Back to the Roots: A Probabilistic Framework for Query-Performance Prediction

Oren Kurland[1]
kurland@ie.technion.ac.il

Anna Shtok[1]
annabel@tx.technion.ac.il

Shay Hummel[1]
hummels@tx.technion.ac.il

Fiana Raiber[1]
fiana@tx.technion.ac.il

David Carmel[2]
carmel@il.ibm.com

Ofri Rom[1]
ofri.rom@gmail.com

1. Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel
2. IBM Research Lab, Haifa 31905, Israel

ABSTRACT

The query-performance prediction task is estimating the effectiveness of a search performed in response to a query when no relevance judgments are available. Although there exist many effective prediction methods, these differ substantially in their basic principles, and rely on diverse hypotheses about the characteristics of effective retrieval. We present a novel fundamental probabilistic prediction framework. Using the framework, we derive and explain various previously proposed prediction methods that might seem completely different, but turn out to share the same *formal* basis. The derivations provide new perspectives on several predictors (e.g., Clarity). The framework is also used to devise new prediction approaches that outperform the state-of-the-art.

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

Keywords: query-performance prediction

1. INTRODUCTION

The query-performance prediction task has attracted a lot of research attention [4]. The goal of this task is to estimate the effectiveness of a search performed in response to a query when there is a lack of relevance judgments. The prediction can be performed before retrieval using the query and corpus-based information [19, 16]. Post-retrieval prediction, on the other hand, also uses information induced from the result list of the most highly ranked documents [4].

Although there exists abundance of effective prediction methods, these often rely on substantially different principles. More specifically, various predictors are based on completely different hypotheses with respect to what results in effective retrieval. For example, some post-retrieval prediction methods rely on the premise that a result list that exhibits high *clarity* with respect to the corpus indicates effective retrieval [10, 1, 11, 5, 18]. Other prediction methods assume that effective retrieval should be manifested in a result list that is robust with respect to query perturbations [37, 42], document perturbations [36, 41], and retrieval method perturbations [2]. Another class of prediction methods is based on various analyses of retrieval scores in the result list [35, 14, 42, 30, 12, 13].

Given the large variety of prediction approaches, and the underlying hypotheses on which they are based, a few questions arise. The most fundamental one is whether there is a unified formal basis (framework) that can help explain various prediction methods and techniques. A related, albeit more specific, question is "what formal aspects of prediction are shared by seemingly different prediction approaches?". The operational question that naturally emerges is whether a formal analysis of the prediction task can give rise to new (effective) prediction approaches.

To address the questions stated here, we present a novel query-performance prediction framework that is based on fundamental probabilistic IR principles. We establish the framework by starting with a basic question that has not been explicitly addressed in previous work on prediction: "What is the probability that *this* result list is relevant to *this* query?". This question is a generalization to the document-list case of the core question of probabilistic retrieval [33, 22, 27]: "What is the probability that *this* document is relevant to *this* query?".

The framework we present sets the first *formal* grounds for integrating pre-retrieval and post-retrieval prediction methods. We show that these two paradigms target different, yet complementary, *formal* aspects of prediction. Empirical evaluation demonstrates the merits of their integration.

We use the framework to explain and derive various post-retrieval predictors that might seem at first glance to rely on completely different hypotheses and use different principles. For example, we derive (explain) the *Clarity* predictor [10]. This derivation provides a novel perspective about the actual property of the result list that Clarity quantifies. Furthermore, we use the framework to show that some predictors that are based on the result-list robustness notion [37, 36, 41, 42] and on analysis of the retrieval-scores distribution [14], implicitly share the same formal basis for prediction.

Our framework also provides a formal ground to using, for prediction, measures of query-independent properties of the result list (e.g., cohesion and dispersion); and, to integrating

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

these with query-dependent measures that are the focus of most previously proposed predictors.

The proposed framework not only provides new insight into existing predictors, it also gives rise to novel prediction approaches. Some of these outperform state-of-the-art methods, as shown by experiments conducted with a large variety of TREC corpora (including ClueWeb).

To summarize, our main contributions are twofold. The first, and perhaps most important, is on the formal side. The prediction framework that we present sets unifying formal grounds for a variety of prediction methods that might seem to be completely different. The formal analysis also results in novel insights about existing, and commonly used, prediction methods and the connections between them. The second contribution is using the framework to devise new prediction approaches that outperform state-of-the-art.

2. RELATED WORK

Pre-retrieval query-performance predictors [19, 29, 25, 16, 39, 28] analyze the query expression, often using corpus-based information. Post-retrieval predictors also use information induced from the result list of the most highly ranked documents [10, 1, 35, 37, 11, 5, 36, 2, 14, 42, 18, 30, 31, 7, 12, 13]. As noted above, the framework we present sets *formal* probabilistic grounds to the integration of pre-retrieval and post-retrieval prediction. We empirically show that the integration yields prediction quality that transcends the state-of-the-art. Furthermore, we show that the framework provides a *unified* formal basis that can be used to explain (derive), and provide new perspectives for, many previously proposed post-retrieval predictors.

A recently proposed framework [21] explains several post-retrieval predictors. The idea is that an effective result list is that which is similar to some pseudo effective list and dissimilar to a pseudo ineffective list. Our framework is shown to provide a formal probabilistic basis for this framework. Furthermore, our framework accounts for prediction aspects not accounted for by this framework [21]. (See Section 3.1.3 for a discussion.) Our framework also provides novel views of predictors such as Clarity [10] and WIG [42], which substantially depart from those previously proposed [21].

A conceptual framework for predicting *query difficulty* [5] is based on measuring similarities between the result list, the query, and the corpus. Our framework provides formal grounds to several aspects of this framework; and, helps explain several prediction methods and techniques that do not naturally fit in this framework; e.g, those using reference document lists [37, 36, 41, 14, 42].

A prediction framework [31], based on statistical decision theory, uses multiple re-rankings of the result list and their estimated effectiveness. We show that our framework provides probabilistic grounds for the underlying prediction principle of this framework. Furthermore, we use our framework to explain predictors that cannot be explained in terms of this approach [31]; e.g., Clarity [10] and WIG [42]. In addition, the integration of pre-retrieval and post-retrieval prediction that emerges in our framework was not addressed.

Notions of result list cohesion (dispersion) [37, 36] were argued to be correlated to some extent with retrieval effectiveness. We show that the *formal* prediction aspect targeted by measures of this query-independent result list property is complementary to that targeted by measures of query-dependent properties; the latter are the focus of most post-retrieval predictors. We also demonstrate the empirical merits of the integration of these two types of measures.

Integrating predictors using a linear interpolation of prediction values was employed with either pre-retrieval predictors [15] or post-retrieval predictors [36, 14, 42, 31]. In contrast, our framework provides formal grounds to the integration of different *types* of predictors which target different *formal* aspects of prediction; namely, pre-retrieval and post-retrieval prediction, and prediction based on query-independent and query-dependent measures of result list properties. Prediction integration as that proposed in previous work can be used in our framework (e.g., for improving pre-retrieval prediction quality) to potentially further improve overall prediction quality. We leave the exploration of this direction for future work.

3. PREDICTION FRAMEWORK

The query-performance prediction task is stated as estimating the effectiveness of a ranking induced by retrieval method $\mathcal{M}$ over a corpus of documents $\mathcal{D}$ in response to query q in lack of relevance judgments [4].

Our first key observation is that the goal of prediction can be stated, in probabilistic terms, as answering the question:

What is the probability that this result list ($\mathcal{D}_{res}$), of the most highly ranked documents, is relevant to this query (q)?

We focus on the result list and its ranking, rather than address the entire corpus ranking, as this is also the focus of the most commonly used evaluation measures (e.g., average precision, precision at top ranks, and NDCG). Furthermore, the question just stated is a generalization to the document list case of the question posed by Sparck Jones et al. [33] with respect to a single document: "What is the probability that *this* document is relevant to *this* query"? This question directly connects to the probability ranking principle [26] that states that maximal retrieval effectiveness is attained if documents are ranked by their relevance probabilities. The question serves as the basis for the probabilistic retrieval approach [33] and was used for a parallel derivation of the language modeling approach [22, 27].

3.1 Probabilistic prediction

In what follows we use the question stated above as the basis for developing our prediction framework. Let r be the event of relevance. The query-performance prediction task is, then, estimating $p(r|q, \mathcal{D}_{res})$, which can be written as

$$p(r|q, \mathcal{D}_{res}) = \frac{p(\mathcal{D}_{res}|q, r)p(r|q)}{p(\mathcal{D}_{res}|q)}. \quad (1)$$

The probability $p(\mathcal{D}_{res}|q)$ does not depend on relevance, but rather on the likelihood of retrieving $\mathcal{D}_{res}$ in response to q by the given retrieval method $\mathcal{M}$. As a case in point, if document-query surface-level similarities are used for retrieval, it is likely that $\mathcal{D}_{res}$ is composed of documents containing many occurrences of query terms. Furthermore, an estimate for $p(\mathcal{D}_{res}|q)$ can serve as a normalization factor to ensure the compatibility of prediction values across retrieval methods. However, as the evaluation of prediction quality in prior work was based on fixing the retrieval method (and varying the queries) [4], such normalization was not called

for. Accordingly, the expression

$$\mathcal{P}_{base}(\mathcal{D}_{res}; q) \stackrel{def}{=} p(\mathcal{D}_{res}|q, r)p(r|q) \quad (2)$$

from Equation 1 reflects the core prediction task. It turns out, as we show below, that $\mathcal{P}_{base}(\mathcal{D}_{res}; q)$ is the basis for numerous query-performance prediction methods that might seem at first glance to be completely different and/or based on different hypotheses with regard to the characteristics of effective retrieval. We further show how new prediction approaches can be devised based on $\mathcal{P}_{base}(\mathcal{D}_{res}; q)$. Finally, we note that if $\mathcal{D}_{res}$ contains a single document, then the prediction task modeled in Equation 2 is based on estimating the likelihood of the document given the query. This is the basis of the probabilistic approach to retrieval [33].

3.1.1 Integrating pre- and post- retrieval prediction

Estimating the prior probability of relevance to q in Equation 2, $p(r|q)$, is (in spirit) the goal of pre-retrieval prediction methods that operate prior to retrieval time [19, 16]. Indeed, pre-retrieval predictors quantify this *query difficulty* notion using information induced only from q and the corpus.

The probability $p(\mathcal{D}_{res}|q, r)$ is the likelihood of the result list $\mathcal{D}_{res}$ given that a relevance event happens for q; i.e., the probability that $\mathcal{D}_{res}$ is *the* list that provides information pertaining to q. This is the list-based generalization of the document likelihood principle used in probabilistic retrieval [33]. Estimating the list likelihood is the implicit goal of post-retrieval predictors.

Hence, our second key observation is that while by design, post-retrieval predictors use result-list-based information in addition to the information used by pre-retrieval predictors (the query and the corpus), the two types of predictors target different formal aspects of prediction; that is, the prior probability of relevance to a query (pre-retrieval) and the likelihood of a result list given relevance to the query (post-retrieval). To the best of our knowledge, Equation 1, and consequently Equation 2, set the first *formal* grounds to integrating pre-retrieval and post-retrieval prediction. We demonstrate the empirical merits of the integration in Section 4. In what follows we focus on $p(\mathcal{D}_{res}|q, r)$, i.e., post-retrieval prediction.

3.1.2 Post-retrieval prediction

Our third key observation is that $p(\mathcal{D}_{res}|q, r)$ can be estimated by using documents in $\mathcal{D}_{res}$ as $\mathcal{D}_{res}$'s proxies. We use $\hat{p}(\cdot)$ to denote an estimate for $p(\cdot)$. Let $p(d|q, r)$ be the probability that d is *the* document relevant to q; i.e., d's likelihood [33]. If we set $\hat{p}(d|q, r) \stackrel{def}{=} 0$ for $d \notin \mathcal{D}_{res}$, i.e., a document not in the result list is not considered relevant; and, $\sum_{d_i \in \mathcal{D}_{res}} \hat{p}(d_i|q, r) = 1$ holds, that is, $\hat{p}(d_i|q, r)$ is a probability distribution over $\mathcal{D}_{res}$, and more generally, over the corpus $\mathcal{D}$, then we can define the following:[1]

$$\hat{p}(\mathcal{D}_{res}|q, r) \stackrel{def}{=} \sum_{d_i \in \mathcal{D}_{res}} \hat{p}(\mathcal{D}_{res}|d_i, r)\hat{p}(d_i|q, r). \quad (3)$$

Assuming that $\hat{p}(r|d_i)$ (the prior probability for d_i's relevance), $\hat{p}(d_i)$ (the prior probability for d_i), and $\hat{p}(\mathcal{D}_{res})$ (the prior probability for the result list) are uniformly distributed, and since $\hat{p}(\mathcal{D}_{res}|d_i, r) = \frac{\hat{p}(d_i|\mathcal{D}_{res}, r)\hat{p}(r|\mathcal{D}_{res})\hat{p}(\mathcal{D}_{res})}{\hat{p}(r|d_i)\hat{p}(d_i)}$, Equation 3 then yields a generic post-retrieval predictor:

$$\mathcal{P}_{generic}(\mathcal{D}_{res}; q) \stackrel{def}{=} \hat{p}(r|\mathcal{D}_{res}) \sum_{d_i \in \mathcal{D}_{res}} \hat{p}(d_i|\mathcal{D}_{res}, r)\hat{p}(d_i|q, r). \quad (4)$$

Devising an estimate for the probability that $\mathcal{D}_{res}$ is relevant regardless of a specific query, $\hat{p}(r|\mathcal{D}_{res})$, is the implicit goal of several post-retrieval prediction methods. For example, some predictors are based on the premise that result list cohesion indicates effective retrieval, wherein cohesion can be quantified by the list radius [5] and its clustering tendency [36][2]. Conversely, a diversified result list might be considered as more likely to cover different aspects of the underlying information need, and thereby assumed to be relevant [5]. Hence, in Section 4 we empirically contrast the two hypotheses of list cohesion versus list dispersion as indicators for list relevance.

The summation in Equation 4 is of the weighted document likelihood values in $\mathcal{D}_{res}$ ($\hat{p}(d_i|q, r)$); d_i's weight is based on the strength of its "association" with $\mathcal{D}_{res}$ given that the latter is relevant ($\hat{p}(d_i|\mathcal{D}_{res}, r)$). For example, $\hat{p}(d_i|\mathcal{D}_{res}, r)$ can be based on the probability that d_i is generated from a language model induced from $\mathcal{D}_{res}$. We next show how Equation 4 can be used to explain additional predictors.

Explaining WIG. Let the document-list association strength in Equation 4, $\hat{p}(d_i|\mathcal{D}_{res}, r)$, be a uniform distribution over $\mathcal{D}_{res}$; i.e., each document in $\mathcal{D}_{res}$ is considered as its equi-important representative. Then, the average document likelihood score in $\mathcal{D}_{res}$, $\frac{1}{k}\sum_{d_i \in \mathcal{D}_{res}} \hat{p}(d_i|q, r)$, where k is the number of documents in $\mathcal{D}_{res}$, is used for prediction. This is perhaps the most direct manifestation of the probability ranking principle; that is, a result list with documents for which the probability for relevance is high is considered effective (relevant).[3] Furthermore, this prediction principle is the basis of the weighted information gain (WIG) predictor [42]. WIG measures the difference between the average retrieval score in the list and that of the corpus. The divergence from the corpus retrieval score serves to ensure inter-query compatibility of prediction values.

Explaining Clarity. As mentioned above, list-cohesion measures (e.g., radius) were suggested in prior work as performance predictors [5, 36]. These implicitly serve for the estimate $\hat{p}(r|\mathcal{D}_{res})$, of the result list relevance prior, in Equation 4. An alternative approach to measuring cohesion is using the corpus. That is, if a model of the result list is distant from that of the corpus, which can be viewed as a pseudo non-relevant document, then the list is considered focused; hence, the retrieval is presumed to be effective. This is the hypothesis underlying the Clarity prediction method [10].

However, the method used to compute Clarity *explicitly* depends on the query [10]. Thus, Clarity, *in implementation*, cannot be considered as an estimate for $p(r|\mathcal{D}_{res})$. Rather, Clarity turns out to be an estimate for the summation in Equation 4 as we show below.

[1] As is standard in work on mixture models, Equation 3 is based on an independence assumption; specifically, $\mathcal{D}_{res}$ is independent of q given d_i and r; i.e., we "back off" from $\mathcal{D}_{res}$ to its proxies (d_i).

[2] In implementation, the query is used when computing query-based inter-document similarities [36]. We come back to this point later on.

[3] In the probabilistic retrieval framework [33], for example, ranking is based on document likelihood values.

In Appendix A we show that the standard approach of computing Clarity can be (re-)written as:

$$\mathcal{P}_{Clarity}(\mathcal{D}_{res}; q) = \sum_{d_i \in \mathcal{D}_{res}} \hat{p}(d_i|q, r)\big(Sim(d_i, \mathcal{D}_{res}) - Sim(d_i, \mathcal{D})\big); \quad (5)$$

$Sim(d_i, X)$, where X is either $\mathcal{D}_{res}$ or $\mathcal{D}$, is the similarity (based on the cross entropy measure) between a language model induced from d_i and that induced from X.

Hence, Clarity is a special case of the generic predictor from Equation 4 with a uniform $\hat{p}(r|\mathcal{D}_{res})$. That is, Clarity is a weighted sum of document likelihood values, which in the language modeling case represent normalized query-likelihood values. (See Appendix A for details.) The document-list association strength ($\hat{p}(d_i|\mathcal{D}_{res}, r)$) is estimated in the Clarity case, as is shown in Equation 5, using the corpus-regularized similarity between the document and the result list; the corpus-based regularization downplays the effect of general non-query-related aspects when computing similarity to the list.

Thus, we attained the following novel perspective about Clarity. Suppose that the retrieval at hand, whose effectiveness we want to predict, uses the language-model-based surface-level similarity between documents and the query. This is the case for the query likelihood [32] and KL retrieval [23] methods that are commonly used in work on using Clarity for prediction. Hence, the induced ranking is essentially based on the document likelihood scores defined in Appendix A (i.e., normalized query-likelihood scores). Then, by Equation 5, a result list for which the highest ranked documents are similar to the entire list — wherein similarity is corpus regularized — is presumed to be effective (relevant).

3.1.3 Using reference lists

Equation 3 served as the basis for deriving the generic post-retrieval prediction approach presented in Equation 4. The underlying idea was to use documents in $\mathcal{D}_{res}$ as its proxies for estimating the result list likelihood, $p(\mathcal{D}_{res}|q, r)$. An alternative type of a proxy for $\mathcal{D}_{res}$ is a *reference document list* [37, 41, 14, 42, 31]. Formally, a similar formulation to that in Equation 3 that uses a set S_{ref} of reference document lists (denoted $\mathcal{D}_{ref}$) is:

$$\mathcal{P}_{ref}(\mathcal{D}_{res}; q) \stackrel{def}{=} \sum_{\mathcal{D}_{ref} \in S_{ref}} \hat{p}(\mathcal{D}_{res}|\mathcal{D}_{ref}, r)\hat{p}(\mathcal{D}_{ref}|q, r). \quad (6)$$

Thus, the relevance of $\mathcal{D}_{res}$ is estimated based on its association with the reference lists, $\hat{p}(\mathcal{D}_{res}|\mathcal{D}_{ref}, r)$, where $\mathcal{D}_{ref}$ is weighted by its presumed relevance, $\hat{p}(\mathcal{D}_{ref}|q, r)$. As we discuss below, Equation 6 sets the probabilistic formal grounds to quite a few predictors. These predictors differ by the choice of reference lists and inter-list association measure.

Explaining QF and autocorrelation. Devising the estimate $\hat{p}(\mathcal{D}_{ref}|q, r)$ is a prediction problem in its own right, which is not addressed by most predictors that utilize reference lists [37, 41, 2, 14, 42]. For example, the query feedback (QF) predictor [42] uses the overlap at top ranks between $\mathcal{D}_{res}$ and a (single) list $\mathcal{D}_{ref}$ for $\hat{p}(\mathcal{D}_{res}|\mathcal{D}_{ref}, r)$; $\mathcal{D}_{ref}$ is retrieved from the corpus using a (relevance) language model induced from $\mathcal{D}_{res}$. However, the presumed relevance of $\mathcal{D}_{ref}$, $\hat{p}(\mathcal{D}_{ref}|q, r)$, is not accounted for. Thus, it is not surprising that QF was found to be a highly effective predictor for the effectiveness of the reference result list ($\mathcal{D}_{ref}$) and not only to that of the original result list ($\mathcal{D}_{res}$) [31]. Indeed, this finding can be formally explained using Equation 6. That is, the effectiveness (relevance) of the reference list $\mathcal{D}_{ref}$ can be estimated using $\mathcal{D}_{res}$ as its proxy, by switching their roles in Equation 6. (The inter-list association estimate, overlap at top ranks, is symmetric.)

Autocorrelation [14] is another example of a predictor that does not account for the presumed relevance of a single reference list used for prediction. The reference list, $\mathcal{D}_{ref}$, is obtained by re-ranking $\mathcal{D}_{res}$ using score regularization; and, Pearson correlation between retrieval scores serves for $\hat{p}(\mathcal{D}_{res}|\mathcal{D}_{ref}, r)$. In Section 4 we show that the autocorrelation predictor does not only predict the effectiveness of $\mathcal{D}_{res}$, as originally reported, but also the effectiveness of $\mathcal{D}_{ref}$. This finding provides further support to the prediction principle presented in Equation 6.

Other predictors that use reference lists without accounting for their presumed relevance include those using query perturbations [37], document perturbations [41], and retrieval method perturbations [2, 14], to induce reference lists.

Explaining additional prediction approaches. The UEF predictor [31] does estimate the presumed relevance of the reference lists used. Specifically, each reference list is created by re-ranking $\mathcal{D}_{res}$ using a relevance language model. The presumed relevance of the reference list is estimated using previously proposed predictors. Thus, while UEF is based on statistical decision theory, it can be directly explained by Equation 6.

A prediction framework recently proposed [21] relies on the premise that the result list $\mathcal{D}_{res}$ is relevant to the extent that it is similar to a pseudo relevant result list and dissimilar to a pseudo non-relevant result list. The framework was used to explain several post-retrieval predictors. In contrast to the framework we present in this work, this framework does not arise from a probabilistic analysis of the prediction task. Moreover, it does not account for several formal aspects of prediction that emerged in the development of our framework. These include integrating pre-retrieval and post-retrieval prediction and integrating query dependent and independent measures of result list properties. Yet, in Appendix B we show that Equation 1, which served as the basis for deriving our framework, can be used to provide formal probabilistic grounds to this framework [21].

4. EVALUATION

We present an empirical exploration of three formal aspects that emerged in the development of our prediction framework. These give rise to new prediction methods and shed new light on existing ones.

We study the merits of the integration of pre-retrieval and post-retrieval prediction in Section 4.2.1. In Section 4.2.2 we explore the use of measures of query-independent properties — specifically, cohesion and dispersion — of the result list for prediction. Finally, in Section 4.2.3 we focus on using reference lists for prediction. We begin by describing the experimental setup we used for evaluation in Section 4.1.

4.1 Experimental setup

Table 1 presents the eight TREC-based experimental settings used for evaluation. TREC5 and ROBUST are composed primarily of newswire documents. WT10G is a small

Collection	Data	Num Docs	Topics
TREC5	Disks 2,4	524,929	251-300
ROBUST	Disks 4,5-CR	528,155	301-450, 601-700
WT10G	WT10g	1,692,096	451-550
GOV2	GOV2	25,205,179	701-850
Clue09	ClueWeb (Category B)	50,220,423	1-50
Clue09+SpamRm	ClueWeb (Category B)	50,220,423	1-50
Clue10	ClueWeb (Category B)	50,220,423	51-100
Clue10+SpamRm	ClueWeb (Category B)	50,220,423	51-100

Table 1: TREC settings used for experiments.

Web collection that contains some pages of low quality (e.g., spam). GOV2 is a much larger Web collection that is a crawl of the .GOV domain and hence contains mainly well edited pages. We also use the ClueWeb corpus (category B) [6], which is a large scale noisy Web collection, with the queries used in TREC 2009 (Clue09) and TREC 2010 (Clue10).

Retrieval effectiveness for ClueWeb can be significantly influenced by spam documents [8]. Thus, in what follows we explore the potential effects of spam on query-performance prediction. Specifically, we also perform evaluation where documents "suspected" as spam that are initially highly ranked are removed from the result list. To that end, we scan the initial ranking from top to bottom and remove documents that are assigned by Waterloo's spam classifier [8] a score below 50 [8, 3], until 1000 documents are accumulated. The score reflects the presumed percentage of documents in the ClueWeb English collection (category A) that are "spammier" than the document at hand. Clue09+SpamRm and Clue10+SpamRm denote the resulting (ranked) corpora for Clue09 and Clue10, respectively.

The titles of TREC topics serve as queries. We applied Porter stemming and stopword removal, using the INQUERY list, to documents and queries using the Lemur/Indri toolkit[4].

To measure prediction quality, we follow the standard practice in work on query-performance prediction [4]. Specifically, we report the Pearson correlation between the prediction values assigned by a predictor to the set of queries per setting, and the ground-truth average precision (AP@1000) values for these queries; the ground truth is determined based on TREC's relevance judgments.[5] Statistically significant differences of prediction quality (between Pearson correlations) are determined at the 95% confidence level [34].

As in many previous reports of work on predicting query performance [10, 11, 41, 14, 16, 30, 17, 31], we use the query likelihood (QL) model [32] for the retrieval method. The goal of the predictors we study is to estimate the retrieval effectiveness of this standard language-model-based retrieval approach. The QL retrieval score assigned to document d in response to query q is $Score_{QL}(q;d) \stackrel{def}{=} \log \prod_{q_i \in q} p(q_i|d)$, where $p(w|d)$ is the probability assigned to term w by a language model induced from d; q_i is a query term. Unless otherwise stated, we use Dirichlet-smoothed unigram language models with the smoothing parameter set to 1000 [38].

Post-retrieval predictors. We use the Clarity [10], query feedback (QF) [42], and weighted information gain (WIG) [42] predictors in the following. These methods represent the three classes of post-retrieval prediction approaches mentioned in Section 1; these include, measuring the clarity of the result list with respect to the corpus (Clarity), quantifying the robustness of the result list (QF), and analyzing properties of retrieval scores in the result list (WIG). WIG and QF were shown to yield state-of-the-art prediction quality [42, 40, 31]. In Section 4.2.3 we address the autocorrelation predictor [14], which uses regularized retrieval scores.

The predictors just mentioned, as other previously proposed post-retrieval predictors, do not provide *direct* estimates for the likelihood of the result list. (Refer back to Equation 2 and the accompanying discussion.[6]) Rather, the assigned prediction values are correlated with this likelihood as implied by the attained prediction quality. We hasten to point out, however, that the purpose of using these predictors in the evaluation to follow is exploring aspects and principles that emerged in the development of our framework; specifically, the integration of different *types* of prediction approaches, namely, pre-retrieval and post-retrieval methods and query-independent and query-dependent measures of properties of the result list; and, the use of reference lists for prediction. Devising novel predictors that provide direct estimates for the result list likelihood, and using them in our framework, is left to future work.

The post-retrieval predictors analyze the result list $\mathcal{D}_{res}$ of the k documents that are the highest ranked by the QL retrieval method. The Clarity predictor [10] measures the KL divergence between a relevance language model $R_{\mathcal{D}_{res}}$ induced from $\mathcal{D}_{res}$ and a language model induced from the corpus. (See Appendix A for details.) The QF prediction value [42] is the number of documents that are both among the ν_{QF} highest ranked by the QL method and among the ν_{QF} highest ranked by retrieval performed over the corpus using the relevance model $R_{\mathcal{D}_{res}}$; ν_{QF} is a free parameter. The prediction value of WIG is $\frac{1}{\sqrt{|q|}} \frac{1}{k} \sum_{d \in \mathcal{D}_{res}} (Score_{QL}(q;d) - Score_{QL}(q;\mathcal{D}))$, where $|q|$ is the number of terms in q; i.e., the (query-length normalized) difference between the average retrieval score in $\mathcal{D}_{res}$ and that of the corpus[7].

One of the goals of the exploration we present in Section 4.2 is to study whether *formal* aspects of prediction that emerged in the framework described above can be used to devise prediction methods that outperform state-of-the-art predictors. Hence, we use highly optimized versions of Clarity, QF, and WIG as baselines. Specifically, we set the free parameters of each of these predictors to values maximizing prediction quality per setting. All three predictors rely on k, the number of documents in the result list; QF also depends on ν_{QF}. For all settings, except for those for ClueWeb, the value of k is selected from $\{5, 10, 50, 100, 150, 200, 300, 500, 700, 1000, 2000, 3000, 4000, 5000\}$. For ClueWeb we use a slightly more moderate range of values to alleviate the computational effort: $k \in \{5, 10, 50, 100, 150, 200, 300, 500, 700,$

[4]www.lemurproject.org

[5]Prediction-quality patterns similar to those we report are observed if Kendall's-τ is used for evaluation [4]. Actual numbers are omitted to avoid cluttering the presentation.

[6]There is work on estimating average precision directly from score distributions [12]. The proposed estimates are still surrogates for estimates for the result list likelihood.

[7]The corpus $\mathcal{D}$ is represented by the unsmoothed language model induced from the concatenation of the documents it contains; concatenation order has no effect, as we use unigram language models. While WIG's original implementation was with the Markov Random Field model [42], it was noted [42, 40] and shown [31] that WIG is highly effective for predicting the effectiveness of the query likelihood model.

	TREC5	ROBUST	WT10G	GOV2	Clue09	Clue09+SpamRm	Clue10	Clue10+SpamRm
SumIDF	0.053	0.287	0.162	0.281	0.618	0.620	0.202	0.201
AvgIDF	0.208	0.466	0.149	0.266	0.211	0.161	0.175	0.227
MaxIDF	0.269	0.486	0.134	0.294	0.409	0.357	0.233	0.252
SumVarTF.IDF	0.049	0.266	0.269	0.328	0.631	0.637	0.267	0.273
AvgVarTF.IDF	0.203	0.428	0.282	0.360	0.365	0.293	0.247	0.309
MaxVarTF.IDF	0.180	0.443	0.406	0.372	0.590	0.528	0.312	0.341
Clarity	**0.431**	0.522	0.432	0.456	0.105	−0.016	−0.147	−0.077
SumIDF∧Clarity	0.170	0.392	0.270	0.368	**0.687***	**0.643***	0.156*	0.225*
AvgIDF∧Clarity	0.310	0.527	0.287	0.367	0.254	0.157	0.093*	0.174*
MaxIDF∧Clarity	0.345	**0.545**	0.265	0.383	0.457*	0.365*	0.149*	0.222*
SumVarTF.IDF∧Clarity	0.187	0.358	0.358	0.420	0.667*	0.623*	**0.207***	**0.289***
AvgVarTF.IDF∧Clarity	0.341	0.482	0.401	**0.459**	0.338*	0.228*	0.139*	0.224*
MaxVarTF.IDF∧Clarity	0.300	0.483	**0.450**	0.447	0.555*	0.458*	0.206*	0.270*
QF	0.447	0.500	0.483	0.566	0.516	0.616	0.393	0.483
SumIDF∧QF	0.388	0.505	0.487	0.502	0.675*	**0.767***	0.496	0.674*
AvgIDF∧QF	0.444	0.602*	0.423	0.518	0.499	0.443	0.520*	0.559*
MaxIDF∧QF	**0.489**	**0.613***	0.482	0.554	0.604	0.595	0.537*	0.610*
SumVarTF.IDF∧QF	0.366	0.496	**0.511**	0.527	**0.689***	0.751*	0.524*	**0.684***
AvgVarTF.IDF∧QF	0.409	0.586*	0.432	0.561	0.557	0.488	0.538*	0.579*
MaxVarTF.IDF∧QF	0.390	0.578*	0.496	**0.574**	0.651*	0.643	**0.557**	0.603*
WIG	**0.297**	0.550	0.376	**0.486**	0.507	0.498	**0.415**	**0.468**
SumIDF∧WIG	0.133	0.425	0.314	0.362	0.611*	0.606*	0.233	0.260
AvgIDF∧WIG	0.222	0.540	0.218	0.348	0.246	0.225	0.240	0.302
MaxIDF∧WIG	0.270	**0.572**	0.261	0.381	0.406	0.385	0.288	0.332
SumVarTF.IDF∧WIG	0.138	0.391	0.404	0.427	**0.677***	**0.666***	0.277	0.305
AvgVarTF.IDF∧WIG	0.239	0.508	0.334	0.432	0.409	0.372	0.287	0.352
MaxVarTF.IDF∧WIG	0.226	0.524	**0.431**	0.465	0.587	0.546	0.333	0.372

Table 2: Integrating pre-retrieval and post-retrieval prediction. The best result per setting in a block of a post-retrieval predictor is boldfaced. The best result in the entire setting (column) is underlined. '*' marks a statistically significant difference with using the post-retrieval predictor alone.

1000}. For all settings, ν_{QF}, the additional free parameter of QF, is set to values in {5, 10, 50, 100, 150, 200, 300, 500, 700, 1000}. Following previous recommendations [31], the number of terms used by the relevance model, $R_{\mathcal{D}_{res}}$, which is utilized by Clarity and QF, is set to 100; and, language models of documents from which the relevance model is constructed are not smoothed (i.e., a maximum likelihood estimate is used).

4.2 Experimental results

4.2.1 Integrating pre- and post- retrieval prediction

Using Equation 2 we showed that pre-retrieval and post-retrieval prediction methods target different, yet complementary, formal aspects of prediction. Hence, we now turn to study whether their integration can yield prediction quality that transcends that of using each alone. To the best of our knowledge, this is the first such empirical study.

For post-retrieval predictors we use the optimized Clarity, QF and WIG methods described above. For pre-retrieval prediction we use two sets of methods, each based on a different type of statistics computed for a query term. The first set, referred to as IDF [10, 19, 16], uses the inverse document frequency (IDF) value of a query term. The second set of pre-retrieval predictors, denoted VarTF.IDF [39], uses the variance of the TF.IDF value of a query term in documents across the corpus in which it appears. Predictors based on VarTF.IDF were shown to substantially outperform other pre-retrieval predictors [16]; and, to outperform highly effective post-retrieval predictors for Clue09 [17].

To aggregate the statistics for the query terms, we use the sum, average, and maximum operators. Accordingly, we study the SumIDF, AvgIDF, MaxIDF, SumVarTF.IDF, AvgVarTF.IDF, and MaxVarTF.IDF predictors. If X is a pre-retrieval predictor and Y is a post-retrieval predictor, we use X∧Y to denote their integration, attained by the multiplication of prediction values following Equation 2.

In what follows we focus on the question of whether the prediction quality of a highly optimized post-retrieval predictor can be improved if integrated, *as is*, with pre-retrieval predictors. Our experiments show that the prediction quality of the *integration* of pre-retrieval and post-retrieval prediction can be further improved, as expected, by optimizing the free-parameter values of the post-retrieval predictors to that end. (The pre-retrieval predictors do not incorporate free parameters.) Actual numbers are omitted due to space considerations and as they convey no additional insight.

Our first observation based on Table 2, which is in line with previous reports [4], is that all post-retrieval predictors, when used alone, almost always outperform all pre-retrieval predictors, when used alone, for TREC5, ROBUST, WT10G and GOV2. For the ClueWeb settings, the pre-retrieval predictors can in some cases outperform the post-retrieval predictors. This finding generalizes that presented in a recent report for (only) the Clue09 setting [17].

Evidently, spam removal for ClueWeb can have considerable (positive or negative) effect on prediction quality. We see in Table 2 that in most cases the prediction quality of the predictors for Clue09+SpamRm is lower than that for Clue09, while the prediction quality for Clue10+SpamRm is superior to that for Clue10; the prediction quality differences can be quite substantial.

Perhaps the most important observation with regard to Table 2 is the following. In quite a few cases, the integration of pre-retrieval and post-retrieval prediction can yield prediction quality that much transcends that of each, even

	TREC5	ROBUST	WT10G	GOV2	Clue09	Clue09+SpamRm	Clue10	Clue10+SpamRm
d-dispersion	0.082	0.253	0.158	−0.228	−0.338	−0.082	−0.327	−0.328
d-cohesion	−0.084	−0.257	−0.149	0.244	0.153	−0.136	−0.097	−0.104
e-dispersion	−0.033	0.139	0.159	−0.100	0.324	0.349	0.391	0.396
e-cohesion	0.028	−0.138	−0.157	0.094	−0.274	−0.323	−0.339	−0.326
Clarity	0.431	**0.522**	0.432	0.456	0.105	−0.016	−0.147	−0.077
d-dispersion∧Clarity	0.313	0.490	0.418	0.066	−0.229	−0.056	−0.284	−0.248
d-cohesion∧Clarity	**0.522**	0.460	0.327	**0.471**	0.130	−0.139	−0.110	−0.118
e-dispersion∧Clarity	0.404	**0.522**	**0.455**	0.447	**0.353***	**0.215***	**0.095***	**0.155***
e-cohesion∧Clarity	0.454	0.518	0.395	0.446	−0.148	−0.154	−0.301	−0.232
QF	0.447	0.500	0.483	**0.566**	0.516	0.616	0.393	0.483
d-dispersion∧QF	**0.504**	**0.528***	**0.496**	0.445	0.465	**0.697**	0.292	0.373
d-cohesion∧QF	0.368	0.447	0.432	0.544	0.527	0.014	−0.038	−0.021
e-dispersion∧QF	0.448	0.503	0.492	0.565	**0.569***	0.679*	**0.495***	**0.549***
e-cohesion∧QF	0.444	0.496	0.473	0.565	0.450	0.508	0.253	0.367
WIG	0.297	**0.550**	0.376	0.486	0.507	0.498	0.415	0.468
d-dispersion∧WIG	0.248	0.539	0.359	0.183	0.128	0.293	−0.091	−0.049
d-cohesion∧WIG	0.294	0.481	0.298	**0.550**	0.430	−0.006	−0.069	−0.072
e-dispersion∧WIG	0.281	**0.550**	**0.394**	0.463	**0.525**	**0.514**	**0.471**	**0.521**
e-cohesion∧WIG	**0.313**	0.547	0.349	0.500	0.293	0.410	0.104	0.183

Table 3: Integrating query-independent result list measures of cohesion and dispersion with the post-retrieval predictors that utilize query-dependent list properties; '*' marks statistically significant differences between the integration and using the post-retrieval predictor alone. The best result per setting in a block of a post-retrieval predictor is boldfaced; the best result in the entire setting (column) is underlined.

in cases where the former is far less effective than the latter; the improvements over using post-retrieval prediction alone are often statistically significant for the ClueWeb settings. For example, QF is the most effective among the three post-retrieval predictors, when these are used alone, for 6 out of the 8 experimental settings. Yet, the best prediction quality reported in the QF block in the table (boldfaced numbers), and more generally, for an entire setting (underlined numbers), is always attained when integrating QF with a pre-retrieval predictor. Furthermore, the prediction quality of the integration in these cases is often much better than that of QF, although QF's prediction quality is often substantially better than that of the pre-retrieval predictor with which it is integrated. Specifically, MaxIDF∧QF and SumVarTF.IDF∧QF turn out to be highly effective predictors. For Clarity and WIG, the best prediction quality is attained when integrating them with a pre-retrieval predictor in 7 and 4 out of the 8 experimental settings, respectively.

All in all, these findings attest to the complementary nature of pre-retrieval and post-retrieval prediction that formally emerged in Equation 2.

4.2.2 List cohesion versus list dispersion

The generic post-retrieval predictor in Equation 4 uses a query-independent estimate for the effectiveness (relevance) of the result list ($\hat{p}(r|\mathcal{D}_{res})$). This estimate plays a complementary formal role to that addressed by the query-dependent estimates used in most post-retrieval prediction methods. Thus, following Equation 4, we turn to study the potential merits of integrating the measures of query-independent and query-dependent properties of the result list.

We contrast two hypotheses. The first is that high cohesion of the result list attests to reduced *query drift* [36], and hence, to improved retrieval. The second is that list dispersion might indicate increased cover of query aspects, and thereby imply effective retrieval [5]. We employ two measures that quantify list cohesion and dispersion: the diameter of the list, adapted in spirit from [5], and the list entropy, a variant of a measure proposed in [20].

List diameter. We define the result list ($\mathcal{D}_{res}$) diameter as

$$dim(\mathcal{D}_{res}) \stackrel{def}{=} -\frac{1}{k(k-1)} \sum_{d_i,d_j \in \mathcal{D}_{res}; d_i \neq d_j} Sim(d_i, d_j);$$

$Sim(d_i, d_j) \stackrel{def}{=} -KL\left(p(\cdot|d_i) \,\middle|\middle|\, p(\cdot|d_j)\right)$ is the negative KL divergence between the (unsmoothed) language model induced from d_i and the (smoothed[8]) language model induced from d_j. Increased KL divergence corresponds to a larger difference between language models (i.e., weaker inter-document similarity); accordingly, $\mathcal{D}_{res}$ is of greater diameter and consequently assumed to be less coherent (more dispersed). We use **d-cohesion** and **d-dispersion** to refer to $\frac{1}{dim(\mathcal{D}_{res})}$ and $dim(\mathcal{D}_{res})$, respectively, that reflect the presumed extent of cohesion and dispersion, respectively.

List entropy. Let $Cent(\mathcal{D}_{res})$ be the (arithmetic) centroid of language models of documents in $\mathcal{D}_{res}$: $p(w|Cent(\mathcal{D}_{res})) \stackrel{def}{=} \frac{1}{k}\sum_{d\in\mathcal{D}_{res}} p(w|d)$, where $p(w|d)$ is the probability assigned to term w by an unsmoothed language model induced from d. We "clip" $Cent(\mathcal{D}_{res})$ by using the 100 terms w to which it assigns the highest probability, and re-normalize to yield a probability distribution. (As noted in Section 4.1, the same clipping practice was employed to the relevance model used by the Clarity and QF predictors.)

We define the entropy of the result list $\mathcal{D}_{res}$ as the entropy of its centroid:

$$H(\mathcal{D}_{res}) \stackrel{def}{=} -\sum_{w} p(w|Cent(\mathcal{D}_{res})) \log p(w|Cent(\mathcal{D}_{res})).$$

Low entropy amounts to the term distribution in $\mathcal{D}_{res}$ being focused around a few terms, which potentially implies increased list cohesion; conversely, high entropy might indicate increased list dispersion. We use **e-cohesion** and **e-dispersion** to refer to $\frac{1}{H(\mathcal{D}_{res})}$ and $H(\mathcal{D}_{res})$, respectively,

[8]The document language model is Jelinek-Mercer smoothed [38] with a 0.1 weight assigned to corpus-based term counts.

which are measures of $\mathcal{D}_{res}$'s presumed cohesion and dispersion, respectively.

To integrate the query-independent list cohesion (dispersion) measures with the optimized query-dependent post-retrieval predictors (Clarity, QF and WIG), we multiply their assigned values following Equation 4. The cohesion (dispersion) measures are computed for the $k = 100$ highest ranked documents. (Using the top-50 documents yields similar prediction quality.) Table 3 presents the prediction quality numbers. ($\wedge$ indicates that a list cohesion (dispersion) measure is integrated with a predictor.)

We can see in Table 3 that in most cases, the query-independent result-list cohesion (dispersion) measures yield very low prediction quality when used alone. The main exception is the e-dispersion measure, which yields somewhat higher prediction quality for the ClueWeb settings. This finding echoes those from previous work [20] about list dispersion being a potential signal for effective retrieval in the noisy Web setting.

We also see in Table 3 that for 7 out of the 8 experimental settings, the best prediction quality (underlined) is attained by integrating a dispersion measure (d-dispersion or e-dispersion) with a post-retrieval predictor. For all (optimized) post-retrieval predictors, Clarity, QF and WIG, the best prediction quality (boldfaced) is almost always obtained when integrated with a dispersion measure rather than when used alone. The improvements over using the post-retrieval predictor alone can be substantial (e.g., for the ClueWeb settings) and statistically significant (e.g., for e-dispersion$\wedge$Clarity and e-dispersion$\wedge$QF over the ClueWeb settings). In most cases, integrating a post-retrieval predictor with the entropy-based dispersion measure (e-dispersion) outperforms using the post-retrieval predictor alone and integrating the predictor with the diameter-based dispersion measure (d-dispersion).

The findings presented above support the merits of the integration of measures of query-independent and query-dependent result list properties that formally emerged in Equation 4. Specifically, quantifying list dispersion by measuring its term-distribution entropy, and integrating this measure with optimized post-retrieval predictors, yields prediction quality that can substantially transcend that of using the predictors alone.

It is important to note that our findings do not contradict those in previous work [36] with regard to increased list cohesion being an indicator for effective retrieval. Specifically, cohesion was shown to be an effective indicator when measured in a *query-dependent* manner utilizing document-query similarities [36]. In contrast, we measure cohesion using query-independent estimates.

4.2.3 Using reference lists for prediction

We presented the fundamental approach of using reference lists for prediction in Equation 6:

$$\mathcal{P}_{ref}(\mathcal{D}_{res}; q) \stackrel{def}{=} \sum_{\mathcal{D}_{ref} \in S_{ref}} \hat{p}(\mathcal{D}_{res}|\mathcal{D}_{ref}, r)\hat{p}(\mathcal{D}_{ref}|q, r).$$

The relevance of $\mathcal{D}_{res}$ to q is estimated based on its association with reference lists $\mathcal{D}_{ref}$ ($\hat{p}(\mathcal{D}_{res}|\mathcal{D}_{ref}, r)$), where each $\mathcal{D}_{ref}$ is weighted by its presumed relevance to q ($\hat{p}(\mathcal{D}_{ref}|q, r)$). As discussed in Section 3, this prediction paradigm underlies several predictors [37, 41, 14, 42, 31]. However, many of these predictors do not estimate, and consequently utilize, the presumed relevance of $\mathcal{D}_{ref}$ [37, 41, 14, 42].

	RefList(Uni)	RefList(Clarity)	Clarity
TREC5	0.414	**0.459**	0.431
ROBUST	0.520	**0.562**^{c}	0.522
WT10G	0.415	0.418	**0.432**
GOV2	0.377	**0.486**^{u}	0.456
Clue09	−0.06	0.019	**0.105**
Clue09+SpamRm	**0.121**	0.039	−0.016
Clue10	**−0.079**	−0.175	−0.147
Clue10+SpamRm	**0.047**	−0.054	−0.077

Table 4: Using reference lists for prediction. 'c' and 'u' mark statistically significant differences with Clarity and RefList(Uni), respectively. Best result in a row is boldfaced.

To further explore the merits, or lack thereof, of the reference-lists-based prediction paradigm, when using both the association between $\mathcal{D}_{res}$ and $\mathcal{D}_{ref}$ and the estimated relevance of $\mathcal{D}_{ref}$ to q, we study the following novel simple predictor, henceforth referred to as **RefList**.

In the empirical evaluation presented thus far, we used the QL retrieval method to induce the corpus ranking used to create the result list $\mathcal{D}_{res}$. The document language model smoothing parameter value was set to 1000. (See Section 4.1.) To create reference lists, $\mathcal{D}_{ref}$, we *re-rank* $\mathcal{D}_{res}$ using the QL method with the smoothing parameter set to values in $\{100, 200, 500, 800, 1500, 2000, 3000, 4000, 5000, 10000\}$. Thus, we get that S_{ref} contains 10 reference lists that are created by varying the document (language) models.

We set the estimate for the association strength of $\mathcal{D}_{res}$ and $\mathcal{D}_{ref}$ ($\hat{p}(\mathcal{D}_{res}|\mathcal{D}_{ref}, r)$) to Pearson's correlation between the retrieval scores of the documents. The estimate for the presumed relevance of $\mathcal{D}_{ref}$ to q ($\hat{p}(\mathcal{D}_{ref}|q, r)$) is set to either the Clarity value of $\mathcal{D}_{ref}$ (**RefList(Clarity)**) or to $\frac{1}{10}$ (**RefList(Uni)**). We employ the predictors upon a result list, $\mathcal{D}_{res}$, of $k = 150$ documents. The relevance model used by Clarity is constructed as described in Section 4.1. As a reference comparison, we use the *optimized* Clarity predictor used above and for which the value of k was tuned. Thus, the RefList predictors are *underoptimized* with respect to the Clarity baseline. We hasten to point out that RefList can also be optimized with respect to k and predictors other than Clarity can be used by RefList to estimate the relevance of $\mathcal{D}_{ref}$. Nevertheless, our goal here is to focus on the underlying principles of using reference lists for prediction rather than devise the most effective predictor. The prediction quality numbers are presented in Table 4.

We see in Table 4 that RefList(Clarity) improves — often substantially, although statistically significantly in a single case — over RefList(Uni) for all non-ClueWeb settings (TREC5, ROBUST, WT10G, and GOV2). This finding provides some support to the importance of using an estimate for the presumed relevance of the reference list. We also see that RefList(Clarity) substantially outperforms Clarity for 3 out of these 4 settings, and statistically significantly so for ROBUST, although RefList(Clarity) is *underoptimized* with respect to Clarity. These results support the merits of using reference lists for prediction, even if the reference lists are created using a very simple approach as we employ here.

Table 4 shows that for the ClueWeb settings, the RefList predictors and the optimized Clarity predictor yield very low prediction quality. This could be attributed to the noisy na-

		TREC5	ROBUST	WT10G	GOV2	Clue09	Clue09+SpamRm	Clue10	Clue10+SpamRm
QF	$\mathcal{D}_{res}$	0.447	0.500	**0.483**	0.560	0.516	**0.616**	0.393	0.483
	$\mathcal{D}_{ref}$	**0.467**	**0.594**	0.482	**0.643**	**0.605**	0.543	**0.445**	**0.574**
autocorrelation	$\mathcal{D}_{res}$	0.277	0.347	0.328	0.417	**0.235**	0.505	**−0.021**	**0.213**
	$\mathcal{D}_{ref}$	**0.287**	**0.427**	**0.399**	**0.442**	0.105	**0.538**	−0.080	0.064

Table 5: The prediction quality of QF and autocorrelation. Prediction quality is measured by the success in predicting the effectiveness of the result list at hand ($\mathcal{D}_{res}$, as originally proposed) and that of the reference list used ($\mathcal{D}_{ref}$). The best result in a column within a block of a predictor is boldfaced.

ture of the corpus-based term counts (e.g., due to spam) that affects Clarity computation and the smoothing of document language models which is very important in the RefList predictors. (See [20] for an additional explanation about Clarity's low prediction quality for ClueWeb.) We found that using Clarity (alone) with "term clipping" [18] yields a somewhat improved prediction for Clue09 and Clue09+SpamRm but not for Clue10 and Clue10+SpamRm.

On the symmetry in using reference lists. As noted in Section 3.1.3, the QF [42] and autocorrelation [14] predictors use a single reference list. However, both predictors do not use an estimate for the presumed relevance of the reference list. Hence, these predictors are based solely on the association between $\mathcal{D}_{res}$ and $\mathcal{D}_{ref}$ ($\hat{p}(\mathcal{D}_{res}|\mathcal{D}_{ref}, r)$). Thus, if we swap the roles of $\mathcal{D}_{res}$ and $\mathcal{D}_{ref}$ in Equation 6, and given that the inter-list association measure for both predictors is symmetric, we should get that the predictors do not only estimate the effectiveness of $\mathcal{D}_{res}$, but also that of $\mathcal{D}_{ref}$. This hypothesis was shown to hold for QF in some previous work [31], but the ClueWeb settings were not used. Hence, we (re-)examine the hypothesis for QF; and, we present a novel study of the hypothesis for autocorrelation.

We use the optimized QF predictor described in Section 4.1. The optimization was with respect to the prediction quality for the effectiveness of $\mathcal{D}_{res}$, which is computed (as is the case for $\mathcal{D}_{ref}$) as at the above; that is, by Pearson correlation with the true AP@1000. We also use an optimized autocorrelation predictor. Specifically, the number of nearest neighbors considered when constructing the graph [14] is selected from $\{5, 10, 50\}$ (a document is included in the set of its own nearest neighbors); and, k, the number of documents in $\mathcal{D}_{res}$ (and $\mathcal{D}_{ref}$) is selected from $\{50, 100\}$ to optimize prediction quality for $\mathcal{D}_{res}$ with respect to AP@k.[9] Accordingly, we report the prediction quality of autocorrelation for both $\mathcal{D}_{res}$ and $\mathcal{D}_{ref}$ with respect to AP@k. We do not use AP@1000 since autocorrelation is based on re-ranking $\mathcal{D}_{res}$ rather than ranking the entire corpus.

We see in Table 5 that in most settings QF and autocorrelation post high prediction quality for the effectiveness of the reference list ($\mathcal{D}_{ref}$); often, the prediction quality transcends that for the original result list ($\mathcal{D}_{res}$). Both QF [42] and autocorrelation [14] were originally stated as targeting the effectiveness of $\mathcal{D}_{res}$ and they were optimized to this end as described above. Hence, these findings — which are novel for autocorrelation — support the hypothesis stated above with regard to predictors using a single reference list without utilizing an estimate for its presumed relevance. That is, prediction is essentially performed for the effectiveness of both the result list and the reference list. These findings further support the underpinning of the reference-list-based prediction approach from Equation 6.

[9]We used a language-model-based inter-document similarity measure rather than a vector-space-based measure as in the original proposal [14].

5. CONCLUSIONS AND FUTURE WORK

We presented a novel probabilistic query-performance prediction framework. The importance of the framework is threefold. First, setting common formal grounds to various previously proposed prediction methods that might seem to rely on completely different principles and hypotheses. Second, providing new insights about commonly used prediction methods such as Clarity and the connections between them. Third, giving rise, based on formal arguments, to new prediction approaches that were empirically shown to improve over the state-of-the-art. Integrating various prediction types that emerged in our framework using additional approaches — e.g., machine-learning-based techniques — is a future venue we intend to explore.

Acknowledgments We thank the anonymous reviewers, and Yuval Nardi, for their comments. This paper is based upon work supported in part by the Israel Science Foundation under grant no. 557/09, by IBM's Ph.D. fellowship and SUR award, by Google's and Yahoo!'s faculty research awards, and by Miriam and Aaron Gutwirth Memorial Fellowship. Any opinions, findings and conclusions or recommendations expressed here are the authors' and do not necessarily reflect those of the sponsors.

6. REFERENCES

[1] G. Amati, C. Carpineto, and G. Romano. Query difficulty, robustness, and selective application of query expansion. In *Proc. of ECIR*, pages 127–137, 2004.

[2] J. A. Aslam and V. Pavlu. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In *Proc. of ECIR*, pages 198–209, 2007.

[3] M. Bendersky, W. B. Croft, and Y. Diao. Quality-biased ranking of web documents. In *Proc. of WSDM*, pages 95–104, 2011.

[4] D. Carmel and E. Yom-Tov. *Estimating the Query Difficulty for Information Retrieval*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2010.

[5] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. What makes a query difficult? In *Proc. of SIGIR*, pages 390–397, 2006.

[6] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2009 Web track. In *Proc. of TREC*, 2009.

[7] K. Collins-Thompson and P. N. Bennett. Predicting query performance via classification. In *Proc. of ECIR*, pages 140–152, 2010.

[8] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *CoRR*, abs/1004.5168, 2010.

[9] W. B. Croft and J. Lafferty, editors. *Language Modeling for Information Retrieval*. Number 13 in Information Retrieval Book Series. Kluwer, 2003.

[10] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proc. of SIGIR*, pages 299–306, 2002.

[11] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. A language modeling framework for selective query expansion. Technical Report IR-338, Center for Intelligent Information Retrieval, University of Massachusetts, 2004.
[12] R. Cummins. Predicting query performance directly from score distributions. In *Proc. of AIRS*, pages 315–326, 2011.
[13] R. Cummins, J. M. Jose, and C. O'Riordan. Improved query performance prediction using standard deviation. In *Proc. of SIGIR*, pages 1089–1090, 2011.
[14] F. Diaz. Performance prediction using spatial autocorrelation. In *Proc. of SIGIR*, pages 583–590, 2007.
[15] C. Hauff, L. Azzopardi, and D. Hiemstra. The combination and evaluation of query performance prediction methods. In *Proc. of ECIR*, pages 301–312, 2009.
[16] C. Hauff, D. Hiemstra, and F. de Jong. A survey of pre-retrieval query performance predictors. In *Proc. of CIKM*, pages 1419–1420, 2008.
[17] C. Hauff, D. Kelly, and L. Azzopardi. A comparison of user and system query performance predictions. In *Proc. of CIKM*, pages 979–988, 2010.
[18] C. Hauff, V. Murdock, and R. Baeza-Yates. Improved query difficulty prediction for the web. In *Proc. of CIKM*, pages 439–448, 2008.
[19] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *Proc. of SPIRE*, pages 43–54, 2004.
[20] S. Hummel, A. Shtok, F. Raiber, O. Kurland, and D. Carmel. Clarity re-visited. In *Proc. of SIGIR*, 2012. Poster.
[21] O. Kurland, A. Shtok, D. Carmel, and S. Hummel. A unified framework for post-retrieval query-performance prediction. In *Proc. of ICTIR*, pages 15–26, 2011.
[22] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. In Croft and Lafferty [9], pages 1–10.
[23] J. D. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of SIGIR*, pages 111–119, 2001.
[24] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proc. of SIGIR*, pages 120–127, 2001.
[25] J. Mothe and L. Tanguy. Linguistic features to predict query difficulty. In *ACM SIGIR 2005 Workshop on Predicting Query Difficulty - Methods and Applications*, 2005.
[26] S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, pages 294–304, 1977.
[27] T. Rölleke and J. Wang. A parallel derivation of probabilistic information retrieval models. In *SIGIR*, pages 107–114, 2006.
[28] F. Scholer and S. Garcia. A case for improved evaluation of query difficulty prediction. In *Proc. of SIGIR*, pages 640–641, 2009.
[29] F. Scholer, H. E. Williams, and A. Turpin. Query association surrogates for web search. *JASIST*, 55(7):637–650, 2004.
[30] A. Shtok, O. Kurland, and D. Carmel. Predicting query performance by query-drift estimation. In *Proc. of ICTIR*, pages 305–312, 2009.
[31] A. Shtok, O. Kurland, and D. Carmel. Using statistical decision theory and relevance models for query-performance prediction. In *Proccedings of SIGIR*, pages 259–266, 2010.
[32] F. Song and W. B. Croft. A general language model for information retrieval (poster abstract). In *Proc. of SIGIR*, pages 279–280, 1999.
[33] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 1. *Information Processing and Management*, 36(6):779–808, 2000.
[34] J. H. Steiger. Tests for comparing elements of a correlation matrix. *Psychological Bulletin*, 87(2):245–251, 1980.
[35] S. Tomlinson. Robust, Web and Terabyte Retrieval with Hummingbird Search Server at TREC 2004. In *Proc. of TREC-13*, 2004.
[36] V. Vinay, I. J. Cox, N. Milic-Frayling, and K. R. Wood. On ranking the effectiveness of searches. In *Proc. of SIGIR*, pages 398–404, 2006.
[37] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *Proc. of SIGIR*, pages 512–519, 2005.
[38] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. of SIGIR*, pages 334–342, 2001.
[39] Y. Zhao, F. Scholer, and Y. Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Proc. of ECIR*, pages 52–64, 2008.
[40] Y. Zhou. *Retrieval Performance Prediction and Document Quality*. PhD thesis, University of Massachusetts, 2007.
[41] Y. Zhou and W. B. Croft. Ranking robustness: a novel framework to predict query performance. In *Proc. of CIKM*, pages 567–574, 2006.
[42] Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *Proc. of SIGIR*, pages 543–550, 2007.

APPENDIX

A. CLARITY

Clarity is defined as the KL divergence between a relevance language model $R_{\mathcal{D}_{res}}$ constructed from $\mathcal{D}_{res}$ and a language model $p(\cdot|\mathcal{D})$ induced from the corpus [10]:

$$\mathcal{P}_{Clarity}(\mathcal{D}_{res};q) \stackrel{def}{=} \sum_w p(w|R_{\mathcal{D}_{res}}) \log \frac{p(w|R_{\mathcal{D}_{res}})}{p(w|\mathcal{D})}; \quad (7)$$

w is a term in the vocabulary. $R_{\mathcal{D}_{res}}$ is a linear mixture of language models of documents in $\mathcal{D}_{res}$ [24]:

$$p(w|R_{\mathcal{D}_{res}}) \stackrel{def}{=} \sum_{d_i \in \mathcal{D}_{res}} p(w|d_i)p(d_i|q,r); \quad (8)$$

$p(w|d_i)$ is the probability assigned to w by a language model induced from d_i; and, $p(d_i|q,r) \stackrel{def}{=} \frac{p(q|d_i,r)}{\sum_{d' \in \mathcal{D}_{res}} p(q|d',r)}$ is d_i's normalized query-likelihood when using uniform distributions for $p(r|d)$ and $p(d)$, where $d \in \mathcal{D}_{res}$.[10]

Using $R_{\mathcal{D}_{res}}$'s definition from Equation 8 in Equation 7, and applying arithmetic manipulation, we get that:

$$\mathcal{P}_{Clarity}(\mathcal{D}_{res};q) = \sum_{d_i \in \mathcal{D}_{res}} p(d_i|q,r)\Big(\sum_w p(w|d_i) \log p(w|R_{\mathcal{D}_{res}}) - \sum_w p(w|d_i) \log p(w|\mathcal{D})\Big).$$

We define the similarity between d_i and X ($\mathcal{D}_{res}$ or $\mathcal{D}$) as the minus cross entropy between their induced language models: $Sim(d_i,X) \stackrel{def}{=} \sum_w p(w|d_i) \log p(w|X)$, with $p(w|\mathcal{D}_{res}) \stackrel{def}{=} p(w|R_{\mathcal{D}_{res}})$. Thus, Clarity is:

$$\mathcal{P}_{Clarity}(\mathcal{D}_{res};q) = \sum_{d_i \in \mathcal{D}_{res}} p(d_i|q,r)\big(Sim(d_i,\mathcal{D}_{res}) - Sim(d_i,\mathcal{D})\big). \quad (9)$$

B. EXPLAINING A PREVIOUSLY PROPOSED PREDICTION FRAMEWORK

Applying log odds, which is a monotonic transformation, upon Equation 1 yields:

$$\log O(r|q,\mathcal{D}_{res}) \stackrel{def}{=} \log \frac{p(r|q,\mathcal{D}_{res})}{p(\bar{r}|q,\mathcal{D}_{res})} = \log p(\mathcal{D}_{res}|r,q)p(r|q) - \log p(\mathcal{D}_{res}|\bar{r},q)p(\bar{r}|q); \quad (10)$$

$\bar{r}$ is the non-relevance event.

Suppose we estimate $p(\mathcal{D}_{res}|r,q)$ and $p(\mathcal{D}_{res}|\bar{r},q)$ based on $\mathcal{D}_{res}$'s similarity with a (pseudo) relevant and non-relevant document lists, respectively; and, use a uniform relevance prior, $p(r|q) = p(\bar{r}|q) = \frac{1}{2}$. Then, we obtain the prediction principle underlying the framework proposed in [21].

[10]While the common notation used in the language modeling framework does not include an explicit relevance event indicator, here we follow Lafferty and Zhai [22] and use it.