Using Document-Quality Measures to Predict Web-Search Effectiveness

Fiana Raiber and Oren Kurland

Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel
fiana@tx.technion.ac.il, kurland@ie.technion.ac.il

Abstract. The query-performance prediction task is estimating retrieval effectiveness in the absence of relevance judgments. The task becomes highly challenging over the Web due to, among other reasons, the effect of low quality (e.g., spam) documents on retrieval performance. To address this challenge, we present a novel prediction approach that utilizes query-independent document-quality measures. While using these measures was shown to improve Web-retrieval effectiveness, this is the first study demonstrating the clear merits of using them for query-performance prediction. Evaluation performed with large scale Web collections shows that our methods post prediction quality that often surpasses that of state-of-the-art predictors, including those devised specifically for Web retrieval.

Keywords: query-performance prediction, Web retrieval.

1 Introduction

The effectiveness of retrieval systems can radically vary across queries [23]. Hence, there has been a large body of work on devising *query-performance* prediction methods [5]. These methods estimate the effectiveness of a search performed in response to a query when no relevance judgments are available. In this paper, we focus on devising improved query-performance predictors for Web search.

Query-performance variability on the Web can be significantly higher than that for "clean", non-adversarial, retrieval settings (e.g., newswire corpora). For example, some queries might be the target of search engine optimization (SEO) efforts. Hence, rankings for these queries can be severely biased by SEO attempts. Furthermore, there are many pages on the Web which are not spam yet bear very little content. These pages can still be ranked high in result lists due to query-terms occurrence; e.g., pages containing tables that include only keywords which are not informative in their own right [2].

Given these observations, we devise a query-performance prediction approach that utilizes query-independent *document quality measures*. These measures include the probability that the document is not spam as determined by a spam classifier; its PageRank score [4]; and, estimates of the "richness" of the language used in the page. Thus, while using document-quality measures was shown to

P. Serdyukov et al. (Eds.): ECIR 2013, LNCS 7814, pp. 134–145, 2013.

improve search effectiveness [29,28,2], we study their utilization for the performance prediction task. Specifically, we devise predictors that are based on the following premise. All other factors being equal, retrieved lists that contain documents of high quality should be assigned with relatively higher effectiveness estimates than those assigned to lists containing documents of lower quality.

The experimental results we present attest to the merits of our prediction approach. For example, the prediction quality is often substantially better than that of state-of-the-art predictors including those devised specifically for Web retrieval.

2 Related Work

Post-retrieval query-performance predictors use information induced from the result list of the most highly ranked documents [5]. Our prediction methods integrate several state-of-the-art post-retrieval predictors with document-quality estimates. The resulting prediction quality substantially transcends that of using the post-retrieval predictor alone. It was recently shown [12] that some pre-retrieval predictors [25], which only use the query and corpus-based information, can outperform state-of-the-art post-retrieval predictors over a large scale Web corpus; namely, Category B of ClueWeb [7]. We show that our methods outperform these pre-retrieval predictors for both categories (A and B) of ClueWeb.

Variants of the Clarity predictor [9] were suggested for query-performance prediction on the Web [13]. We demonstrate the substantial prediction-quality merits of our method with respect to one such effective variant.

The coherence and dispersion of the result list, quantified using several measures, were suggested as query-independent indicators for retrieval effectiveness [6,22,14]. We study the resultant prediction quality of using a list coherence indicator in our approach.

The dissimilarity between the result list and a list induced using *only* query-independent document quality measures was suggested as a query performance predictor [3]. Experiments we conducted with this approach (actual results are omitted as they convey no additional insight) showed that it is by far less effective than the prediction method we propose here that uses document quality measures in a completely different way. This finding resonates with the arguments made in [3] with respect to other post-retrieval predictors outperforming this past approach [3]. Indeed, these predictors are shown here to be substantially outperformed by our approach.

3 Prediction Approach

Suppose that a retrieval method is employed upon a corpus of documents $\mathcal{D}$ in response to query q. The goal of query-performance predictors is estimating the effectiveness of the resultant ranking when no relevance judgments are available. As in previous work on post-retrieval prediction [5], the prediction methods we present analyze $\mathcal{D}_q^{[k]}$ — the result list of the k most highly ranked documents.

That is, the goal is predicting the effectiveness of $\mathcal{D}_q^{[k]}$ with respect to the information need expressed by q.

The prediction methods that we devise utilize query independent *document-quality* measures. These measures include, for example, the PageRank score [4] of a document and the probability that the document is not spam as estimated by a spam classifier. In Section 3.2 we present the quality measures we use. The fundamental premise is that the higher the quality of documents in $\mathcal{D}_q^{[k]}$, although measured in a query-independent way, the higher $\mathcal{D}_q^{[k]}$'s predicted effectiveness should be. The estimated quality of $\mathcal{D}_q^{[k]}$, denoted $F(\mathcal{D}_q^{[k]})$, is an aggregate of the document-quality estimates $f(d)$ for documents d in $\mathcal{D}_q^{[k]}$.

3.1 The List-Quality-Based Approach

Let $\mathcal{P}_{basic}$ be some post-retrieval predictor applied upon the result list $\mathcal{D}_q^{[k]}$. In Section 3.3 we discuss several such previously proposed predictors. We use $\mathcal{P}_{basic}(q; \mathcal{D}_q^{[k]})$ to denote the prediction value. This value reflects the likelihood that the result list $\mathcal{D}_q^{[k]}$ satisfies the information need expressed by q.

To motivate the use of document-quality measures as an additional source of evidence for query-performance prediction, we consider the following case in point. Many post-retrieval predictors are based on an analysis of the retrieval scores in $\mathcal{D}_q^{[k]}$ [21,10,27,18]. More specifically, the underlying premise of several of these predictors [10,27,18], which might be implicit, is that retrieval scores reflect surface-level document-query similarities. Over the Web, these similarities can be severely affected by search engine optimization (SEO) efforts; for example, keyword stuffing [11]. Specifically, some of the documents in $\mathcal{D}_q^{[k]}$ that exhibit high query similarity can even be complete spam [17,8]. Other documents, for example, can contain tables with the query terms [2], but with no additional significant content that can satisfy any information needs. Thus, prediction based solely on retrieval scores, specifically, those reflecting document-query similarities, can fall short.

The observations just stated give rise to the premise posted above that the prediction value for a result list that contains high quality documents should be relatively high. Now, the query-independent result-list quality measure, $F(\mathcal{D}_q^{[k]})$, represents the presumed quality of documents in $\mathcal{D}_q^{[k]}$. Thus, we can use $F(\mathcal{D}_q^{[k]})$ so as to bias the prediction in favor of lists containing high quality documents. Specifically, we define the list-quality-based predictor, **LQ**, as:

$$\mathcal{P}_{LQ}(q; \mathcal{D}_q^{[k]}) \stackrel{def}{=} \mathcal{P}_{basic}(q; \mathcal{D}_q^{[k]}) F(\mathcal{D}_q^{[k]}) \; . \tag{1}$$

LQ considers $\mathcal{D}_q^{[k]}$ to be effective to the extent that (i) it is estimated to be effective by the basic predictor $\mathcal{P}_{basic}$; and, (ii) it contains documents that are estimated, using query-independent measures, to be of high quality.

3.2 Quality Measures

Spam Spam documents are prevalent on the Web and can quite degrade retrieval effectiveness [17,2,8]. Let $f_{NS}(d)$ denote the score assigned by a spam classifier to d, which reflects the likelihood that d is *not* spam. If $f_{NS}(d) > t_{spam}$ then d is considered non-spam; otherwise d is deemed spam; t_{spam} is a threshold. The set of documents in $\mathcal{D}_q^{[k]}$ considered non-spam is denoted $\mathcal{D}_{NS}$; $m \stackrel{def}{=} |\mathcal{D}_{NS}|$ is their number.

The following is a key observation which is novel to this study. We can apply retrieval effectiveness measures upon $\mathcal{D}_q^{[k]}$ by considering documents deemed non-spam as "relevant" and documents deemed spam as "non-relevant". Specifically, we adapt, in spirit, the average precision measure. Yet, we do not apply normalization with respect to the number of "relevant" (i.e., non-spam) documents in the corpus, as the more non-spam documents there are in $\mathcal{D}_q^{[k]}$, and the higher these are ranked, the higher $\mathcal{D}_q^{[k]}$'s predicted effectiveness should be.

Formally, let δ be Kronecker's delta function; that is, $\delta[x] \stackrel{def}{=} 1$ if the statement x holds, and $\delta[x] \stackrel{def}{=} 0$ otherwise; ϵ (= 0.00001) is a smoothing parameter. The spam-based quality estimate, denoted **NS**, assigned to $\mathcal{D}_q^{[k]}$ is:

$$F_{NS}(\mathcal{D}_q^{[k]}) \stackrel{def}{=} \sum_{i=1}^{k} \frac{\delta[d_i \in \mathcal{D}_{NS}]}{i} \sum_{j=1}^{i} \delta[d_j \in \mathcal{D}_{NS}] + \epsilon \; . \tag{2}$$

The additional document-quality measures we discuss below are used in the LQ predictor for estimating $\mathcal{D}_q^{[k]}$'s quality, $F(\mathcal{D}_q^{[k]})$. The measures are computed only for documents $d \in \mathcal{D}_{NS}$ — i.e., documents considered as non-spam. Indeed, experiments reveal — actual numbers are omitted as they convey no additional insight — that this practice yields better prediction quality than that of computing the measures upon all the documents in $\mathcal{D}_q^{[k]}$.

Stopwords. The number of stopwords in a Web page was recently proposed as a document quality measure [2]. Increased number is assumed to imply "rich" use of language, and consequently, high quality. Using this quality measure was shown to improve retrieval effectiveness [2]. We use $f_{SW1}(d)$ to denote the fraction of the stopwords on the INQUERY list that appear in d. $f_{SW2}(d)$ denotes the ratio of the number of stopwords to non-stopwords in d.

Document Entropy. The next quality measure we consider is the entropy of the term distribution in a document [15,2]. High entropy potentially implies to the richness of language used in the document. Low entropy indicates that the term distribution is concentrated around a few terms; hence, potentially less information needs can be satisfied by the document. Formally,

$$f_{Ent}(d) \stackrel{def}{=} -\sum_{w} p(w|d) \log p(w|d) \; ;$$

$p(w|d)$ is the probability assigned to term w by a language model induced from d. Language-model induction details are provided in Section 4.1.

Inter-document Similarities. The coherence of a result list, measured for example by using inter-document similarities [6,22], was argued to be correlated with retrieval effectiveness. Accordingly, we define the **IDS** quality measure for document d as its average similarity with (non-spam) documents in $\mathcal{D}_{NS}$:

$$f_{IDS}(d) \stackrel{def}{=} \frac{1}{m+1} \sum_{d_i \in \mathcal{D}_{NS}} sim(d, d_i) \; ;$$

$sim(d, d_i) \stackrel{def}{=} \exp\left(\sum_w p(w|d) \log p(w|d_i)\right)$ is the exponent of the negative cross entropy between d's and d_i's language models. We use $m+1$ to avoid zero division in case $\mathcal{D}_{NS}$ is empty.

PageRank. The document quality measures presented above are all based solely on the document content. (This also holds for the spam classifier which we discuss in Section 4.1.) The PageRank score of document d [4], which we also consider as a quality measure, $f_{PR}(d)$, is based on hyperlink information.

From Document Quality to List Quality. The LQ predictor uses an estimate $F(\mathcal{D}_q^{[k]})$ for the quality of the result list $\mathcal{D}_q^{[k]}$. One such estimate is the NS measure defined above. Another type of estimate is based on aggregating the per-document quality values assigned by the measures SW1, SW2, Ent, IDS and PR to documents in $\mathcal{D}_{NS}$ — the set of (presumably) non-spam documents in $\mathcal{D}_q^{[k]}$. To aggregate the per-document quality values, so as to form the list-based quality estimate $F(\mathcal{D}_q^{[k]})$, we simply use their (ϵ-smoothed) sum: $\sum_{d \in \mathcal{D}_{NS}} f(d) + \epsilon$. Finally, we also study the **Combine** measure (**Cmb** in short) that integrates the different list-based *quality measures* $F(\mathcal{D}_q^{[k]})$ by multiplying their values without normalization.

3.3 Basic Predictors

We use a wide variety of basic predictors. Many of these were shown to post state-of-the-art prediction quality [5]. None of these predictors utilizes document-quality measures.

The **Clarity** [9] predictor (**Clr** in short) measures the KL divergence between a (relevance) language model induced from $\mathcal{D}_q^{[k]}$ and that induced from the corpus. The higher the divergence, the more focused $\mathcal{D}_q^{[k]}$ is assumed to be; consequently, retrieval is presumed to be more effective. We also use a variant of Clr, **IClr**, which was proposed for noisy Web settings [13]; specifically, only the terms that appear in less than p% of all documents in the corpus are considered when inducing $\mathcal{D}_q^{[k]}$'s language model; p is a free parameter.

The **WIG** predictor [27] is based on the premise that the higher the difference between the mean retrieval score in $\mathcal{D}_q^{[k]}$, $\mu \stackrel{def}{=} \frac{1}{k}\sum_{d_i \in \mathcal{D}_q^{[k]}} Score(d_i; q)$, and the retrieval score of the corpus[1], the more effective the retrieval:

$$\mathcal{P}_{WIG}(q; \mathcal{D}_q^{[k]}) \stackrel{def}{=} \frac{1}{\sqrt{|q|}}(\mu - Score(\mathcal{D}; q)) \ ;$$

$Score(x; q)$ is the retrieval score assigned to x in response to q; query-length ($|q|$) normalization serves to ensure inter-query compatibility of prediction values.

The **NQC** predictor [18] measures the standard deviation of retrieval scores in $\mathcal{D}_q^{[k]}$:

$$\mathcal{P}_{NQC}(q; \mathcal{D}_q^{[k]}) \stackrel{def}{=} \frac{\sqrt{\frac{1}{k}\sum_{d_i \in \mathcal{D}_q^{[k]}}(Score(d_i; q) - \mu)^2}}{|Score(\mathcal{D}; q)|} \ ;$$

normalization with respect to the corpus retrieval score serves to ensure inter-query compatibility of prediction values. It was argued [18] that increased standard deviation of retrieval scores is correlated with potentially decreased *query drift* in $\mathcal{D}_q^{[k]}$, and hence, with improved retrieval.

The query feedback (**QF**) [27] predictor measures the number of documents that are among the ν_{qf} most highly ranked both in $\mathcal{D}_q^{[k]}$ and by a ranking induced over the corpus using a relevance language model induced from $\mathcal{D}_q^{[k]}$; ν_{qf} is a free parameter. The higher the number of shared documents, the less "noise" $\mathcal{D}_q^{[k]}$ is considered to exhibit; accordingly, the retrieval is presumed to be more effective.

The **UEF** prediction framework [19] is based on the following principle. A relevance language model is induced from $\mathcal{D}_q^{[k]}$ and is used to re-rank it. Then, the similarity between $\mathcal{D}_q^{[k]}$ and its re-ranking is scaled by the value $\mathcal{P}_{basic}(q; \mathcal{D}_q^{[k]})$ assigned by a basic predictor $\mathcal{P}$ to $\mathcal{D}_q^{[k]}$. The basic prediction value quantifies the confidence in the quality of the relevance model induced from $\mathcal{D}_q^{[k]}$; and, the similarity between $\mathcal{D}_q^{[k]}$ and its re-ranking (measured using Pearson correlation), scaled by this confidence level, presumably attests to $\mathcal{D}_q^{[k]}$'s effectiveness. To instantiate a specific predictor from the UEF framework, we used QF for the basic predictor $\mathcal{P}$, because this resulted in better prediction quality than that of using the other basic predictors to this end; namely, Clr, IClr, NQC, and WIG.

4 Evaluation

4.1 Experimental Setup

Our experiments were conducted with the ClueWeb collection [7]. We used (i) the entire English subset, which includes about 500 million pages (Category A); and, (ii) the first 50 million English pages (Category B). For each category three query

[1] We represent the corpus by the concatenation of all documents it contains. The order of concatenation has no effect as the retrieval model we use utilizes unigram language models that assume term independence. See Section 4.1 for details.

sets are used: 1-50 from TREC 2009, 51-100 from TREC 2010, and 101-150 from TREC 2011. Thus, we have 6 basic experimental settings, denoted **ClueWeb-09A**, **ClueWeb-10A**, **ClueWeb-11A**, **ClueWeb-09B**, **ClueWeb-10B**, and **ClueWeb-11B**, with A and B indicating the ClueWeb category used, and 09/10/11 indicating the TREC's query set. We applied Krovetz stemming and stopword removal upon queries (but not over documents) using the INQUERY list, via the Indri toolkit (`www.lemurproject.org`), which was also used for retrieval.

The first document result list for which we predict effectiveness using the methods we proposed is denoted **QL**. The list is retrieved using the query likelihood (QL) model [20] which served as the retrieval method in many reports on query-performance prediction [9,26,27,10,13,18,12,19]. Specifically, let $p(w|d)$ be the probability assigned to term w by a language model induced from d; then, $Score(d;q) \stackrel{def}{=} \log \prod_{q_i \in q} p(q_i|d)$, where q_i is a query term, is used to induce the corpus ranking. We used Dirichlet-smoothed unigram document language models [24] with the smoothing parameter μ set to 1000. The non-smoothed maximum likelihood estimate, which amounts to setting $\mu = 0$, was used to measure document entropy. To compute the IDS measure introduced in Section 3.2, the inter-document similarity estimate ($sim(d_i, d_j)$) is defined as the exponent of the negative cross entropy between the non-smoothed language model of d_i and the Dirichlet-smoothed language model (with $\mu = 1000$) of d_j.

To (further) examine the effect of spam documents on our prediction methods, we use an additional document result list created as follows. Documents suspected as spam, based on using the t_{spam} threshold specified in Section 3.2, are removed from the QL ranking, top to bottom, until 1000 (presumably) non-spam documents are accumulated [2,8]. These (presumably) non-spam documents constitute our second result list, **QL-SR**. Thus, we have two retrieved lists (QL and QL-SR) that serve as the basis for applying prediction; QL-SR presumably contains much fewer spam documents than QL. Accordingly, together with the 6 basic experimental settings described above, we have 12 settings using which we evaluate the prediction methods.

Prediction quality is measured, as is common [5], using Pearson's correlation coefficient between scores assigned to queries by a prediction method and the actual average precision, at cutoff 1000, measured using relevance judgments.

We employ a train-test approach to set the free-parameter values of the prediction methods. Specifically, we apply the following procedure, independently, for the query sets in TREC 2009, TREC 2010 and TREC 2011. A query set is randomly split into two equal-sized sets. Each of the two sets serves once as the train fold and once as the test fold. Thus, each split results in two train sets used to serve two test sets. The free-parameter values that optimize prediction quality over the train set are applied on the test set. The prediction quality of a predictor for the split is its average prediction quality over the two test sets. We repeat the splitting process 30 times and report the average prediction quality. The same splits are used for Category A and Category B. Statistically significant differences of prediction quality are determined using the two-tailed paired t-test with a p-value of 0.05 computed with respect to the 30 splits.

The number of most highly ranked documents in the result list, k, considered by the various predictors, is set to a value in $\{5, 10, 25, 50, 100, 250, 500, 1000\}$. To construct a relevance model (RM1) [16], which is used in Clr, QF and UEF, the number of terms and the Dirichlet smoothing parameter of the documents language models are selected from $\{50, 100\}$ and $\{0, 1000\}$, respectively. For the IClr predictor, we set p to a value in $\{1, 10\}$. To construct RM1 for IClr the Dirichlet smoothing parameter is set to a value in $\{0, 1000\}$, as at the above. The value of ν_{qf}, the QF overlap cutoff, is selected from $\{5, 10, 25, 50, 100\}$. For the PR quality measure, we use the publicly available un-normalized PR scores that were computed over the hyperlink graph of ClueWeb Category A.[2]

We used Waterloo's spam classifier [8] for (i) constructing the QL-SR result list, and (ii) determining the set $\mathcal{D}_{NS}$ of non-spam documents upon which the various query-independent quality measures are computed. Each document is assigned a score in $[0, 100]$ by the classifier. The score represents the percentage of documents in the entire ClueWeb English collection (Category A) that are "spammier" than the document at hand. To create the QL-SR result list, we set $t_{spam} = 50$ for Category B and $t_{spam} = 70$ for Category A, based on previous recommendations [8]. To create the non-spam document set $\mathcal{D}_{NS}$, used for computing list-quality measures, the t_{spam} threshold is set to a value in $\{0, 10, 20, ..., 90\}$ for the QL result list; for the QL-SR list, t_{spam} is set to a value in $\{50, 60, 70, 80, 90\}$ for Category B and in $\{70, 80, 90\}$ for Category A.

Pre-retrieval Predictors. In what follows we use highly effective pre-retrieval predictors as reference comparisons to our methods. Some of these were recently shown to outperform effective post-retrieval predictors for ClueWeb-09B [12].

The SCQ predictors measure the similarity between the query and the collection using the term frequency (TF) and inverse document frequency (IDF) of a query term in the collection [25]. The Var predictors measure the variance of the TF.IDF values of a query term across the documents in the corpus it appears in [25]. The IDF predictors use the IDF value of a query term [9]. For each of the three types of predictors just mentioned, we analyzed the prediction quality when using the sum, average and maximum prediction value(s) assigned to the query terms. Using the sum of the per query-term values yields, in general, the best prediction quality. The resultant predictors, denoted **SumSCQ**, **SumVar** and **SumIDF**, serve for reference comparison below. The train-test approach described above is used for evaluating the prediction quality of these predictors, as is the case for all other prediction methods we study. Yet, we note that the pre-retrieval predictors do not incorporate free parameters. Integrating pre-retrieval predictors with quality measures is a future venue to explore.

4.2 Experimental Results

The prediction quality numbers are presented in Table 1. We use LQ(X;Y) to indicate that the X quality measure is integrated with the Y basic predictor in

[2] `http://boston.lti.cs.cmu.edu/clueweb09/wiki/tiki-index.php?page=PageRank`

Table 1. Prediction quality of the LQ approach. The best result per an experimental setting and a basic post-retrieval predictor is boldfaced. The best result in an experimental setting (column) is underlined. '⋆' marks a statistically significant difference with using the basic predictor alone.

	ClueWeb-09A		ClueWeb-10A		ClueWeb-11A		ClueWeb-09B		ClueWeb-10B		ClueWeb-11B	
	QL	QL-SR	QL	QL-SR	QL	QL-SR	QL	QL-SR	QL	QL-SR	QL	QL-SR
SumSCQ	.573	.552	.449	.339	.329	.187	.494	.524	.365	.359	.110	.076
SumVar	.578	.554	.462	.350	.290	.169	.532	.547	.367	.381	.147	.097
SumIDF	.624	.630	.478	.365	.389	.234	.569	.579	.384	.385	.188	.152
NS	.699	.624	.755	.615	.522	.275	.594	.554	.697	.547	.483	.279
SW2	.692	.592	.846	.620	.651	.542	.569	.556	.747	.592	.625	.457
Cmb	.744	.644	.755	.733	.583	.376	.580	.602	.763	.703	.591	.332
Clr	−.058	−.222	−.342	−.104	−.124	−.199	.034	−.034	−.287	−.183	−.112	−.111
LQ(NS;Clr)	.718⋆	.575⋆	.740⋆	.580⋆	.476⋆	.076⋆	.610⋆	.513⋆	.673⋆	.530⋆	.462⋆	.159⋆
LQ(SW2;Clr)	.728⋆	.581⋆	**.829**⋆	.650⋆	**.663**⋆	**.456**⋆	**.615**⋆	.602⋆	.733⋆	.615⋆	**.616**⋆	**.481**⋆
LQ(Cmb;Clr)	**.748**⋆	**.642**⋆	.744⋆	**.730**⋆	.575⋆	.357⋆	.589⋆	**.603**⋆	**.743**⋆	**.692**⋆	.590⋆	.331⋆
IClr	.299	.084	.105	.042	.068	.036	.402	.330	.050	.018	.278	.153
LQ(NS;IClr)	.732⋆	.614⋆	.744⋆	.606⋆	.526⋆	.212⋆	.621⋆	.594⋆	.688⋆	.549⋆	.523⋆	.291⋆
LQ(SW2;IClr)	.743⋆	.618⋆	**.835**⋆	.642⋆	**.660**⋆	**.501**⋆	**.629**⋆	**.624**⋆	.743⋆	.629⋆	**.647**⋆	**.491**⋆
LQ(Cmb;IClr)	**.748**⋆	**.649**⋆	.750⋆	**.728**⋆	.582⋆	.373⋆	.582⋆	.619⋆	**.760**⋆	**.704**⋆	.596⋆	.347⋆
QF	.431	.564	.374	.705	.386	.516	.637	.755	.617	.609	.572	.552
LQ(NS;QF)	.783⋆	.788⋆	.848⋆	.783⋆	.612⋆	.522⋆	.708⋆	.800⋆	.768⋆	.690⋆	.684⋆	.607⋆
LQ(SW2;QF)	.773⋆	**.811**⋆	**.890**⋆	**.796**⋆	**.711**⋆	**.658**⋆	**.711**⋆	**.809**⋆	**.831**⋆	**.758**⋆	**.738**⋆	**.682**⋆
LQ(Cmb;QF)	**.798**⋆	.745⋆	.845⋆	.780⋆	.651⋆	.532	.641	.702⋆	.795⋆	.726⋆	.675⋆	.534
WIG	.348	.424	.281	.374	.231	.320	.473	.448	.339	.375	.293	.317
LQ(NS;WIG)	.710⋆	**.663**⋆	.743⋆	.645⋆	.558⋆	.370⋆	**.623**⋆	**.620**⋆	.691⋆	.583⋆	.550⋆	.411⋆
LQ(SW2;WIG)	.723⋆	.661⋆	**.831**⋆	.635⋆	**.689**⋆	**.617**⋆	.610⋆	.615⋆	.746⋆	.610⋆	**.682**⋆	**.566**⋆
LQ(Cmb;WIG)	**.755**⋆	.660⋆	.755⋆	**.742**⋆	.604⋆	.423⋆	.581⋆	.614⋆	**.755**⋆	**.710**⋆	.613⋆	.398⋆
NQC	.262	.388	.238	.261	.251	.421	.509	**.658**	.271	.367	.360	.504
LQ(NS;NQC)	.708⋆	**.697**⋆	.802⋆	.550⋆	.508⋆	.439⋆	**.588**⋆	.657	.668⋆	.559⋆	.486⋆	.549⋆
LQ(SW2;NQC)	.685⋆	.654⋆	**.827**⋆	.557⋆	**.637**⋆	**.527**⋆	.551⋆	.649	.659⋆	.645⋆	.500⋆	**.637**⋆
LQ(Cmb;NQC)	**.767**⋆	.671⋆	.780⋆	**.737**⋆	.554⋆	.394⋆	.587⋆	.595⋆	**.735**⋆	**.687**⋆	**.553**⋆	.451⋆
UEF	.467	.618	.426	.717	.395	.572	.645	.792	.614	.642	.600	.576
LQ(NS;UEF)	.758⋆	**.784**⋆	.787⋆	.773⋆	.613⋆	.575⋆	**.702**⋆	**.805**⋆	.704⋆	.704⋆	**.689**⋆	.632⋆
LQ(SW2;UEF)	.720⋆	.765⋆	.793⋆	**.797**⋆	.629⋆	**.625**⋆	.689⋆	.801⋆	.728⋆	**.741**⋆	.669⋆	**.643**⋆
LQ(Cmb;UEF)	**.784**⋆	.758⋆	**.816**⋆	.783⋆	**.650**⋆	.585	.659	.719⋆	**.738**⋆	.719⋆	.669⋆	.593⋆

the LQ method. We focus here on the NS and SW2 quality measures as these turn out to be among the most effective we considered as we show later on. We also present the prediction quality for Cmb (Combine), which integrates all the quality measures. An additional reference comparison for LQ that we consider is a predictor that uses *only* the query-independent list quality measure (or Cmb) without integrating it with a basic predictor as in LQ.

Our main observation based on Table 1 is that the LQ predictors yield prediction quality that transcends in a majority of the relevant comparisons — often substantially and statistically significantly — that of the basic predictors when used alone. Furthermore, the LQ predictors outperform the pre-retrieval predictors; many of these improvements are quite substantial.

Interestingly, using the quality measures alone yields prediction quality that is in most cases superior to that of the pre-retrieval predictors and to that of the post-retrieval predictors when used alone. This finding further attests to the substantial merits of using document-quality measures for query-performance

Table 2. Comparing the prediction quality of using the various document quality measures in LQ. The best result per experimental setting and a basic predictor is boldfaced, and the best result per experimental setting (column) is underlined. '⋆' marks a statistically significant difference with using the basic predictor alone.

	ClueWeb-09A		ClueWeb-10A		ClueWeb-11A		ClueWeb-09B		ClueWeb-10B		ClueWeb-11B	
	QL	QL-SR	QL	QL-SR	QL	QL-SR	QL	QL-SR	QL	QL-SR	QL	QL-SR
QF	.431	.564	.374	.705	.386	.516	.637	.755	.617	.609	.572	.552
LQ(NS;QF)	.783⋆	.788⋆	.848⋆	.783⋆	.612⋆	.522⋆	.708⋆	.800⋆	.768⋆	.690⋆	.684⋆	.607⋆
LQ(SW1;QF)	.768⋆	.779⋆	.874⋆	**.796**⋆	.699⋆	.621⋆	**.730**⋆	.797⋆	.806⋆	.734⋆	.717⋆	.643⋆
LQ(SW2;QF)	.773⋆	**.811**⋆	**.890**⋆	**.796**⋆	**.711**⋆	**.658**⋆	.711⋆	.809⋆	**.831**⋆	**.758**⋆	**.738**⋆	**.682**⋆
LQ(Ent;QF)	.774⋆	.787⋆	.848⋆	.786⋆	.614⋆	.537⋆	.724⋆	**.816**⋆	.778⋆	.702⋆	.675⋆	.617⋆
LQ(IDS;QF)	.735⋆	.681⋆	.773⋆	.750⋆	.540⋆	.446⋆	.675⋆	.708⋆	.736⋆	.642⋆	.619⋆	.518⋆
LQ(PR;QF)	.770⋆	.736⋆	.828⋆	.756⋆	.615⋆	.492⋆	.697⋆	.706⋆	.743⋆	.663⋆	.624⋆	.547
LQ(Cmb;QF)	**.798**⋆	.745⋆	.845⋆	.780⋆	.651⋆	.532	.641	.702⋆	.795⋆	.726⋆	.675⋆	.534
UEF	.467	.618	.426	.717	.395	.572	.645	.792	.614	.642	.600	.576
LQ(NS;UEF)	.758⋆	**.784**⋆	.787⋆	.773⋆	.613⋆	.575⋆	.702⋆	.805⋆	.704⋆	.704⋆	**.689**⋆	.632⋆
LQ(SW1;UEF)	.705⋆	.743⋆	.780⋆	.790⋆	.614⋆	.605⋆	**.707**⋆	.799⋆	.712⋆	.724⋆	.659⋆	.633⋆
LQ(SW2;UEF)	.720⋆	.765⋆	.793⋆	**.797**⋆	.629⋆	**.625**⋆	.689⋆	.801⋆	.728⋆	**.741**⋆	.669⋆	**.643**⋆
LQ(Ent;UEF)	.707⋆	.760⋆	.757⋆	.768⋆	.570⋆	.575⋆	.704⋆	**.809**⋆	.689⋆	.714⋆	.654⋆	.635⋆
LQ(IDS;UEF)	.706⋆	.660⋆	.725⋆	.739⋆	.477⋆	.533⋆	.705⋆	.748⋆	.641⋆	.692⋆	.653⋆	.577
LQ(PR;UEF)	.728⋆	.736⋆	.743⋆	.737⋆	.568⋆	.570	.690⋆	.780⋆	.647⋆	.684⋆	.659⋆	.605⋆
LQ(Cmb;UEF)	**.784**⋆	.758⋆	**.816**⋆	.783⋆	**.650**⋆	.585	.659	.719⋆	**.738**⋆	.719⋆	.669⋆	.593⋆

prediction on the Web. Yet, using the quality measures alone yields prediction quality that is inferior to that of integrating them with the post-retrieval predictors in our LQ approach.

A closer look at Table 1 reveals the following. The Clr (Clarity) predictor is ineffective for the Web as was previously reported [1]. IClr, which was specifically designed for Web settings, shows some improvement over Clr, yet it is outperformed by almost all other basic predictors and by all the LQ predictors. Although QF and UEF are the most effective non-LQ predictors, when used as part of the LQ prediction method the prediction quality almost always improves.

The best prediction quality for an experimental setting (marked by underline) is obtained for the LQ predictor when integrated with either QF or UEF. For 11 out of the 12 experimental settings, the best prediction quality is attained when using the SW2 measure. This finding supports the benefit in quantifying the "richness" of language in a document and using the quantification as a document quality measure in the prediction task.

The Cmb (Combine) method, which integrates *all* the various quality measures, often yields lower prediction quality than that of using the highly effective NS and SW2 quality measures alone. This could be attributed to the simple product-based integration employed by Cmb. Using other (parameterized) integration approaches is left for future work.

The LQ methods are effective for both the QL (created without suspected-spam removal) and QL-SR (created using suspected-spam removal) result lists. The fact that using the NS measure for QL-SR is effective implies that this result list is not "clean of spam".

Comparing Document Quality Measures. The comparison of the prediction quality of using the various document quality measures is presented in Table 2. We use QF and UEF for the basic predictors as these were found in Table 1 to be the most effective basic predictors when used alone. We can see in Table 2 that in a vast majority of the relevant comparisons integrating a basic predictor with a quality measure in LQ yields prediction quality that transcends that of using the basic predictor alone. As QF and UEF are state-of-the-art prediction methods in their own right, these findings attest to the substantial merits of utilizing document quality measures for predicting Web-search effectiveness.

Table 2 also shows that IDS and PR (PageRank) are less effective, in most cases, than all other quality measures.[3] A case in point, the best prediction quality per column and basic predictor is never attained for IDS nor for PR. Thus, we see that quality measures that are solely based on the document content, rather than on its association with other documents, turn out to be the most effective for query-performance prediction among those we considered.

5 Conclusion

We addressed the task of query-performance prediction for Web search. We showed that using information induced from query-independent document quality measures, in addition to that utilized by previously proposed predictors, can yield prediction quality that is much better than the state-of-the-art.

Acknowledgments. We thank the reviewers for their comments. This work has been supported by and carried out at the Technion-Microsoft Electronic Commerce Research Center. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

References

1. Balasubramanian, N., Kumaran, G., Carvalho, V.R.: Predicting query performance on the web. In: Proc. of SIGIR, pp. 785–786 (2010)
2. Bendersky, M., Croft, W.B., Diao, Y.: Quality-biased ranking of web documents. In: Proc. of WSDM, pp. 95–104 (2011)
3. Bernstein, Y., Billerbeck, B., Garcia, S., Lester, N., Scholer, F., Zobel, J.: RMIT university at trec 2005: Terabyte and robust track. In: Proc. of TREC-14 (2005)
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: Proc. of WWW, pp. 107–117 (1998)
5. Carmel, D., Yom-Tov, E.: Estimating the Query Difficulty for Information Retrieval. In: Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool (2010)
6. Carmel, D., Yom-Tov, E., Darlow, A., Pelleg, D.: What makes a query difficult? In: Proc. of SIGIR, pp. 390–397 (2006)

[3] It is worth noting that increased list diversity, as measured by the list entropy, was recently shown [14] to attest to effective retrieval in large-scale Web settings in contrast to the case for newswire collections.

7. Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the trec 2009 web track. In: Proc. of TREC (2009)
8. Cormack, G.V., Smucker, M.D., Clarke, C.L.A.: Efficient and effective spam filtering and re-ranking for large web datasets. Information Retrieval 14(5), 441–465 (2011)
9. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: Proc. of SIGIR, pp. 299–306 (2002)
10. Diaz, F.: Performance prediction using spatial autocorrelation. In: Proc. of SIGIR, pp. 583–590 (2007)
11. Gyöngyi, Z., Garcia-Molina, H.: Web spam taxonomy. In: Proc. of AIRWeb, pp. 39–47 (2005)
12. Hauff, C., Kelly, D., Azzopardi, L.: A comparison of user and system query performance predictions. In: Proc. of CIKM, pp. 979–988 (2010)
13. Hauff, C., Murdock, V., Baeza-Yates, R.A.: Improved query difficulty prediction for the web. In: Proc. of CIKM, pp. 439–448 (2008)
14. Hummel, S., Shtok, A., Raiber, F., Kurland, O., Carmel, D.: Clarity re-visited. In: Proc. of SIGIR, pp. 1039–1040 (2012)
15. Kurland, O., Lee, L.: PageRank without hyperlinks: Structural re-ranking using links induced by language models. In: Proc. of SIGIR, pp. 306–313 (2005)
16. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: Proc. of SIGIR, pp. 120–127 (2001)
17. Lin, J., Metzler, D., Elsayed, T., Wang, L.: Of Ivory and Smurfs: Loxodontan MapReduce Experiments for Web Search. In: Proc. of TREC 2009 (2010)
18. Shtok, A., Kurland, O., Carmel, D.: Predicting Query Performance by Query-Drift Estimation. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 305–312. Springer, Heidelberg (2009)
19. Shtok, A., Kurland, O., Carmel, D.: Using statistical decision theory and relevance models for query-performance prediction. In: Proc. of SIGIR (2010)
20. Song, F., Croft, W.B.: A general language model for information retrieval (poster abstract). In: Proc. of SIGIR, pp. 279–280 (1999)
21. Tomlinson, S.: Robust, Web and Terabyte Retrieval with Hummingbird Search Server at TREC 2004. In: Proc. of TREC-13 (2004)
22. Vinay, V., Cox, I.J., Milic-Frayling, N., Wood, K.R.: On ranking the effectiveness of searches. In: Proc. of SIGIR, pp. 398–404 (2006)
23. Voorhees, E.M.: Overview of the TREC 2004 Robust Retrieval Track. In: Proc. of TREC-13 (2004)
24. Zhai, C., Lafferty, J.D.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proc. of SIGIR, pp. 334–342 (2001)
25. Zhao, Y., Scholer, F., Tsegay, Y.: Effective Pre-retrieval Query Performance Prediction Using Similarity and Variability Evidence. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 52–64. Springer, Heidelberg (2008)
26. Zhou, Y., Croft, B.: Ranking robustness: a novel framework to predict query performance. In: Proc. of CIKM, pp. 567–574 (2006)
27. Zhou, Y., Croft, B.: Query performance prediction in web search environments. In: Proc. of SIGIR, pp. 543–550 (2007)
28. Zhou, Y., Croft, W.B.: Document quality models for web ad hoc retrieval. In: Proc. of CIKM, pp. 331–332 (2005)
29. Zhu, X., Gauch, S.: Incorporating quality metrics in centralized/distributed information retrieval on the world wide web. In: Proc. of SIGIR, pp. 288–295 (2000)