents the prediction quality of the measures integrated by NQC, namely, NQC_+ and NQC_-. Recall that NQC_+ addresses documents with retrieval scores above the mean score, $\hat{\mu}$, while NQC_- addresses those with retrieval scores below $\hat{\mu}$. For reference comparisons we use NQC itself, and the three predictors used before: Clarity, QF, and WIG.

It is evident in Table III that NQC_+ and NQC_- are highly effective in prediction. For example, there are numerous relevant comparisons for which these measures outperform Clarity, QF, and WIG. Of a specific interest is the comparison of NQC_+ with WIG. NQC_+ measures the deviation of top-retrieved scores from $\hat{\mu}$, which was argued earlier to represent the retrieval score of a pseudo nonrelevant document exhibiting relatively high query similarity. WIG, on the other hand, measures the deviation of top-retrieved scores from that of the corpus, which could be considered as a general nonrelevant document. Now, we can see in Table III that NQC_+ outperforms WIG in a vast majority of the relevant comparisons (6 corpora $\times$ 2 evaluation measures). This finding further supports the merits of using the mean retrieval score in the result list

Table III. Prediction Quality of NQC Subcomponents: NQC_+ and NQC_-

(a) Evaluation using the Optimal paradigm.

	TREC1-3		TREC4		TREC5		ROBUST		WT10G		GOV2	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	.472	.360	.476	.370	.431	.312	.522	.408	.432	.368	.456	.315
QF	.691	**.501**	**.651**	**.511**	.447	**.468**	.500	.398	.483	**.372**	**.566**	**.413**
WIG	.683	.459	.554	.500	.297	.258	.550	.386	.376	.300	.486	.340
NQC_+	.698	.464	.638	.487	.491	.336	**.574**	.418	**.531**	.326	.468	.364
NQC_-	**.700**	.470	.630	.496	**.521**	.353	.565	.418	.507	.329	.421	.349
NQC	**.700**	.465	.641	.494	.502	.340	.566	**.419**	.527	.331	.462	.362

(b) Evaluation using the Cross-Validation paradigm. (Numbers in parentheses indicate the standard deviation of prediction quality.)

	TREC1-3		TREC4		TREC5	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	.444(.071)	.347(.054)	.458(.122)	.351(.095)	**.399**(.121)	.288(.088)
QF	.664(.050)	**.481**(.045)	.581(.142)	.440(.106)	.275(.134)	**.312**(.121)
WIG	.677(.050)	.453(.042)	.538(.097)	**.482**(.068)	.268(.151)	.221(.109)
NQC_+	.688(.041)	.455(.043)	**.591**(.116)	.447(.104)	.351(.180)	.285(.090)
NQC_-	.685(.043)	.458(.045)	.567(.105)	.456(.085)	.371(.220)	.282(.102)
NQC	**.690**(.040)	.455(.044)	.586(.120)	.456(.086)	.365(.167)	.272(.094)

	ROBUST		WT10G		GOV2	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	.516(.062)	.396(.037)	.429(.073)	**.361**(.068)	.418(.076)	.290(.060)
QF	.465(.056)	.362(.048)	.387(.091)	.291(.063)	**.510**(.082)	**.372**(.063)
WIG	.540(.055)	.382(.038)	.368(.101)	.293(.066)	.483(.071)	.336(.050)
NQC_+	**.542**(.068)	.408(.037)	**.488**(.118)	.282(.081)	.430(.062)	.345(.048)
NQC_-	.531(.067)	.406(.040)	.471(.121)	.294(.088)	.351(.056)	.318(.053)
NQC	.535(.055)	**.409**(.037)	.487(.119)	.286(.085)	.416(.066)	.343(.047)

Best result per column is boldfaced.

as a reference comparison for computing deviation of retrieval scores. We revisit this point shortly.

The next comparison we are interested in is that of NQC with its subcomponents, NQC_+ and NQC_-. Table (a) of Table III shows that with the Optimal evaluation paradigm, by which the most effective result list size (k) for *each* of these three predictors is used, NQC outperforms its subcomponents in a majority of the relevant comparisons when considering pairwise comparisons, that is, comparing NQC with NQC_+ and comparing NQC with NQC_-. Furthermore, neither NQC_+ nor NQC_+ seems to dominate the other in terms of prediction quality. With the Cross-Validation evaluation paradigm (Table (b)), NQC outperforms (and also posts a lower standard deviation of prediction quality than that of) NQC_- in most relevant comparisons, but NQC does not dominate (nor is dominated by) NQC_+ across the relevant comparisons.

All in all, the findings discussed before support the analysis in Section 3 with respect to the merits of using both NQC_+ and NQC_- for query performance prediction.

4.2.4. KL-Divergence-Based Performance Prediction. The NQC measure is based on the dispersion of retrieval scores in $\mathcal{D}_q^{[k]}$ from that of the mean score, $\hat{\mu}$. To further

Table IV. Comparison of the Prediction Quality of the KL-Based Predictor with that of the Other Predictors

(a) Evaluation using the Optimal paradigm.

	TREC1-3		TREC4		TREC5		ROBUST		WT10G		GOV2	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	.472	.360	.476	.370	.431	.312	.522	.408	.432	**.368**	.456	.315
QF	.655	**.466**	.638	**.511**	.414	**.406**	.481	.390	.473	.343	.476	.336
WIG	.683	.459	.554	.500	.297	.258	.550	.386	.376	.300	**.486**	.340
NQC	**.700**	.465	**.641**	.494	.502	.340	**.566**	.419	**.527**	.331	.462	.362
KL	.693	.463	.558	.474	**.510**	.345	.558	**.420**	.379	.259	.424	**.377**

(b) Evaluation using the Cross-Validation paradigm. (Numbers in parentheses indicate the standard deviation of prediction quality.)

	TREC1-3		TREC4		TREC5	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	.444(.071)	.347(.054)	.458(.122)	.351(.095)	**.399**(.121)	.288(.088)
QF	.664(.050)	**.481**(.045)	.581(.142)	.440(.106)	.275(.134)	**.312**(.121)
WIG	.677(.050)	.453(.042)	.538(.097)	**.482**(.068)	.268(.151)	.221(.109)
NQC	**.690**(.040)	.455(.044)	**.586**(.120)	.456(.086)	.365(.167)	.272(.094)
KL	.682(.045)	.455(.044)	.509(.133)	.447(.084)	.389(.211)	.292(.097)
	ROBUST		WT10G		GOV2	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	.516(.062)	.396(.037)	.429(.073)	**.361**(.068)	.418(.076)	.290(.060)
QF	.465(.056)	.362(.048)	.387(.091)	.291(.063)	**.510**(.082)	**.372**(.063)
WIG	**.540**(.055)	.382(.038)	.368(.101)	.293(.066)	.483(.071)	.336(.050)
NQC	.535(.055)	.409(.037)	**.487**(.119)	.286(.085)	.416(.066)	.343(.047)
KL	.527(.054)	**.410**(.038)	.364(.112)	.214(.081)	.366(.054)	.365(.050)

Boldface marks the best result in a column.

explore the importance of $\hat{\mu}$ as a basis for computing score divergence, we examine an alternative for measuring the overall diversity of retrieval scores in $\mathcal{D}_q^{[k]}$. We do so by measuring the KL divergence between the (normalized) retrieval-scores function and a constant score function. The latter corresponds to scoring all documents in $\mathcal{D}_q^{[k]}$ by $\hat{\mu}$. Note that such a scenario represents ineffective retrieval as described in Section 3. Thus, the more distant the score function is from a constant one (i.e., larger KL value), the more effective the retrieval is presumed to be. To perform this estimation, we define for $i \in \{1, \ldots, k\}$, $U(i) \stackrel{def}{=} \frac{1}{k}$ and $P(i) \stackrel{def}{=} \frac{Score(d_i)}{\sum_{d_j \in \mathcal{D}_q^{[k]}} Score(d_j)}$. Then, $KL(P(\cdot)||U(\cdot)) = \log k + \sum_{i=1}^{k} P(i) \log P(i)$ is the (regularized) minus entropy of the score function[9].

In Table IV we compare the prediction quality of the KL-based measure with that of NQC, Clarity, QF, and WIG.

The results in Table IV show that the KL-based measure is highly effective in predicting query performance. Specifically, its prediction quality transcends in many cases that of Clarity, QF, and WIG. This finding gives further support to the connection

[9]The entropy of the score function is defined as $Ent(P(\cdot)) \stackrel{def}{=} -\sum_{i=1}^{k} P(i) \log P(i)$; its maximum value is $\log k$. Higher entropy implies less diversity of scores.

between the diversity of retrieval scores in $\mathcal{D}_q^{[k]}$ and query performance. We can also see in Table IV that NQC is more effective than the KL-based measure in most relevant comparisons, both for the Optimal and Cross-Validation evaluation paradigms; the robustness of NQC and the KL-based measure, as manifested in the standard deviation of their prediction quality that is reported in Table (b) of Table IV, is on par. The superiority of NQC to the KL-based measure supports the importance of using $\hat{\mu}$, the mean retrieval score in the result list, as a basis for estimating score divergence.

4.2.5. Performance Prediction for Additional Retrieval Models. Heretofore, the evaluation focused on a language-model-based retrieval model, that is, the query likelihood approach. However, our NQC measure can be employed with other retrieval methods wherein relevance is estimated based on surface-level query similarity. (Refer back to Section 3 for discussion.) Thus, we also test NQC with the cosine measure in the vector-space model [Salton et al. 1975]. Since cosine scores are embedded in the unit sphere, further normalization with the corpus retrieval score is redundant. In fact, such normalization degrades prediction quality and is therefore not used for prediction. We also employ NQC to predict the effectiveness of Okapi-BM25 retrieval [Robertson et al. 1994]. We use the centroid of all documents in the corpus, rather than a simple concatenation of the documents, as a basis for computing the corpus retrieval score so as to avoid issues that arise from using very long documents [Fang et al. 2004; Lv and Zhai 2011]. We used Lemur's implementation of both retrieval methods (cosine and Okapi) with default parameter settings.

As reference comparisons to NQC, we use two variants of Clarity. The first, *Clarity(score)*, uses the sum-normalized retrieval scores for weighing documents when constructing the relevance model rather than the query-likelihood scores (refer back to Eq. (5)). The second, *Clarity(rank)* [Cronen-Townsend et al. 2006], is based on using a linear function of document ranks, rather then their retrieval scores, as weights.

We also explored several predictors that are variants (in spirit) of the WIG predictor and which are based on using Eq. (6). For example, we used raw retrieval scores with and without query-length normalization and with and without using the corpus retrieval score. (For Okapi, as is the case for NQC, the corpus retrieval score is that of the centroid of all documents in the corpus; and, in the cosine case, the corpus retrieval score is not used.) Furthermore, we experimented with retrieval scores that are sum-normalized with respect to all scores in the result list. We found that the most effective variant of WIG for Okapi is the mean of raw retrieval scores in the result list normalized by the query length as in Eq. (6), that is, using the corpus score resulted in decreased prediction quality. For cosine, the mean of sum-normalized retrieval scores in the result list turned out to be the most effective WIG variant; applying query-length normalization decreased prediction quality. (As mentioned before in the cosine case, the corpus retrieval score was not used.) These two variants of WIG were used for Okapi and cosine, respectively, in the prediction-quality comparison presented in Table V.[10]

We can see in Table V that for the Optimal paradigm, NQC posts the highest prediction quality in 13 out of 20 relevant comparisons (5 corpora $\times$ 2 retrieval methods $\times$ 2 evaluation measures). For the Cross-Validation paradigm, NQC is the best-performing predictor in 9 out of 20 relevant comparisons, while the WIG variants are the most effective in 8 relevant comparisons. The Clarity variants are often less effective than WIG and NQC for both the the Optimal and Cross-Validation paradigms. (In most

[10]We omit GOV2 from the evaluation here, as Lemur's implementation of Okapi-BM25 and cosine in the vector-space model does not scale up so as to handle large corpora such as GOV2.

Table V. Comparison of NQC with Clarity(score), Clarity(rank) and WIG for the Okapi-BM25 and Vector Space Model (with the cosine measure) Retrieval Methods

(a) Evaluation using the Optimal paradigm.

	TREC1-3				TREC4			
	Okapi		Cosine		Okapi		Cosine	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity(score)	.535	.392	.336	.210	.605	.468	.452	.280
Clarity(rank)	.526	.396	.259	.197	.609	.463	.415	.248
WIG	**.701**	**.489**	.500	.403	**.635**	**.554**	.625	.271
NQC	.651	.444	**.639**	**.407**	.628	.486	**.664**	**.379**

	TREC5				ROBUST			
	Okapi		Cosine		Okapi		Cosine	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity(score)	.423	.302	.413	.267	.335	.241	.410	.320
Clarity(rank)	.416	.312	.380	.249	.341	.241	.380	.306
WIG	.342	.284	.319	.235	.493	.367	.485	**.410**
NQC	**.472**	**.372**	**.455**	**.272**	**.603**	**.381**	**.544**	.383

	WT10G			
	Okapi		Cosine	
	P-ρ	K-τ	P-ρ	K-τ
Clarity(score)	.173	.130	.112	.091
Clarity(rank)	.167	.120	.000	.052
WIG	**.432**	**.331**	.015	.303
NQC	.311	.305	**.416**	**.339**

(b) Evaluation using the Cross-Validation paradigm. (Numbers in parentheses indicate the standard deviation of prediction quality.)

	TREC1-3			
	Okapi		Cosine	
	P-ρ	K-τ	P-ρ	K-τ
Clarity(score)	.526(.050)	.389(.042)	.276(.091)	.180(.064)
Clarity(rank)	.511(.053)	.389(.043)	.198(.082)	.178(.062)
WIG	**.694**(.044)	**.487**(.039)	.486(.106)	.379(.044)
NQC	.631(.048)	.427(.051)	**.605**(.062)	**.385**(.052)

	TREC4			
	Okapi		Cosine	
	P-ρ	K-τ	P-ρ	K-τ
Clarity(score)	.542(.148)	.431(.106)	.322(.219)	.190(.131)
Clarity(rank)	.534(.140)	.411(.100)	.262(.221)	.118(.147)
WIG	**.615**(.116)	**.543**(.076)	**.553**(.162)	.186(.116)
NQC	.586(.114)	.450(.101)	.542(.172)	**.308**(.095)

	TREC5			
	Okapi		Cosine	
	P-ρ	K-τ	P-ρ	K-τ
Clarity(score)	**.324**(.123)	.236(.104)	**.326**(.152)	**.187**(.114)
Clarity(rank)	.313(.109)	.253(.104)	.269(.149)	.177(.125)
WIG	.302(.131)	.255(.106)	.271(.135)	.184(.093)
NQC	.289(.171)	**.314**(.112)	.308(.166)	.178(.092)

	ROBUST			
	Okapi		Cosine	
	P-ρ	K-τ	P-ρ	K-τ
Clarity(score)	.306(.061)	.222(.036)	.354(.088)	.301(.037)
Clarity(rank)	.322(.053)	.230(.035)	.358(.063)	.292(.036)
WIG	.483(.060)	.364(.037)	.457(.078)	**.409**(.033)
NQC	**.587**(.036)	**.367**(.033)	**.534**(.070)	.379(.035)

	WT10G			
	Okapi		Cosine	
	P-ρ	K-τ	P-ρ	K-τ
Clarity(score)	.098(.087)	.057(.065)	−.011(.126)	.027(.083)
Clarity(rank)	.085(.091)	.042(.066)	−.072(.097)	.011(.076)
WIG	**.418**(.084)	**.316**(.054)	.123(.236)	.286(.070)
NQC	.288(.155)	.255(.071)	**.390**(.083)	**.316**(.058)

Boldface marks the best result in a column.

cases, Clarity(score) outperforms Clarity(rank).) Thus, we conclude that as was the case for the language-model-based retrieval method used earlier, NQC is a highly effective predictor when using the Okapi-BM25 and vector-space retrieval methods.

4.2.6. Comparison with Preretrieval Predictors. The NQC measure is a postretrieval predictor, as it operates on the result list of top-retrieved documents; so are Clarity, QF, and WIG that have served, heretofore, as reference comparisons. Now, we turn to compare the prediction quality of NQC with that of some effective preretrieval predictors, which use only the query and corpus-based information.

The first set of predictors we consider, referred to as *SCQ*, is based on the product of the term frequency and inverse document frequency of a query term, wherein both frequencies are computed with respect to the entire corpus [Zhao et al. 2008]. The idea is that a query term that is frequent in the corpus, but does not appear in many documents, has a high discriminative power. Thus, a query containing such terms could be considered relatively easy. Hence, conceptually, this query performance prediction approach is a preretrieval analog of the Clarity postretrieval predictor that measures how focused is the result list, rather than the query itself, with respect to the corpus.

We use SumSCQ(TF.IDF), AvgSCQ(TF.IDF), and MaxSCQ(TF.IDF) to denote the sum, average, and maximum, respectively, computed over all query terms, of the frequency values just mentioned[11]. As our goal is predicting the effectiveness of a language-model-based retrieval, we also experimented with variants of SumSCQ (TF.IDF), AvgSCQ(TF.IDF), and MaxSCQ(TF.IDF) that use language-model-based estimates, rather than term frequency and document frequency values. However, the resultant language-model-based predictors turned out to be less effective than those originally proposed [Zhao et al. 2008]; hence, we do not report their prediction quality numbers.

The second set of preretrieval predictors that we use for reference comparisons is denoted *Var*, as these predictors are based on the variance of the TF.IDF values of query terms across documents in the corpus [Zhao et al. 2008]. Specifically, the TF.IDF value for each query term within each document in the corpus it appears in (i.e., the product of its term frequency in the document with its inverse document frequency in the corpus) is computed. Then, the variance of the term's TF.IDF values across documents in the corpus is used as a basis for prediction. Terms with high variance are considered as more discriminative, and hence, can potentially attest to the difficulty of the query that contains them. We use SumVar(TF.IDF), AvgVar(TF.IDF), and MaxVar(TF.IDF) to denote the sum, average, and maximum, computed over all query terms, of the term-based variance values.[12]

It is interesting to note that the SumVar(TF.IDF) predictor is a conceptual preretrieval reminiscent of our NQC postretrieval predictor. That is, SumVar(TF.IDF) sums the variance of TF.IDF values of query terms, wherein variance is computed across *all* documents in the corpus that a term appears in; NQC, on the other hand, measures the standard deviation of retrieval scores of documents in the result list of top-retrieved documents. Although document retrieval scores that are computed in response to a query may not be linear in per-term retrieval scores; and, as in the language-model case here, TF.IDF might not serve as a basis for estimates, SumVar(TF.IDF) can still be conceptually thought of as a specific preretrieval approximation of NQC. (Note that SumVar(TF.IDF) can be computed at index time, prior to retrieval time.) Furthermore,

[11]In the original report [Zhao et al. 2008], SumSCQ(TF.IDF), AvgSCQ(TF.IDF), and MaxSCQ(TF.IDF) were termed SCQ, NSCQ, and MaxSCQ, respectively. We use a different notation here for consistency with the names of other predictors we use as reference comparisons.

[12]These predictors were originally denoted σ_1, σ_2, and σ_3, respectively.

to better equate the frequency-based estimates that the Var predictors and the NQC predictor rely on, that is, to focus on the difference between using variance-based estimates over all documents in the corpus and over the result list, we also report the prediction quality of SumVar(LM), AvgVar(LM), and MaxVar(LM); these are the language-model-based analogs of SumVar(TF.IDF), AvgVar(TF.IDF), and MaxVar(TF.IDF), respectively.

The third set of preretrieval predictors that we use as reference comparisons is based on the *IDF* (Inverse Document Frequency) values of query terms. Such predictors were used in several studies [Cronen-Townsend et al. 2002; Hauff et al. 2008a, 2010]. Specifically, we use SumIDF, AvgIDF, and MaxIDF to denote the sum, average, and maximum IDF values of query terms.

The prediction-quality comparison of NQC with the preretrieval predictors is presented in Table VI. We use both the Optimal and Cross-Validation evaluation paradigms. Note that while NQC depends on a single free parameter (the result list size), the preretrieval predictors do not incorporate any free parameters; yet, their prediction quality can vary with respect to the specific set of queries at hand, as is shown in Table VI.

We can see in Table VI that the prediction quality of NQC is in almost all cases better (often by quite a large margin) than that of the preretrieval predictors considered. This finding holds both for the preretrieval predictors that use the TF.IDF estimates, and those that use the language-model-based (LM) estimates. This finding is not surprising as NQC, as a postretrieval predictor, analyzes the result list, while the preretrieval predictors only use the information in the query and the corpus.

We also observe in Table VI that among the preretrieval predictors, those that belong to the Var set are often the most effective. This finding is in line with previous reports [Zhao et al. 2008]. Finally, we note that, although NQC incorporates a free parameter and the preretrieval predictors do not, the standard deviation of NQC's prediction quality (under the Cross-Validation paradigm) is often smaller than that of the preretrieval predictors. This finding further attests to the overall prediction-quality robustness of NQC.

4.2.7. Query Performance Prediction for ClueWeb. The evaluation presented earlier was conducted using a variety of TREC corpora that were also used in previous work on query performance prediction. The largest corpus among these is GOV2, a Web collection that contains about 25 million documents that were crawled from the .GOV domain. We now turn to study the prediction quality of NQC, and that of the reference comparisons (both postretrieval and preretrieval predictors), when employed over a large noisy Web corpus. Specifically, we use the ClueWeb benchmark (category B) [Clarke et al. 2009], which is an English Web collection that contains about 50 million documents; the topics used are 1-50 from TREC 2009. As noted before, titles of TREC's topics serve for queries and Porter stemming and stopword removal (using the INQUERY list) are applied to all data.

We use the Query Likelihood (QL) [Song and Croft 1999] model for a retrieval method; the document language model Dirichlet smoothing parameter was set to 1000 as before. We apply two retrieval approaches. The first is using the *QL* method for ranking all documents in the corpus as was the case before. The second approach, denoted *QL+SpamRemoval*, is based on removing documents from the result list that are "suspected" to be spam. Specifically, we apply Waterloo's spam detector [Cormack et al. 2010] upon the initial QL-based ranking, top to bottom, and remove documents d with a spam score that is below 50 until 1000 documents are accumulated. The spam score for d is in $[0, 100]$, and it reflects the presumed percentage of documents in the entire ClueWeb English collection (category A, which includes around 500 million

Table VI. Comparison with Preretrieval Predictors

(a) Evaluation using the Optimal paradigm.

	TREC1-3		TREC4		TREC5	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
SumSCQ(TF.IDF)	−.112	−.059	.053	.055	−.029	−.042
AvgSCQ(TF.IDF)	.506	.318	.313	.234	.095	.080
MaxSCQ(TF.IDF)	.391	.309	.363	.232	.164	.039
SumVar(TF.IDF)	.021	.058	.153	.154	.049	.015
SumVar(LM)	−.104	−.050	.067	.063	−.038	−.016
AvgVar(TF.IDF)	.578	.390	.533	.391	.203	.136
AvgVar(LM)	.516	.322	.360	.283	.054	.085
MaxVar(TF.IDF)	.375	.335	.600	.440	.180	.110
MaxVar(LM)	.210	.227	.496	.382	.084	.044
SumIDF	−.047	.003	.101	.105	.053	.011
AvgIDF	.438	.303	.371	.234	.208	.132
MaxIDF	.258	.257	.288	.165	.269	.077
NQC	**.700**	**.465**	**.641**	**.494**	**.502**	**.340**

	ROBUST		WT10G		GOV2	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
SumSCQ(TF.IDF)	.053	.095	.149	.105	.227	.140
AvgSCQ(TF.IDF)	.292	.215	.300	.218	.308	.197
MaxSCQ(TF.IDF)	.358	.332	.452	.355	.375	.238
SumVar(TF.IDF)	.266	.269	.269	.189	.328	.219
SumVar(LM)	.043	.077	.190	.101	.161	.095
AvgVar(TF.IDF)	.428	.359	.282	.244	.360	.250
AvgVar(LM)	.262	.202	.273	.195	.309	.204
MaxVar(TF.IDF)	.443	.366	.406	.330	.372	.249
MaxVar(LM)	.286	.221	.379	.242	.311	.216
SumIDF	.287	.216	.162	.124	.281	.213
AvgIDF	.466	.288	.149	.181	.266	.195
MaxIDF	.486	.336	.134	.221	.294	.211
NQC	**.566**	**.419**	**.527**	**.331**	**.462**	**.362**

(b) Evaluation using the Cross-Validation paradigm. (Numbers in parentheses indicate the standard deviation of prediction quality.)

	TREC1-3		TREC4		TREC5	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
SumSCQ(TF.IDF)	−.113(.069)	−.060(.049)	.057(.142)	.053(.101)	−.033(.152)	−.041(.109)
AvgSCQ(TF.IDF)	.503(.069)	.318(.056)	.309(.155)	.238(.110)	.101(.100)	.080(.095)
MaxSCQ(TF.IDF)	.394(.067)	.310(.051)	.354(.111)	.228(.086)	.174(.107)	.047(.099)
SumVar(TF.IDF)	.021(.077)	.056(.053)	.153(.135)	.149(.098)	.046(.153)	.017(.104)
SumVar(LM)	−.104(.073)	−.051(.052)	.068(.153)	.058(.105)	−.041(.161)	−.014(.112)
AvgVar(TF.IDF)	.577(.059)	.389(.059)	.526(.113)	.392(.082)	.207(.156)	.141(.112)
AvgVar(LM)	.514(.068)	.321(.056)	.361(.127)	.280(.102)	.063(.139)	.090(.120)
MaxVar(TF.IDF)	.379(.080)	.333(.064)	**.596**(.104)	.442(.073)	.183(.169)	.119(.101)
MaxVar(LM)	.213(.075)	.226(.059)	.493(.099)	.382(.078)	.092(.144)	.049(.116)
SumIDF	−.046(.065)	.003(.048)	.105(.121)	.101(.086)	.044(.154)	.014(.108)
AvgIDF	.438(.070)	.304(.055)	.359(.177)	.236(.117)	.205(.144)	.135(.099)
MaxIDF	.261(.064)	.259(.047)	.280(.138)	.160(.110)	.250(.184)	.083(.100)
NQC	**.690**(.040)	**.455**(.044)	.586(.120)	**.456**(.086)	**.365**(.167)	**.272**(.094)

	ROBUST		WT10G		GOV2	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
SumSCQ(TF.IDF)	.055(.067)	.095(.045)	.152(.104)	.108(.078)	.227(.072)	.140(.051)
AvgSCQ(TF.IDF)	.292(.059)	.214(.038)	.297(.067)	.217(.058)	.306(.075)	.197(.056)
MaxSCQ(TF.IDF)	.361(.063)	.333(.032)	.453(.067)	.353(.066)	.374(.072)	.239(.059)
SumVar(TF.IDF)	.278(.084)	.270(.037)	.273(.094)	.191(.071)	.329(.071)	.221(.056)
SumVar(LM)	.045(.072)	.077(.047)	.191(.104)	.104(.082)	.162(.081)	.096(.057)
AvgVar(TF.IDF)	.439(.077)	.359(.033)	.282(.057)	.243(.052)	.360(.072)	.250(.051)
AvgVar(LM)	.264(.066)	.202(.039)	.283(.125)	197(.072)	.308(.074)	.202(.055)
MaxVar(TF.IDF)	.453(.079)	.367(.035)	.408(.075)	**.328**(.052)	.371(.077)	.251(.057)
MaxVar(LM)	.290(.070)	.223(.036)	.381(.119)	.240(.064)	.311(.081)	.215(.060)
SumIDF	.286(.071)	.216(.041)	.167(.109)	.129(.070)	.282(.070)	.213(.049)
AvgIDF	.465(.061)	.287(.033)	.157(.101)	.179(.067)	.266(.078)	.195(.055)
MaxIDF	.486(.061)	.336(.035)	.143(.110)	.219(.081)	.294(.078)	.212(.057)
NQC	**.535**(.055)	**.409**(.037)	**.487**(.119)	.286(.085)	**.416**(.066)	**.343**(.047)

documents) that are presumably "spammier" than d. This suspected spam removal approach is known to improve precision at top ranks, yet somewhat degrade MAP [Bendersky et al. 2011; Cormack et al. 2010; Lin et al. 2010]. We note that, while the result list upon which the postretrieval predictors are employed changes due to the removal of documents suspected as spam, this has no effect on the prediction of preretrieval predictors that use only the query- and corpus-based information. Indeed, preretrieval predictors do not depend on, or analyze, the result list. (We do not remove documents suspected as spam from the corpus, but only from the result list, and this is a postretrieval step.)

In addition to the postretrieval predictors that served as reference comparisons earlier, we also use the *ImpClarity* predictor [Hauff et al. 2008b], which was shown effective for large (and noisy) Web collections [Hauff et al. 2010]. This is the Clarity predictor when considering only (and all) terms that occur in less than 1% of documents in the corpus when computing Clarity. (Using 10% of the terms showed to yield lower prediction quality.)

The result list size, k, for all postretrieval predictors, and the overlap cuttoff for the QF predictor (m_{QF}), are set to a value in {5, 10, 50, 100, 150, 200, 300, 500, 700, 1000}. The number of terms used for constructing a relevance model for QF was set to 100 as at the preceding; for Clarity, the number of terms was set to 25, which yields better prediction quality. The prediction quality numbers, for both the Optimal and Cross-Validation paradigms of setting free-parameter values, are presented in Table VII.

We first analyze the Optimal case in Table VII. We can see that NQC outperforms both Clarity and ImpClarity in almost all relevant comparisons; so do WIG and QF. Furthermore, NQC is the best postretrieval predictor in terms of Pearson's correlation (for both QL and QL+SpamRemoval), and QF is the best postretrieval predictor in terms of Kendall's-τ. Overall, these results attest to the effectiveness of NQC as a postretrieval predictor for a large-scale and noisy Web collection. However, we can see that the preretrieval predictor SumVar(TF.IDF) is the most effective predictor when using the Optimal paradigm. Moreover, the SumIDF preretrieval predictor also improves over most postretrieval predictors in many cases. These findings are in line with those recently reported for ClueWeb [Hauff et al. 2010]. The superiority of preretrieval predictors to postretrieval predictors could potentially be attributed to the fact that the free-parameter values of the postretrieval predictors are set to the same value for all queries; thus, a per-query free-parameter setting might be called for when addressing noisy and large-scale collections such as ClueWeb.

For the Cross-Validation paradigm, the SumVar(TF.IDF) predictor is outperformed by QF for the QL+SpamRemoval case, while the reverse holds for QL. More generally, QF is the most effective postretrieval predictor for Cross-Validation. Yet, we hasten to point out that QF is highly inefficient for ClueWeb in that it requires running a relevance model (i.e., an expanded query composed of 100 terms) against a large-scale collection. Our NQC predictor posts in the Cross-Validation case prediction quality that is (much) better than that of Clarity and ImpClarity in almost all relevant comparisons. NQC is outperformed by WIG (and QF) for QL, but outperforms WIG for QL+SpamRemoval.

The varying relative prediction-quality patterns just mentioned between the Optimal and Cross-Validation cases could be attributed to the fact that only 50 queries are used for evaluation with ClueWeb. This means that for each partition of the query set, only 25 queries are used for training and the other 25 are used for testing. Thus, 25 queries might not be enough for setting the free-parameter values of the postretrieval predictors. Indeed, Table II showed that the postretrieval predictors had the largest prediction-quality variance for the TREC5 corpus which was the only one in Table II to

Table VII. Prediction Quality for ClueWeb

(a) Evaluation using the Optimal paradigm.

	QL		QL+SpamRemoval	
	P-ρ	K-τ	P-ρ	K-τ
SumSCQ(TF.IDF)	.587	.443	.606	.400
AvgSCQ(TF.IDF)	.329	.217	.239	.161
MaxSCQ(TF.IDF)	.528	.335	.442	.276
SumVar(TF.IDF)	**.631**	**.456**	.637	**.420**
SumVar(LM)	.367	.328	.423	.276
AvgVar(TF.IDF)	.365	.258	.293	.228
AvgVar(LM)	.222	.181	.232	.197
MaxVar(TF.IDF)	.590	.337	.528	.306
MaxVar(LM)	.323	.251	.357	.259
SumIDF	.618	.449	.620	**.420**
AvgIDF	.211	.228	.161	.189
MaxIDF	.409	.320	.357	.271
Clarity	.235	.140	.157	.124
ImpClarity	.427	.251	.333	.200
QF	.516	.371	.616	.425
WIG	.507	.340	.498	.338
NQC	.523	.233	**.638**	.336

(b) Evaluation using the Cross-Validation paradigm. (Numbers in parentheses indicate the standard deviation of prediction quality.)

	QL		QL+SpamRemoval	
	P-ρ	K-τ	P-ρ	K-τ
SumSCQ(TF.IDF)	.575(.031)	.441(.021)	.588(.032)	.397(.021)
AvgSCQ(TF.IDF)	.322(.030)	.213(.031)	.234(.029)	.156(.032)
MaxSCQ(TF.IDF)	.515(.027)	.337(.025)	.431(.026)	.275(.028)
SumVar(TF.IDF)	**.616**(.029)	.452(.025)	.616(.037)	.415(.018)
SumVar(LM)	.357(.036)	.328(.020)	.407(.032)	.279(.022)
AvgVar(TF.IDF)	.365(.038)	.253(.032)	.289(.038)	.223(.028)
AvgVar(LM)	.222(.029)	.178(.034)	.231(.030)	.196(.031)
MaxVar(TF.IDF)	.572(.036)	.336(.035)	.508(.041)	.306(.028)
MaxVar(LM)	.323(.028)	.253(.031)	.351(.032)	.261(.026)
SumIDF	.600(.035)	**.457**(.023)	.601(.037)	.424(.024)
AvgIDF	.230(.048)	.231(.028)	.174(.038)	.187(.030)
MaxIDF	.411(.047)	.323(.026)	.355(.040)	.269(.025)
Clarity	.264(.028)	.179(.029)	.182(.036)	.150(.029)
ImpClarity	.452(.048)	.278(.036)	.339(.044)	.223(.034)
QF	.584(.039)	.427(.029)	**.650**(.030)	**.471**(.032)
WIG	.503(.042)	.349(.033)	.486(.039)	.345(.028)
NQC	.481(.045)	.234(.024)	.627(.050)	.372(.032)

Best result in a column is boldfaced.

have only 50 queries as is the case for ClueWeb. Furthemore, as noted earlier, due to the noisy nature of the ClueWeb collection, a per-query free-parameter setting might be called for, a future venue we intend to explore.

4.2.8. Query Performance Prediction for TREC Runs. The discussion about the effectiveness of using the standard deviation of retrieval scores as a query performance predictor, which was presented in Section 3.3.1, was based on the premise that the scores reflect surface-level document-query similarities. The same premise, in fact, underlies the WIG predictor [Zhou 2007], which served as a reference comparison. We now turn to study the prediction quality of NQC and WIG when employed over *some* retrieval scores that are produced by a retrieval method that is not known to the predictor; these scores need not necessarily reflect document-query similarities.

Both NQC and WIG are based on normalization with the corpus retrieval score. (Refer back to Eqs. (3) and (6)). However, as the retrieval method here is not known, we do not employ this normalization. Consequently, NQC amounts to using the standard deviation of retrieval scores in the result list, and WIG amounts to using their mean.[13] We also experiment with NQC(norm) and WIG(norm) which are computed after the retrieval scores were sum-normalized as was the case for the Clarity(score) predictor from Section 4.2.5; that is, the sum of retrieval scores in the result list serves for normalization. As additional reference comparisons we use the Clarity variants used in Section 4.2.5, namely, Clarity(score) and Clarity(rank).

For benchmarks we use the ad hoc track of TREC3 and the robust track of TREC12. Our goal is to predict the effectiveness of runs submitted to these tracks. Specifically, each run is composed of retrieval results for (all) queries in the track. Our goal is to predict, *per query*, the effectiveness of the returned results in the run. Then, we measure prediction quality as before, that is, by using the prediction values for the queries for the run and the true AP values for these queries attained by the run. Hence, each run serves as a retrieval method and the relative effectiveness of the run's returned results over a set of queries is the goal of prediction. This evaluation should be differentiated from that of predicting the relative average effectiveness of runs (computed over all queries) with respect to each other.

We consider two different settings. The first, referred to as *Best Runs*, is predicting the effectiveness of a highly effective run. To that end, we employ the predictors upon each of the 5 best MAP-performing runs in a track, and report the average resultant prediction quality. The second setting, referred to as *Random Runs*, is predicting the effectiveness of a randomly selected run from all those submitted to a track. Specifically, we randomly select 5 runs and compute the average resultant prediction quality. To avoid irregularities rising from a single random selection of 5 runs, we repeat this (random) selection 30 times and report the resultant average prediction quality.

The only free parameter on which the postretrieval predictors depend here is the result-list size (k). For each of the postretrieval predictors tested (Clarity(score), Clarity(rank), WIG, WIG(norm), NQC, and NQC(norm)) we set k to a value in $\{5, 10, 50, 100, 150, 200, 300, 500, 700, 1000\}$ that yields the best prediction quality, that is, we use the Optimal paradigm so as to study the potential of the predictors in this setting. The number of terms used to construct the relevance model for Clarity(score) and Clarity(rank) was set to 100. We do not use the QF predictor here as the large number of runs, and the parametrization, calls for running hundreds (and even thousands) of relevance-model-based retrievals, which is computationally very expensive.

[13]We do not use the normalization with respect to query length in Eq. (6), as it did not result in improved prediction.

Table VIII. Employing the Predictors upon TREC Runs

	TREC3				TREC12			
	Best Runs		Random Runs		Best Runs		Random Runs	
	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
SumVar(TF.IDF)	−.194	−.131	−.143	−.110	.303	.196	.349	.263
SumVar(LM)	−.280	−.187	−.226	−.170	.201	.120	.228	.151
AvgVar(TF.IDF)	.392	.255	.390	.253	.335	.223	.378	.309
AvgVar(LM)	.393	.225	.410	.246	.298	.171	.328	.224
MaxVar(TF.IDF)	.114	.117	.128	.138	.338	.224	.382	.310
MaxVar(LM)	.029	.057	.053	.093	.261	.165	.298	.228
Clarity(score)	.494	.348	.533	.392	.400	.250	**.459**	.341
Clarity(rank)	.503	.353	**.588**	**.394**	.396	.249	.457	**.345**
WIG	.489	.364	.498	.367	.454	.350	.322	.241
WIG(norm)	.544	.392	.532	.382	.400	.284	.411	.323
NQC	**.623**	**.439**	.561	.388	**.594**	**.462**	.452	.325
NQC(norm)	.592	.409	.523	.368	.511	.355	.428	.319

The Optimal paradigm is used for evaluation. Best result in a column is boldfaced.

We also use the Var-based preretrieval predictors discussed before as reference comparisons. These predictors were the most effective among the preretrieval predictors considered. Furthermore, recall that the SumVar predictors bear conceptual connections with our NQC predictor.

The prediction values of preretrieval predictors, by definition, do not depend on the run (or more generally, the retrieval method) at hand, as preretrieval prediction only utilizes the query- and corpus-based information. Therefore, all runs, or more specifically, all different result lists for a query, as far as a preretrieval predictor is concerned, are of the same effectiveness. Yet, it is important to recall that we do not compare the effectiveness of runs, but rather the effectiveness of each run with respect to the different queries. Hence, preretrieval predictors could be thought of as predicting the "general difficulty" of the query for any retrieval method (run).

We can see in Table VIII that our NQC predictor posts the best prediction quality for the Best Runs setting for both TREC3 and TREC12. For the Random Runs setting, NQC is outperformed by either (or both) Clarity predictors; yet, NQC outperforms the WIG-based predictors and the preretrieval predictors for Random Runs. The improved prediction quality of NQC for the Best Runs with respect to the Random Runs can potentially be attributed to the fact that in the Random Runs setting more runs have retrieval scores that are meaningless. More generally, we note that the differences between prediction quality for Best Runs and Random Runs for the various predictors (both the preretrieval and postretrieval ones) are in line with some recent findings [Scholer and Garcia 2009] with regard to the variance of prediction quality (of Clarity and preretrieval predictors) across all TREC runs.

We can also see in Table VIII that normalizing retrieval scores hurts the prediction quality of NQC (compare NQC with NQC(norm)), while it improves the prediction quality of WIG (compare WIG with WIG(norm)) in most cases. Finally, we see that the prediction quality of the preretrieval predictors is quite below that of the postretrieval predictors in most cases. This finding can be attributed to the fact that the preretrieval predictors do not analyze the specific result list of a run per query, as opposed to postretrieval predictors. Specifically, this further supports the merits of our NQC predictor with respect to the Var-based predictors.

All in all, the results presented in Table VIII show that NQC is quite an effective predictor when employed over TREC runs. Some of these runs do not necessarily contain retrieval scores that reflect document-query surface-level similarities, or that are even meaningful.

5. SUMMARY

We presented a novel approach to predicting query performance that is based on measuring the standard deviation of retrieval scores among top-retrieved documents.

We argued that this standard deviation can be thought of as a proxy for measuring the amount (or more precisely, lack thereof) of query drift in the result list, the list of the documents most highly ranked. Our argument is based on the observation that the mean retrieval score in the result list is, for several retrieval methods that are based on document-query surface-level similarity, the retrieval score of a pseudo nonrelevant document that exhibits query drift; that is, a centroid representation of the list. Hence, high deviation of retrieval scores from the mean might reflect decreased query drift, and accordingly, improved query performance.

Through an extensive array of experiments, performed over a wide variety of TREC corpora, with a few highly effective postretrieval and preretrieval query performance predictors serving as reference comparisons, we demonstrated the effectiveness of our approach for query performance prediction. In addition, we studied the different factors that affect prediction quality (e.g., the size of the result list), demonstrated the importance of using standard deviation for measuring score dispersion, and showed that our prediction approach is effective with several different retrieval methods.

We also showed that our approach is quite useful for predicting the effectiveness of TREC runs, wherein retrieval scores do not necessarily reflect document-query similarities. We note, however, that as is the case for other prediction methods that rely on analysis of retrieval scores [Diaz 2007; Zhou and Croft 2007], if retrieval scores are not available (or are meaningless), then our approach cannot be employed.

For future work we intend to study whether our NQC predictor can help improve overall task performance for tasks wherein previously proposed predictors were applied, for example, the missing-content-analysis task [Yom-Tov et al. 2005]. It was recently shown [Cummins et al. 2011b], for example, that NQC and its variants can be used to select a query that is effective for representing a given information need from a set of candidate queries that are created based on a (TREC's) description of the need.

APPENDIX

A. EXAMPLES FOR RETRIEVAL-SCORES CURVES

In each row of Figure 4 we present the retrieval scores curve for "easy" and "difficult" queries, respectively. The queries are those in the ROBUST or WT10G benchmarks for which the effectiveness of the standard language-modeling approach—in terms of Average Precision (AP)—is the lowest and highest, respectively. (Refer to Section 4.1 for details of the benchmarks.)

We can see in Figure 4 that, in most cases, for easy queries the dispersion of retrieval scores with respect to the mean retrieval score, $\hat{\mu}$, is (much) higher than that for difficult queries. This dispersion is estimated by the proposed predictor, NQC.

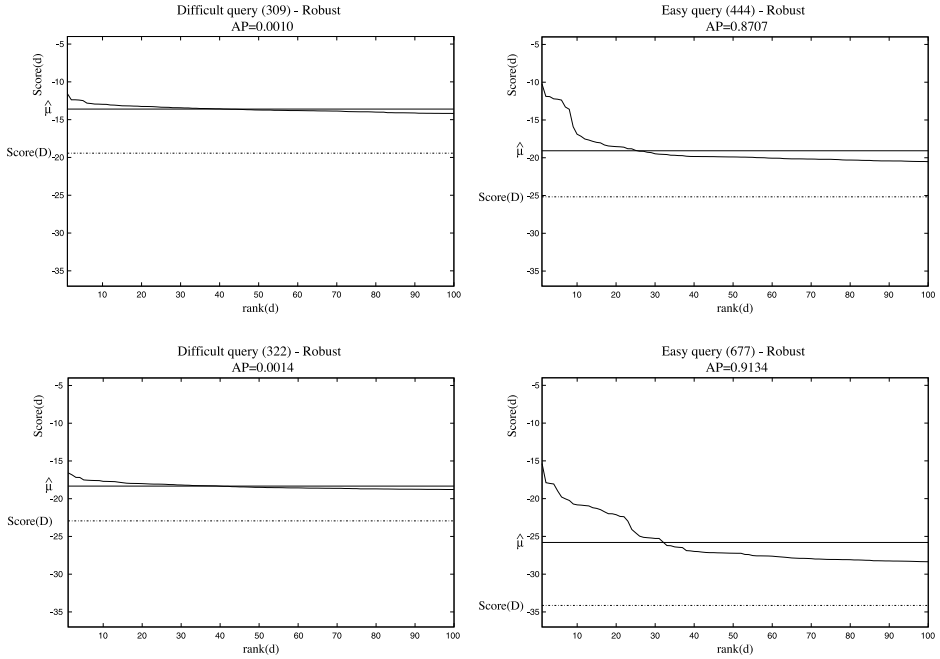
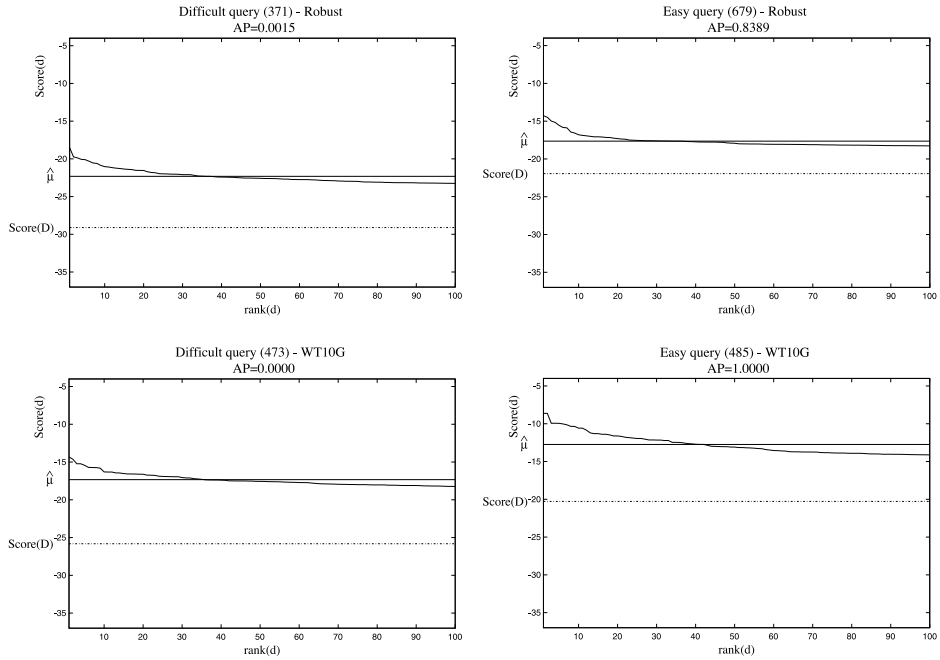


Fig. 4. The retrieval scores curves for "easy" and "difficult" queries taken from the ROBUST and WT10G benchmarks according to Average Precision (AP) performance of a language-model-based retrieval approach.



REFERENCES

ABDUL-JALEEL, N., ALLAN, J., CROFT, W. B., DIAZ, F., LARKEY, L., LI, X., SMUCKER, M. D., AND WADE, C. 2004. UMASS at trec 2004 – Novelty and hard. In *Proceedings of the Text Retrieval Conference (TREC-13)*.

AMATI, G., CARPINETO, C., AND ROMANO, G. 2004. Query difficulty, robustness and selective application of query expansion. In *Proceedings of the European Conference on IR Research (ECIR'04)*. 127–137.

ARAMPATZIS, A. AND ROBERTSON, S. 2011. Modeling score distributions in information retrieval. *Inf. Retriev. 14*, 1, 26–46.

ARAMPATZIS, A., KAMPS, J., AND ROBERTSON, S. 2009. Where to stop reading a ranked list? Threshold optimization using truncated score distributions. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 524–531.

ASLAM, J. A. AND PAVLU, V. 2007. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In *Proceedings of the European Conference on IR Research (ECIR'07)*. 198–209.

BENDERSKY, M., CROFT, W. B., AND DIAO, Y. 2011. Quality-Biased ranking of Web documents. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM'11)*. 95–104.

BERNSTEIN, Y., BILLERBECK, B., GARCIA, S., LESTER, N., SCHOLER, F., AND ZOBEL, J. 2005. RMIT university at trec 2006: Terabyte and robust track. In *Proceedings of the Text Retrieval Conference (TREC-14)*.

BUCKLEY, C. 2004. Why current IR engines fail. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. Poster. 584–585.

BUCKLEY, C., SALTON, G., ALLAN, J., AND SINGHAL, A. 1994. Automatic query expansion using SMART: TREC3. In *Proceedings of the Text Retrieval Conference (TREC-3)*. 69–80.

CARMEL, D. AND YOM-TOV, E. 2010. *Estimating the Query Difficulty for Information Retrieval*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool.

CARMEL, D., YOM-TOV, E., DARLOW, A., AND PELLEG, D. 2006. What makes a query difficult? In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 390–397.

CLARKE, C. L. A., CRASWELL, N., AND SOBOROFF, I. 2009. Overview of the trec 2009 Web track. In *Proceedings of the Text Retrieval Conference (TREC)*.

CORMACK, G. V., SMUCKER, M. D., AND CLARKE, C. L. A. 2010. Efficient and effective spam filtering and re-ranking for large Web datasets. CoRR abs/1004.5168.

CROFT, W. B. AND LAFFERTY, J. 2003. *Language Modeling for Information Retrieval*. Information Retrieval Book Series, Number 13. Kluwer.

CRONEN-TOWNSEND, S., ZHOU, Y., AND CROFT, W. B. 2002. Predicting query performance. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 299–306.

CRONEN-TOWNSEND, S., ZHOU, Y., AND CROFT, W. B. 2004. A language modeling framework for selective query expansion. Tech. rep. IR-338, Center for Intelligent Information Retrieval, University of Massachusetts.

CRONEN-TOWNSEND, S., ZHOU, Y., AND CROFT, W. B. 2006. Precision prediction based on ranked list coherence. *Inf. Retriev. 9*, 6, 723–755.

CUMMINS, R., JOSE, J. M., AND O'RIORDAN, C. 2011a. Improved query performance prediction using standard deviation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 1089–1090.

CUMMINS, R., LALMAS, M., O'RIORDAN, C., AND JOSE, J. M. 2011b. Navigating the user query space. In *Proceedings of the International Symposium on String Processing and Information Retrieval (SPIRE'11)*. 380–385.

DAI, K., KANOULAS, E., PAVLU, V., AND ASLAM, J. A. 2011. Variational bayes for modeling score distributions. *Inf. Retriev. 14*, 1, 47–67.

DIAZ, F. 2007. Performance prediction using spatial autocorrelation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 583–590.

FANG, H. AND ZHAI, C. 2005. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 480–487.

FANG, H., TAO, T., AND ZHAI, C. 2004. A formal study of information retrieval heuristics. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 49–56.

FUHR, N. 1992. Probabilistic models in information retrieval. *Comput. J. 35*, 3, 243–255.

HARMAN, D. 1992. Relevance feedback revisited. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 1–10.

HARMAN, D. AND BUCKLEY, C. 2004. The NRRC reliable information access (ria) workshop. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 528–529.

HAUFF, C., HIEMSTRA, D., AND DE JONG, F. 2008a. A survey of preretrieval query performance predictors. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'08)*. 1419–1420.

HAUFF, C., MURDOCK, V., AND BAEZA-YATES, R. 2008b. Improved query difficulty prediction for the Web. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'08)*. 439–448.

HAUFF, C., KELLY, D., AND AZZOPARDI, L. 2010. A comparison of user and system query performance predictions. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'10)*. 979–988.

HE, B. AND OUNIS, I. 2004. Inferring query performance using pre-retrieval predictors. In *Proceedings of the International Symposium on String Processing and Information Retrieval (SPIRE'04)*. 43–54.

KANOULAS, E., DAI, K., PAVLU, V., AND ASLAM, J. A. 2010. Score distribution models: Assumptions, intuition, and robustness to score manipulation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 242–249.

LAFFERTY, J. D. AND ZHAI, C. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 111–119.

LAVRENKO, V. AND CROFT, W. B. 2001. Relevance-based language models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 120–127.

LIN, J., METZLER, D., ELSAYED, T., AND WANG, L. 2010. Of ivory and smurfs: Loxodontan mapreduce experiments for Web search. In *Proceedings of the Text Retrieval Conference (TREC)*.

LIU, X. AND CROFT, W. B. 2008. Evaluating text representations for retrieval of the best group of documents. In *Proceedings of the European Conference on IR Research (ECIR'08)*. 454–462.

LV, Y. AND ZHAI, C. 2011. When documents are very long, bm25 fails! In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 1103–1104.

MANMATHA, R., RATH, T. M., AND FENG, F. 2001. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 267–275.

METZLER, D. AND CROFT, W. B. 2005. A Markov random field model for term dependencies. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 472–479.

METZLER, D. AND CROFT, W. B. 2007. Linear feature-based models for information retrieval. *Inf. Retriev. 10*, 3, 257–274.

MITRA, M., SINGHAL, A., AND BUCKLEY, C. 1998. Improving automatic query expansion. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 206–214.

MOTHE, J. AND TANGUY, L. 2005. Linguistic features to predict query difficulty. In *ACM SIGIR'05 Workshop on Predicting Query Difficulty - Methods and Applications*.

PÉREZ-IGLESIAS, J. AND ARAUJO, L. 2009. Ranking list dispersion as a query performance predictor. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval (ICTIR'09)*. 371–374.

PÉREZ-IGLESIAS, J. AND ARAUJO, L. 2010. Standard deviation as a query hardness estimator. In *Proceedings of the International Symposium on String Processing and Information Retrieval (SPIRE'10)*. 207–212.

PONTE, J. M. AND CROFT, W. B. 1998. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 275–281.

RAIBER, F. AND KURLAND, O. 2010. On identifying representative relevant documents. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'10)*. 99–108.

ROBERTSON, S. 2007. On score distributions and relevance. In *Proceedings of the European Conference on IR Research (ECIR'07)*. 40–51.

ROBERTSON, S. E., WALKER, S., JONES, S., HANCOCK-BEAULIEU, M., AND GATFORD, M. 1994. Okapi at trec-3. In *Proceedings of the Text Retrieval Conference (TREC)*.

ROCCHIO, J. J. 1971. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, G. Salton Ed., Prentice Hall, 313–323.

SALTON, J., WONG, A., AND YANG, C. S. 1975. A vector space model for automatic indexing. *Comm. ACM 18*, 11, 613–620.

SCHOLER, F. AND GARCIA, S. 2009. A case for improved evaluation of query difficulty prediction. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 640–641.

SCHOLER, F., WILLIAMS, H. E., AND TURPIN, A. 2004. Query association surrogates for Web search. *J. Am. Soc. Inf. Sci. Technol. 55*, 7, 637–650.

SEO, J. AND CROFT, W. B. 2010. Geometric representations for multiple documents. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 251–258.

SHTOK, A., KURLAND, O., AND CARMEL, D. 2009. Predicting query performance by query-drift estimation. In *Proceedings of the International Conference on Theory of Information Retrieval (ICTIR'09)*. 305–312.

SHTOK, A., KURLAND, O., AND CARMEL, D. 2010. Using statistical decision theory and relevance models for query performance prediction. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 259–266.

SONG, F. AND CROFT, W. B. 1999. A general language model for information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval* (Poster abstract). 279–280.

TERRA, E. L. AND WARREN, R. 2005. Poison pills: Harmful relevant documents in feedback. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'05)*. 319–320.

TOMLINSON, S. 2004. Robust, Web and terabyte retrieval with hummingbird search server at trec 2004. In *Proceedings of the Text Retrieval Conference (TREC-13)*.

TURTLE, H. R. AND CROFT, W. B. 1990. Inference networks for document retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 1–24.

VINAY, V., COX, I. J., MILIC-FRAYLING, N., AND WOOD, K. R. 2006. On ranking the effectiveness of searches. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 398–404.

VOORHEES, E. M. 2004. Overview of the trec 2004 robust retrieval track. In *Proceedings of the Text Retrieval Conference (TREC-13)*.

YOM-TOV, E., FINE, S., CARMEL, D., AND DARLOW, A. 2005. Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 512–519.

ZHAI, C. AND LAFFERTY, J. D. 2001a. Model-Based feedback in the language modeling approach to information retrieval. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'01)*. 403–410.

ZHAI, C. AND LAFFERTY, J. D. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 334–342.

ZHAO, Y., SCHOLER, F., AND TSEGAY, Y. 2008. Effective preretrieval query performance prediction using similarity and variability evidence. In *Proceedings of the European Conference on IR Research (ECIR'08)*. 52–64.

ZHOU, Y. 2007. Retrieval performance prediction and document quality. Ph.D. thesis, University of Massachusetts Amherst.

ZHOU, Y. AND CROFT, W. B. 2006. Ranking robustness: A novel framework to predict query performance. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'06)*. 567–574.

ZHOU, Y. AND CROFT, W. B. 2007. Query performance prediction in Web search environments. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 543–550.

Received March 2011; revised January 2012; accepted February 2012