Enhanced Mean Retrieval Score Estimation for Query Performance Prediction

Haggai Roitman, Shai Erera, Oren Sar-Shalom, Bar Weiner
IBM Research - Haifa
Haifa, Israel 31905
haggai,shaie,orensr,barw@il.ibm.com

ABSTRACT

We study the problem of mean retrieval score estimation for query performance prediction (QPP). We propose an enhanced estimator which estimates the mean based on calibrated retrieval scores. Each document score is adjusted based on features that model potential tradeoffs that may exist in the retrieval process of that specific document. Using the proposed estimator, we derive several previously suggested QPP methods, from which we gather an initial set of calibration features. Based on these features and few additional ones, we propose two estimator instantiations. Using an evaluation over several TREC benchmarks, we demonstrate the effectiveness of our estimation approach.

1 INTRODUCTION

In the absence of relevance judgements, query performance prediction (QPP) methods may be used for estimating search quality [4]. Similar to many previous post-retrieval QPP works [7–9, 12, 27, 30–32], we estimate query performance by analyzing retrieval scores. Specifically, we focus on the mean retrieval score as the search effectiveness indicator. The mean score captures the *central tendency* of the retrieval scores' distribution. The higher this tendency is, the more we shall assume that the observed retrieval scores represent actual effectiveness.

Our goal is to estimate the mean retrieval score as accurate as possible so as to improve prediction. We build on top of Kurland et al.'s [14] probabilistic QPP framework and derive a general "calibrated" mean score estimator. To this end, each document's retrieval score is calibrated based on several generative terms; each such term models a different aspect of the document's retrieval process that may affect its relevance. We further derive two variants of this estimator. While the first variant emphasizes more on query-related retrieval issues, the second one emphasizes more on corpus-related issues.

Evidently, there are tradeoffs in the design of such estimators. To address these tradeoffs, we treat the score calibration problem as a *multi-criteria decision problem*; and suggest an alternative estimator based on a discriminative calibration approach. For that, we represent the retrieval process of each document by a set of calibration features; each feature captures a different retrieval quality criterion that may govern the actual document's relevance. We then combine the various features using their weighted product.

Using our proposed approach, we derive several previously suggested QPP methods, including Clarity [7], where we show that they share the same basic grounds of mean estimation. Using this connection allows us to gather and study an initial set of calibration features that can be utilized, in addition to several other features that we study.

Using an evaluation over several TREC benchmarks, we demonstrate that, overall, our calibrated mean score estimator results in an enhanced QPP. Moreover, we demonstrate that, those previously suggested QPP methods can be further improved by redesigning them as private instances of our proposed estimator.

The rest of this paper is organized as follows. We discuss related works in Section 2, followed by the description of our mean estimation framework in Section 3. Using this framework, we study several previously suggested methods in Section 4 and suggest two new instantiations in Section 5. We then evaluate the proposed approach in Section 6 and shortly conclude in Section 7.

2 RELATED WORK

Retrieval scores in IR have been extensively studied [6–10, 13, 20, 27, 30–32]. Among previous works, those related to QPP can be further classified based on two main types of descriptive statistics that are being utilized for prediction, either the mean score [7, 31, 32] or score dispersion [9, 20, 27, 30, 31]. Similar to [7, 31, 32], we utilize the mean retrieval score as the main indicator for QPP and aim at estimating it as accurately as possible. We show that, such previous works can be viewed as private instances of our proposed approach. Compared to [6, 8, 10, 13], we do not try to fit a generative (mixed) model of score distributions. Rather, we calibrate the scores provided by the underlying retrieval model in a discriminative way by adjusting them according to **multiple retrieval quality criteria**.

Our proposed mean estimation approach is built on top of Kurland et al.'s [14] probabilistic QPP framework. Using this framework, the authors explained several of previously suggested QPP methods [14]. Yet, our work differs from [14]. First, we describe a general mean score estimation framework for QPP, a venue of research, that, to the best of our knowledge, was not previously explored. Similar to [14], we show how to derive some of previously suggested QPP methods such as Clairty and WIG. Yet, our work has a different motivation, where we propose **new interpretations** to these methods. In addition, compared to [14], we are further able to improve such baseline methods and derive a general predictor that outperforms them all.

ICTIR '17, October 01–04, 2017, Amsterdam, Netherlands.
© 2017 ACM. ISBN 978-1-4503-4490-6/17/10…$15.00
DOI: https://doi.org/10.1145/3121050.3121051

Raiber and Kurland [22] have studied several document level features for identifying representative "relevant" documents for retrieval. While we share some motivation with [22], our goal is completely different. The main goal in [22] is to improve document and cluster retrieval using a learning-to-rank method, trained over several document and cluster features; while our goal is to predict the effectiveness of a given retrieval, **without modifying it**.

Finally, some other previous works have combined the prediction of several QPP methods [3, 5, 23, 24, 26, 29]. Yet, within these works, such combination was performed **on whole lists basis**. This is in comparison to our approach that combines predictions based on the retrieval effectiveness of individual documents. Moreover, while we acknowledge that [3, 24] did utilize additional document-level retrieval features, their actual prediction is based on the **aggregations of each document-level feature** (e.g., the mean retrieval score itself, standard deviation, maximum/minimum scores, etc). Hence, similar to [5, 23, 26, 29], the actual combined predictor derived in [3, 24] is based on **list-level** properties.

3 FRAMEWORK

For a given query q, let D be a subset of documents, retrieved from a given corpus C by some underlying retrieval method $\mathcal{M}$. Let $s(d)$ further denote the corresponding retrieval score assigned by method $\mathcal{M}$ to document $d \in C$. Unless stated otherwise, we assume that D includes the top-k documents $d \in C$ with the highest scores.

In this work, we focus on post-retrieval QPP [4]. Let r denote the relevance event, our goal is, therefore, to estimate $p(D|q,r)$ – *the likelihood of finding relevant information for q in D* [14]. Similar to many previous works [7–9, 12, 30–32], we estimate $p(D|q,r)$ by analyzing the retrieval scores $s(d)$ of the documents in D. Specifically, we focus on the mean retrieval score $\mathbb{E}(s|D)$ as an indicator of QPP, whose value we wish to estimate as accurately as possible. The mean score "encodes" the *central tendency* of the retrieval scores's distribution. The higher this tendency is, the more confidence we assign to the ability of the underlying method $\mathcal{M}$ to retrieve relevant documents in D [4].

3.1 Towards a general mean estimator for QPP

Our main goal is to estimate the mean retrieval score $\mathbb{E}(s|D)$ as accurately as possible. To accomplish that, we wish to derive a general mean estimator that would adjust retrieval scores based on various retrieval quality criteria.

We start by setting the theoretical grounds (and motivation) behind our approach. As a first step, we now build on top of Kurland et al.'s [14] probabilistic QPP framework. Using their framework, $p(D|q,r)$ can be derived as follows[1]:

$$p(D|q,r) \overset{def}{=} p(r|D) \sum_{d \in D} p(d|q,r)p(d|D,r). \tag{1}$$

$p(r|D)$ denotes the probability that D is relevant regardless of a specific query [14]. $p(r|D)$ may be estimated, for example, by analyzing properties of D that may indicate the existence of relevant information in D. Such properties may include, among others, D's cohesion (e.g., by measuring list diameter), its clustering tendency, diversity, etc [14]. $p(d|q,r)$ denotes the probability that document $d \in D$ provides a relevant answer to query q [14]. $p(d|q,r)$ may be estimated, for example, using d's (normalized) query likelihood [15]. Finally, $p(d|D,r)$ is the probability that document d is generated by the relevant document set D, representing the strength of d's association with D [14]. This probability may be estimated according to the likelihood of generating d from D's induced relevance model compared to the likelihood of generating it from the background model induced from C [14].

Next, noting that $p(d|q,r) \overset{def}{=} \frac{p(q|d,r)p(r|d)p(d)}{p(r|q)p(q)}$ and assuming that $p(q)$ is uniform, we now obtain:

$$p(D|q,r) \propto \frac{p(r|D)}{p(r|q)} \sum_{d \in D} p(q|d,r)p(r|d)p(d)p(d|D,r) \tag{2}$$

Here, $p(q|d,r)$ represents the query likelihood of document d [21], which is commonly estimated according to the observed score $s(d)$ determined by the underlying retrieval method $\mathcal{M}$. In many retrieval model implementations, such a score is usually determined according to the similarity between the query q and the document d. $p(r|q)$ denotes the probability that query q is a relevant representation of the (hidden) information need I_q. Estimating this probability is the primary goal of many pre-retrieval QPP methods [11].

Further assuming that documents $d \in D$ are uniformly distributed over D, i.e., $p(d) \overset{def}{=} \frac{1}{|D|}$; and noting that $s(d) \overset{def}{=} p(q|d) \approx p(q|d,r)$, we now obtain our first proposed general mean estimator:

$$p(D|q,r) \overset{def}{=} \frac{1}{|D|} \sum_{d \in D} s(d) \cdot \left[\frac{p(r|d)p(d|D,r)p(r|D)}{p(r|q)} \right] \tag{3}$$

3.2 Deriving alternative estimators

We next derive two variants of the general mean estimator in Eq. 3. The first, is a *query-sensitive* estimator, which emphasizes more on query-related properties that may govern document d's retrieval quality. On the other hand, the second is a *corpus-sensitive* estimator, which emphasizes more on corpus-related properties. These two variants are further derived from the prior document relevance probability $p(r|d)$. On the theoretical side, we wish to show that, putting more emphasis on query-related properties would mean that less attention can be given to corpus-related ones, and vice versa. This in turn, suggests the existence of **tradeoffs** in the design of such "general" mean estimators.

3.2.1 Query-sensitive estimator. To derive a query-sensitive variant of the proposed general mean estimator in Eq. 3, we first note that $p(r|d) \overset{def}{=} \frac{p(r|d,q)p(q|d)}{p(q|d,r)}$. Using again the assumption that $p(q|d) \approx p(q|d,r)$ we get that $p(r|d) \approx p(r|d,q)$. Putting this back into Eq. 3, we now obtain a query-sensitive mean estimator:

$$p(D|q,r) \overset{def}{=} \frac{1}{|D|} \sum_{d \in D} s(d) \cdot \left[\frac{p(r|d,q)p(d|D,r)p(r|D)}{p(r|q)} \right] \tag{4}$$

[1] See Eq. 4 in [14].

$p(r|d,q)$ serves as the basis of all *probabilistic relevance models*, commonly estimated in proportion to $\log \frac{p(d|r)}{p(d|\bar{r})}$, e.g., using the Okapi-BM25 document score [25].

Finally, please note that, since $p(r|D) \overset{def}{=} \frac{p(r|D,q)p(q|D)}{p(q|D,r)}$, we could potentially make this estimator variant even more sensitive to query-related issues.

3.2.2 Corpus-sensitive estimator. To derive a corpus-sensitive variant of the proposed general mean estimator in Eq. 3, we first note that $p(r|d) \overset{def}{=} \frac{p(r|d,C)p(C|d)}{p(C|d,r)}$. Putting this back into Eq. 3, we now obtain a corpus-sensitive mean estimator:

$$p(D|q,r) \overset{def}{=} \frac{1}{|D|}\sum_{d\in D} s(d)\cdot\left[\frac{p(r|d,C)p(C|d)p(d|D,r)p(r|D)}{p(r|q)p(C|d,r)}\right] \quad (5)$$

Here $p(r|d,C)$ denotes the probability that document d is the most relevant document in C (i.e., "most focused"), regardless of a specific query. In Section 5 we suggest one new option of how to estimate this probability. $p(C|d)$ models the relative importance of document d in C, which can be estimated, for example, by measuring the document's centrality in C (e.g., PageRank [19]). Finally, $p(C|d,r)$ further denotes the probability that a relevant document d belongs to corpus C. Since $p(C|d,r) \propto p(d|C,r)$, this term can be potentially estimated in a similar manner to $p(d|D,r)$. In this case, $p(d|C,r)$ will capture the association strength of document d with the (presumably) relevant corpus. Yet, compared to $p(d|D,r)$, in this case, the weaker such association is, the more we shall assume that document d is relevant.

Finally, similar to the query-sensitive case, noting that $p(r|D) \overset{def}{=} \frac{p(r|D,C)p(C|D)}{p(C|D,r)}$, we could potentially make this variant even more sensitive to corpus-related issues.

3.3 Calibrated mean retrieval score estimation

Estimating $p(D|q,r)$ according to either of the three proposed alternatives (i.e., Eq. 3, Eq. 4 or Eq. 5) dictates a generic estimation scheme that adjusts the original retrieval scores $s(d)$ according to several retrieval quality criteria. To better reflect this fact, we now further define the following (generic) calibrated version of the proposed estimators:

$$p(D|q,r) \overset{def}{=} \frac{1}{|D|}\sum_{d\in D} s(d)\cdot\phi_r(d), \quad (6)$$

where $\phi_r(d) \equiv \phi(r|q,C,D,d)$ is a (generic) score **calibration factor** that denotes the overall calibration that is applied on a given document score $s(d)$. The score calibrator $\phi_r(d)$ should basically consider all quality aspects that may affect document d's relevance during its retrieval process. Apart from considering document d's own properties, the properties of the query q, the corpus C and document d's associated result list D should be also considered. Therefore, the original document score $s(d)$ should be adjusted based on the likelihood that, at the end of this process, the "emitted" document d will be relevant. If the observed score $s(d)$ over- or underestimates this likelihood, then $\phi_r(d)$ should rescale $s(d)$ accordingly.

3.3.1 Score calibration as tradeoff analysis. We now make the observation that, the retrieval process of document d evaluated by the calibrator $\phi_r(d)$ is potentially a *multi-criteria decision making process.* Closer examination of the relationship between the three alternative mean estimators we previously derived, demonstrates that, capturing every possible retrieval quality aspect within $\phi_r(d)$ may be very hard, sometimes even impossible. This becomes more evident, for example, by examining the two query-sensitive (Eq. 4) and corpus-sensitive (Eq. 5) estimators, side by side. For example, putting more emphasis on corpus-related aspects would mean that we can put less emphasis on query-related aspects.

In an idle case, the retrieval process (implemented by the underlying retrieval method $\mathcal{M}$) provides the best possible response $D^* \subseteq C$ by considering all possible quality criteria that may govern an effective retrieval. Yet in practice, an effective retrieval process may be governed by several, possibly contradicting, quality aspects that need to be considered in parallel, e.g., relevance, retrievability, diversity, personalization, etc. Hence, not all retrieval quality criteria can be fully satisfied by the underlying method, and therefore, in many cases a **tradeoff** may exist among many of such criteria.

3.3.2 Discriminative score calibration. It becomes more evident that, it is eminently impossible to derive a (single) general generative mean estimator that can consider all possible retrieval quality criteria at once. Using different generative factors for the various retrieval components, will commonly instantiate a mean estimator that may prioritize the retrieval quality criteria in a different way. Every such instantiation further dictates the specific calibration $\phi_r(d)$ scheme that needs to be applied on D's document scores $s(d)$. Trying to overcome this "hurdle", we next describe an alternative way in which $\phi_r(d)$ can be effectively derived.

To this end, instead of calculating $\phi_r(d)$ directly in a generative form, we propose to calculate it in a discriminative fashion. Accordingly, we now associate with each retrieved document $d \in D$ a set of features $F(d) = \{f_1(d), \ldots, f_h(d)\}$. Each feature $f_j(d) \in F(d)$ is assumed to encode some unique aspect of the document's retrieval that might affect its actual relevance. Let $\phi_{r,F}(d)$ further denote this discriminative version of the calibrator. $\phi_{r,F}(d)$ now gets $F(d)$ as an input and outputs the calibration value for $s(d)$. Whenever $|F(d)| = \emptyset$, we get the null calibrator which is simply defined as $\phi_{r,\emptyset}(d) \overset{def}{=} 1$ (meaning we estimate the mean score using only the original document scores $s(d)$).

3.3.3 $\phi_{r,F}(d)$ implementation. Since the calibration problem by itself is a multi-criteria decision problem, we now propose to derive $\phi_{r,F}(d)$ using the *Weighted Product Model* [17] (WPM for short). WPM is a general multi-criteria decision analysis (MCDA) approach that can be used to combine different decision criteria. In our case, each calibration feature $f_j(d) \in F(d)$ defines a single decision criterion. Our goal is, therefore, to find the best criteria combination policy. Based on WPM, such a policy is defined by a set of (non-negative) real weights, having weight $\alpha_j \geq 0$ $(1 \leq j \leq h)$ model the relative importance of feature $f_j(d) \in F(d)$. We note that, the weights do not depend on a specific document, but rather on the feature type.

Overall, the calibrator $\phi_{r,F}(d)$ is implemented by calculating the weighted product of the feature values: $\phi_{r,F}(d) \overset{def}{=} \prod_{j=1}^{h} \left(f_j(d)\right)^{\alpha_j}$. Note that, zeroing all weights also results in null (no) calibration.

Finally, "plugging" the defined $\phi_{r,F}(d)$ calibrator back into Eq. 6, we obtain our proposed (general) mean score estimator, hereinafter, termed **WPM** estimator. Later on, in Section 5 we shall propose two instantiations of WPM.

4 DERIVING PREVIOUS PREDICTORS

We next demonstrate that, several previously suggested QPP methods, namely Clarity [7], WIG [32] and SMV [31], can be directly derived as private instantiations of our calibrated-mean score estimation approach. Using this connection in mind, therefore, allows us to obtain and study a preliminary set of calibration features $F(d)$ that can be utilized within our approach.

4.1 Deriving Clarity

The Clarity [7] method estimates query performance proportionally to the divergence between a relevance language model [15] induced from D and that induced from C. Higher divergence is assumed to imply that D is more "focused" [7].

Given D, Clarity's prediction is calculated as follows:

$$p_{Clarity}(D|q,r) \overset{def}{=} \sum_{w} p(w|R) \log \frac{p(w|R)}{p(w|C)}, \tag{7}$$

where $p(w|R) \overset{def}{=} \sum_{d \in D} p(d)p(w|d)p(q|d)$ denotes the likelihood that term w is generated by the relevance model R induced from D [15]. Using again $s(d) \overset{def}{=} p(q|d)$ and $p(d) = \frac{1}{|D|}$, we get that:

$$p_{Clarity}(D|q,r) = \frac{1}{|D|} \sum_{d \in D} s(d) \cdot \left[\sum_{w} p(w|d) \log \frac{p(w|R)}{p(w|C)} \right] \tag{8}$$

Next, let $p(w|d) \overset{def}{=} \frac{c(w,d)}{|d|}$, where $c(w,x)$ denotes term w's occurrence count in text x and $|d| \overset{def}{=} \sum_{w \in d} c(w,d)$ is document d's length. We now further obtain:

$$\begin{aligned} p_{Clarity}(D|q,r) &= \frac{1}{|D|} \sum_{d \in D} s(d) \cdot \left[\sum_{w} \frac{c(w,d)}{|d|} \log \frac{p(w|R)}{p(w|C)} \right] \\ &= \frac{1}{|D|} \sum_{d \in D} s(d) \cdot \left[\frac{1}{|d|} \log \frac{\prod_{w} p(w|R)^{c(w,d)}}{\prod_{w} p(w|C)^{c(w,d)}} \right] \\ &\overset{def}{=} \frac{1}{|D|} \sum_{d \in D} s(d) \cdot \left[\frac{1}{|d|} \cdot \log \frac{p(d|R)}{p(d|C)} \right]. \end{aligned} \tag{9}$$

Hence, using the following equal feature weights $\alpha_1 = \alpha_2 = 1$ yields that, Clarity is basically a private case of our calibrated mean score estimator. In Clarity's case, two calibration features are utilized as follows. The first feature, $f_1(d) = \frac{1}{|d|}$ (denoted **invDocLen**), calibrates the document score $s(d)$ by scaling it reversely to its document length $|d|$. Therefore, such calibration prefers shorter documents to longer ones. The shorter the document is, the higher its chance for being retrieved, regardless of a specific query in mind [2]; whereas, longer documents have a higher chance of being relevant. Therefore, in this case: $f_1(d) \propto \frac{1}{p(r|d)}$.

The second feature, $f_2(d) = \log \frac{p(d|R)}{p(d|C)}$ (denoted **dLogRel**), calibrates $s(d)$ according to the log-likelihood ratio between d's generation from the relevance model induced from D to its generation from the background model induced from C. Therefore, $f_2(d) \propto p(d|D,r)$, and $s(d)$ is calibrated based on d's association strength with D [14], where the later is assumed by Clarity to be relevant [7].

All in all, it becomes apparent that, Clarity's calibration scheme aims at capturing the tradeoff between relevance and retrievability [2]. In Clarity, such tradeoff is implemented by a simple product of a couple of calibration features. As will be shown in our evaluation, a simple extension of Clarity with a weighted calibration scheme for its two identified basic features, leads to a significantly better query performance prediction.

4.2 Deriving WIG

The WIG method [32] estimates query performance according to the difference between the average retrieval score in D and that of C. The larger the difference is, the better query performance is assumed to be [32]. WIG prediction is given as follows:

$$p_{WIG}(D|q,r) = \frac{1}{\sqrt{|q|}} \cdot \frac{1}{|D|} \sum_{d \in D} \left(s(d) - s(C)\right), \tag{10}$$

where $s(C) \approx p(q|C)$ denotes the corpus query likelihood [32]. $s(C)$ can be calculated, for example, by treating the corpus as a single document [32].

Noting that $s(d) - s(C) = s(d) \cdot \left(1 - \frac{s(C)}{s(d)}\right)$, we can rewrite:

$$p_{WIG}(D|q,r) = \frac{1}{|D|} \sum_{d \in D} s(d) \cdot \left[\frac{1}{\sqrt{|q|}} \cdot \left(1 - \frac{s(C)}{s(d)}\right) \right] \tag{11}$$

Therefore, fixing again $\alpha_1 = \alpha_2 = 1$, we observe that, WIG is also a private case of our proposed estimator. Moreover, similar to Clarity, WIG utilizes two (yet different) calibration features. The first feature $f_1(d) = \frac{1}{\sqrt{|q|}}$ (denoted **invQLen**) is document-independent, which calibrates $s(d)$ reversely to query q's (scaled) length $|q| \overset{def}{=} \sum_{w \in q} c(w,q)$. In this case $f_1(d) \propto p(r|q)$, where the longer the query is, the more difficult it may be to answer such a query [11]. In accordance, $f_1(d)$ down-scales $s(d)$ more for longer queries; regardless of the document's identity.

The second feature, $f_2(d) = 1 - \frac{s(C)}{s(d)}$ (denoted **invCd**), is document-dependent and shares resemblance with the second calibration feature of Clarity. In this case $f_2(d) \propto \frac{1}{p(C|d,r)} \propto \frac{1}{p(d|C,r)}$ (see $p(C|d,r)$ derivation in Eq. 5). As we already noted, the probability $p(d|C,r)$ shares the same motivation with the definition of the $p(d|D,r)$ term. Therefore, $f_2(d)$ estimates document d's association strength with the corpus, when the later is assumed to be relevant. The higher such association is, the more difficult it would be to separate it from the corpus, and hence, the document score will be down-scaled. When $s(C) \to 0$, then $f_2(d) \to 1$, which implies that no association is evident between the document and the corpus. Therefore, in this case, $s(d)$ shall remain unaffected.

Similar to Clarity, WIG's performance can be further boosted by tuning the weights of its two calibration features.

4.3 Deriving SMV

Another QPP method that is based on assessing score magnitudes is the SMV method [31], whose prediction can be directly expressed using our approach as follows:

$$p_{SMV}(D|q,r) = \frac{1}{|D|}\sum_{d\in D} s(d)\cdot\left[\frac{1}{|s(C)|}\cdot\left|\ln\frac{s(d)}{\hat{\mu}_D}\right|\right], \qquad (12)$$

where $\hat{\mu}_D = \frac{1}{|D|}\sum\limits_{d\in D} s(d)$ denotes D's mean document score.

Similar to the two previous methods, SMV also utilizes two (yet again, different) calibration features (with $\alpha_1 = \alpha_2 = 1$). The first feature is $f_1(d) = \frac{1}{|s(C)|}$ (denoted **invCS**), which is document-independent. Here, $f_1(d)$ inversely rescales the document score according to the corpus's similarity to the query. Hence, in this case $f_1(d) \propto \frac{1}{p(r|q,C)}$. The higher the similarity $s(C)$ is, the more documents in C may be similar to q, and hence, the more difficult it would be to point out that document d is the one relevant to q. Therefore, higher corpus-query similarity will down-scale $s(d)$.

The second feature $f_2(d) = \left|\ln\frac{s(d)}{\hat{\mu}}\right|$ (denoted **invSD**), is document-dependent, which adjusts the document's score $s(d)$ relatively to its absolute "divergence" from the mean score $\hat{\mu}_D$. In this case, the mean score $\hat{\mu}_D$ which represents the score of D's centroid [30] is assumed to capture the uncertainty in whether documents in D are either relevant or not. Hence, $f_2(d) \propto \frac{1}{p(d|D,q)}$, the more $s(d)$ is different from $\hat{\mu}_D$, the less uncertainty is associated with $s(d)$ [30].

Finally, similar to the two previous methods, tuning SMV's two feature weights can further boost its performance.

5 WPM ESTIMATOR INSTANTIATIONS

We now propose two calibration feature sets, used for instantiating the WPM estimator that we derived in Section 3. As the first feature set, we reutilize the six features that we identified and studied in Section 4, namely: **invDocLen**, **dLogRel**, **invQLen**, **invCd**, **invCS** and **invSD**. We now denote the resulting estimator instance as **WPM1**. By learning WPM1's feature weights, we expect to obtain a more accurate estimator.

As a second instantiation, denoted **WPM2**, we utilize an additional set of calibration features to that of WPM1. Our purpose here is not to design or explore many such features, but rather to demonstrate, as a proof of concept, that, with more features, a further improvement may be achieved. We only choose additional features that are **document-level dependent**, which we believe are more interesting to explore in the context of our work. We note again that, various combinations of other feature types (i.e., ones that aim to capture $p(r|D)$, $p(r|C)$ or additional $p(r|q)$ features), were already studied by several previous works [3, 5, 23, 29].

Overall, we use the following four additional features in WPM2. The first feature **dEnt**, borrowed from [22], captures $p(r|d)$ and estimates the document's content diversity according to its induced entropy: $-\sum\limits_{w\in d} p(w|d)\log p(w|d)$. The second feature **dClarity**, also borrowed from [22], captures $p(d|C,r)$ which estimates the focus of d according to the KL divergence: $\sum\limits_{w\in d} p(w|d)\log\frac{p(w|d)}{p(w|C)}$. The third feature **BM25**, captures $p(r|q,d)$ and is simply calculated as the Okapi-BM25 [25] score of document d given q.

Finally, the last feature **dCFocus**, aims at capturing $p(r|d,C)$, i.e., the probability that document d is the **most focused document in** C. Document d's relative focus is estimated by measuring to what extent d's term saliency "agrees" with C's global term importance. To this end, we now define the normalized inverse document frequency $nidf(w) = \frac{idf(w)}{\sum\limits_{w'} idf(w')}$. **dCFocus** is then calculated as the KL-divergence based similarity: $\exp\left(-\sum\limits_{w\in d} p(w|d)\log\frac{p(w|d)}{nidf(w)}\right)$.

6 EVALUATION

6.1 Datasets

Corpus	#documents	Queries	Disks
AP	242,918	51-150	1-3
TREC4	567,529	201-250	2-3
TREC5	524,929	251-300	2&4
ROBUST	528,155	301-450, 601-700	4&5-{CR}
WT10g	1,692,096	451-550	WT10g
GOV2	25,205,179	701-850	GOV2

Table 1: TREC benchmarks used for experiments.

The TREC corpora and queries used for the evaluation are specified in Table 1. These benchmarks were used by many previous QPP works [4]. Titles of TREC topics were used as queries, except for the TREC4 benchmark, where no titles are available and topic descriptions were used instead. The Apache Lucene[2] open source search library was used for indexing and searching documents. Documents and queries were processed using Lucene's English text analysis (i.e., tokenization, Porter stemming, stopwords, etc.). As the underlying retrieval method $\mathcal{M}$ we used Lucene's Dirichlet-smoothed query-likelihood implementation. Following previous works [14, 23, 28, 30], we fixed the Dirichlet parameter to $\mu = 1000$.

6.2 Baselines

We compared the two instantiations of the WPM estimator, **WPM1** and **WPM2**, with several different baseline methods. As a trivial baseline, we considered D's original mean score (denoted **Mean**(org)). It was simply implemented using the null calibrator $\phi_{r,\emptyset}(d)$.

As a first line of baselines we considered **Clarity** [7], **WIG** [32] and **SMV** [31], whose details were already discussed in Section 4. As we have shown, each of these baselines can be directly derived as a private instance of the WPM estimator employed with two different calibration features whose weights equal to 1. For each method, we further implemented its calibrated (weighted) version which utilized the same pair of features, except for the weights which were treated as free parameters. This resulted in three more corresponding calibrated baselines, which we further refer to as **C-Clarity**, **C-WIG** and **C-SMV**.

[2] http://lucene.apache.org

Next, we implemented several competitive baseline methods as follows. The first is the **ImpClarity** [12] method, a variant of Clarity, which given a parameter t, induces a relevance model from D using only those terms w that appear in less than t% of the documents in C. We also implement **NQC** [30], a method that shares resemblance to **SMV** and is commonly used as a strong baseline that also utilizes retrieval scores. **NQC** predicts query performance according to the standard deviation of (D's documents) retrieval scores $s(d)$. Higher deviation is assumed to testify for lower chance of query drift, hence better performance [30]. Similar to **SMV**, **NQC** further normalizes the standard deviation by the corpus query likelihood $s(C)$. As another common strong baseline, we also implemented the **QF** method [32], which was proposed as an alternative to **WIG** by the same authors of [32]. **QF** predicts query performance according to the overlap between D and another list $D' \subseteq C$, obtained by evaluating a new (weighted) query q' over C. q' is formulated from the top-n terms with the highest contribution to the KL-divergence between the relevance model induced from D and the background (corpus) model. The higher the overlap is (which is simply measured as $|D \cap D'|$), the better the performance is predicted to be [32].

As another (strong) alternative, we also implemented the **UEF** method [28]. **UEF** predicts the query performance of a given result list D based on the combination of two features. The first, is the similarity of D with its re-ranked version π_D (measured using Pearson's-ρ correlation between scores [28]). π_D is obtained by scoring documents in D according to a relevance model (RM1) induced from $D^{[m]} \subseteq D$ - the top-m scored documents in D. Higher similarity is assumed to result in a better performance [28]. The second feature is the estimated performance of $D^{[m]}$ itself. For this estimation, any baseline QPP method can be applied on $D^{[m]}$ [28]. The final prediction is obtained by multiplying both feature values [28]. We employed **UEF** with Clarity, WIG and SMV as its baseline methods and obtained three corresponding UEF variants: **UEF**[Clarity], **UEF**[WIG] and **UEF**[SMV].

Finally, we further implemented the **LTRoq** method [23], a predictor inspired by the Markov Random Field (MRF) model. **LTRoq** combines several list-level post-retrieval QPP features (e.g., Clarity, WIG, NQC, UEF, etc) with several pre-retrieval features [23] (e.g., various variants of SCQ, VAR, and IDF [11]). **LTRoq** learns to combine the various QPP features using SVM^{rank} [23]. Therefore, we consider **LTRoq** as a very strong baseline.

6.3 Setup

We predicted the performance of each query based on its top-1000 retrieved documents [4]. Following the common practice [4], we assessed prediction over queries quality according to the correlation between the predictor's values and the actual average precision (AP@1000) values calculated using TREC's relevance judgments. To this end, we report the Pearson's-ρ (P-ρ) and Kendall's-τ (K-τ) correlations which are the most common measures [4].

Most of the methods that we evaluated (including WPM variants) required to tune some free parameters. Common to all methods is the free parameter $k \overset{def}{=} |D|$, which is the number of top scored documents (out of a total of 1000 retrieved documents) **to be used for the prediction**. To this end, for each method we selected $k \in \{5, 10, 20, 50, 100, 150, 200, 500, 1000\}$.

Next, some of the methods we evaluated required additional parameters to tune. For example, **Clarity**, **ImpClarity**, **QF** and **UEF** variants all utilize a relevance model (RM1) that is induced from D. For the first two, we used all documents in D [7, 12], while for **QF** and **UEF** variants we only used the top-m scored docs in D [28, 32] (i.e., $D^{[m]}$), with $m \in \{1, 3, 5, \ldots, |D|\}$. Following the common practice [7, 12, 28, 32], in all these methods, we further clipped the induced relevance model at the top-n terms cutoff, with $n \in \{5, 10, 20, 50, 100, 150, 200\}$. For **ImpClarity**, its term selection parameter t was further selected as follows: $t \in \{1, 2, 3, 5, 10\}$.

To implement **LTRoq**, we used the same set of post-retrieval and pre-retrieval features[3] that was used in [23]. For training **LTRoq**, we further closely followed [23]'s two-phase learning approach[4].

To have a direct way of measuring the impact of feature calibration (weighing) in our WPM variants (i.e., **C-Clarity**, **C-WIG**, **C-SMV**, **WPM1** and **WPM2**), we fixed the $\langle k, n \rangle$ parameters to the same configuration that was initially learned for the non-calibrated baselines (i.e., **Clarity**, **WIG** and **SMV**). For the **dEnt**, **dClarity** and **dCFocus** features utilized by WPM2, we further tuned the number of top-l terms in d (i.e., term cutoff according to $p(w|d)$) that should be considered for each feature calculation, with $l \in \{5, 10, 20, 50, 100\}$. For the **BM25** feature we used its common parameters $k1 = 1.2, b = 0.75$ [25]. We further smoothed each feature used by the WPM variants as follows $f_j(d; \epsilon) \overset{def}{=} \max(f_j(d), \epsilon)$, where $\epsilon = 10^{-10}$ is a hyperparameter.

To learn the calibration feature weights of the WPM variants, we used a Coordinate Ascent approach [16]. To this end, we selected the feature weights $\{\alpha_j\}_{j=1}^{h}$ in the grid $[0, 5]^h$, assuming there are $h \in \{2, 6, 10\}$ such different features, depending on the WPM variant (with a step size of 0.1 within each dimension).

Training and testing of all methods was performed similarly to previous works [23, 29, 30] using an holdout (2-fold cross validation) approach. Accordingly, on each benchmark, we generated 30 random splits of the query set; each split had two folds. The first fold was used as the (query) train set, where parameters were tuned to maximize P-ρ. The second fold was kept untouched for testing. We recorded the average prediction quality (i.e., P-ρ and K-τ) over the 30 splits. Finally, we measured statistical significant differences of prediction quality using a two-tailed paired t-test with $p < 0.05$ computed over all 30 splits (with a Bonferroni correction whenever more than two methods were compared).

6.4 Results

6.4.1 Impact of score calibration. Table 2 depicts the results of the comparison of each non-calibrated version of **Clarity**, **WIG** and **SMV** (i.e., all weights α_j are set to the value of 1) with its corresponding calibrated (weighted) version, i.e., **C-Clarity**, **C-WIG** and **C-SMV**. For each calibrated version we also report the relative improvement in prediction quality over its corresponding non-calibrated version.

[3] The full set of features is described in Section 3.5.1 in [23].

[4] Due to space considerations, we cannot describe in details this learning approach. The reader is kindly encouraged to refer to Section 4.1.1 in [23] for more details.

	AP		TREC4		TREC5		Robust		WT10g		GOV2	
Method	P−ρ	K−τ	P−ρ	K−τ	P−ρ	K−τ	P−ρ	K−τ	P−ρ	K−τ	P−ρ	K−τ
Mean(org)	.383	.326	.356	.340	.215	.152	.314	.290	.375	.290	.411	.297
Clarity	.596	.428	.456	.380	.490	.258	.477	.328	.380	.240	.407	.305
C-Clarity	.626* (+5.0%)	.447* (+4.4%)	.537* (+17.8%)	.392 (+3.3%)	.492 (+0.4%)	.271 (+5.0%)	.532* (+11.5%)	.370* (+12.8%)	.456* (+20%)	.313* (+30.4%)	.435* (+6.9%)	.323* (+5.9%)
WIG	.526	.380	.533	.502	.347	.252	.411	.358	.434	.364	.535	.387
C-WIG	.672* (+27.8%)	.400* (+5.3%)	.561* (+5.2%)	.535* (+6.5%)	.375* (+8.1%)	.278* (+10.3%)	.551* (+34.1%)	.383* (+7.0%)	.454* (+4.6%)	.374* (+2.74%)	.550* (+2.8%)	.391* (+1.0%)
SMV	.631	.398	.524	.499	.459	.268	.586	.432	.292	.206	.418	.304
C-SMV	.668* (+5.9%)	.450* (+13.1%)	.572* (+9.2%)	.541* (+8.4%)	.483* (+5.3%)	.283* (+4.4%)	.601* (+2.6%)	.433 (+0.2%)	.432* (+47.9%)	.330* (+60.2%)	.589* (+40.9%)	.423* (+39.1%)

Table 2: Comparison between each non-calibrated baseline method and its corresponding calibrated version. Percentages reported in parentheses "()" represent the relative change in quality of each calibrated version over its corresponding non-calibrated version. The superscript $*$ further denotes a statistically significant difference between the two ($p < 0.05$).

	AP		TREC4		TREC5		Robust		WT10g		GOV2	
Method	P−ρ	K−τ	P−ρ	K−τ	P−ρ	K−τ	P−ρ	K−τ	P−ρ	K−τ	P−ρ	K−τ
ImpClarity	.582	.418	.502	.389	.501	.291	.514	.354	.422	.272	.520	.359
NQC	.554	.361	.624	.562	.483	.318	.575	.406	.486	.354	.432	.304
QF	.575	.385	.632	.570	.413	.378	.483	.371	.436	.343	.515	.383
UEF[Clarity]	.620	.435	.618	.532	.554	.327	.560	.387	.483	.360	.427	.307
UEF[WIG]	.574	.389	.625	.542	.507	.308	.542	.384	.450	.371	.527	.361
UEF[SMV]	.652	.427	.568	.527	.537	.317	.594	.430	.358	.229	.571	.389
LTRoq	.684	.506	.653	.588	.566	.345	.570	.391	.367	.312	.582	.410
WPM1	.730^c_b (+8.6%) [+6.7%]	.539^c_b (+19.8%) [+6.5%]	.682^c_b (+19.2%) [+4.4%]	.600^c (+10.9%) [+2.0%]	.702^c_b (+42.7%) [+24.0%]	.386^c (+37.0%) [+2.1%]	.628^c_b (+4.5%) [+5.7%]	.464^c_b (+7.2%) [+7.9%]	.497^c_b (+9.0%) [+2.3%]	.378 (+1.1%) [+1.9%]	.634^c_b (+7.6%) [+8.9%]	.462^c_b (+9.2%) [+12.7%]
WPM2	**.738**^c_b (+9.8%) [+7.9%]	**.553**^c_b (+22.9%) [+9.3%]	**.702**^c_b (+22.7%) [+7.5%]	**.613**^c_b (+13.3%) [+4.2%]	**.738**^c_b (+50.0%) [+30.4%]	**.412**^{c} (+45.6%) [+3.0%]	**.640**^c_b (+6.5%) [+7.7%]	**.495**^c_b (+14.3%) [+15.1%]	**.540**^c_b (+18.4%) [+11.1%]	**.433**^c_b (+15.8%) [+16.7%]	**.655**^c_b (+11.2%) [+12.5%]	**.478**^c_b (+13.0%) [+16.6%]

Table 3: Comparison between WPM1 and WPM2 and the alternative baseline methods. Percentages reported in parentheses "()" represent the relative change in quality of each WPM instance over the best calibrated baseline (i.e., either C-Clarity, C-WIG or C-SMV). Percentages reported in brackets "[]" further represent the relative change in quality of each WPM instance over the best alternative baseline. The superscript c and subscript b further denote a statistically significant difference of WPM1/WPM2 with the calibrated and alternative baselines, respectively (Bonferroni corrected for $p < 0.05$).

Overall, calibrating all three methods by tuning the two calibration feature weights of each method resulted in an enhanced prediction quality. The average improvement, regardless of specific correlation measure, was 10.3(±2.5)%, 9.5(±3.0)% and 19.7(±6.0)% for **C-Clarity**, **C-WIG** and **C-SMV**, respectively. This attests the merits of using calibration within these methods. By better tuning their identified calibration features, tradeoffs that exist within the core design of these methods could be better handled.

Table 3 further reports the prediction quality of all other baselines we implemented (including **WPM1** and **WPM2**). A comparison of **C-Clarity**, **C-WIG**, and **C-SMV** with **ImpClarity**, **NQC**, **QF** and the **UEF** variants, reveals that, each one of the three calibrated versions, in at least 50% of the usecases, resulted in a better prediction quality than that of its potential alternative (i.e., **C-Clarity** vs. **ImpClarity** and **UEF**[Clarity]; **C-WIG** vs. **QF** and **UEF**[WIG]; **C-SMV** vs. **NQC** and **UEF**[SMV]). This is yet another empirical testimony to the potential of calibration. We next investigate this potential more closely.

6.4.2 WPM vs. alternative baselines. We now compare **WPM1** and **WPM2** (whose details were described in Section 5) with all the other baselines. To recall, **WPM1** uses the super-set of calibration features (six in total) of the calibrated baseline methods **C-Clarity**, **C-WIG** and **C-SMV**. **WPM2** further uses 4 additional features. For **WPM1** and **WPM2** we also report the relative improvement over the **best calibrated baseline** (i.e., out of **C-Clarity**, **C-WIG** and **C-SMV**) and the relative improvement over the **best alternative baseline** in Table 3 (excluding **WPM1** and **WPM2**).

First, comparing **WPM1** side-by-side with the three calibrated baselines, shows a significant boost in prediction quality over these methods (an average improvement of 15.0(±4.0)% regardless of the correlation measure). This is yet another empirical proof that, considering more tradeoffs that govern an effective retrieval of a given document, results in an enhanced prediction quality. This is further supported by examining **WPM2** side-by-side with **WPM1**. **WPM2**'s additional calibration features provide further improvement (an average additional improvement of 3.8(±1.1)% regardless

AP	TREC4	TREC5	Robust	WT10g	GOV2
dLogRel	invSD	invCd	invCS	dEnt	dEnt
invCS	invCS	invSD	invSD	dLogRel	invCd
invSD	dCFocus	invDLen	invDLen	BM25	invCS
BM25	invDLen	dLogRel	dLogRel	dCFocus	invQLen
invCd	invCd	invCS	BM25	invCd	dClarity

Table 4: The top-5 calibration features with the highest contribution to WPM2's prediction quality.

of the correlation measure). This actually comes with no surprise, as **WPM2** considers even more retrieval effectiveness tradeoffs that may govern an effective document retrieval.

We next compare **WPM1** and **WPM2** with the other baselines in Table 3. As can be observed, both WPM instances **completely outperformed all other baselines** (significantly in most cases). The average improvement over the **best alternative baseline** was 7.8(±1.9)% for **WPM1** and 13.0(±1.9)% for **WPM2**, regardless of the correlation measure. Quite notable is the significant difference with the **LTRoq** method, one of the strongest baselines in the QPP literature to date that also utilizes supervised learning. We attribute this difference to the fact that WPM learns to combine document-level features for QPP; this in comparison to **LTRoq**, which similarly to previous supervised approaches [3, 5], uses only list-level post-retrieval and pre-retrieval features [23].

6.4.3 Calibration feature analysis. Table 4 reports the top-5 calibration features with the highest contribution to **WPM2**'s prediction quality[5]. Among these features, both **invCS** and **invCd** are the most notable (appearing in 5 out of the 6 top-5 lists). This suggests that, corpus-sensitivity should play a major role in the design of retrieval score calibrators.

The next significant features are **invSD** and **dLogRel** (appearing in 4 out of the 6 top-5 lists). This suggests that, the next line of calibration features that should get closer attention are those that consider the association of a given document d with its containing result list D. A score of a document that is associated with an effective list (captured by **dLogRel**) should be trusted more than one that is associated with an ineffective list (captured by **invSD**). Another feature that "stands out" in Table 4 is **BM25**. Such feature importance implies that, rather than using only document scores that were obtained from a single retrieval method, it would be better to use also document scores that were obtained from another (preferably indepedent) retrieval method. Indeed, some of previously suggested QPP methods share a similar motivation [1].

Finally, another notable feature is **dEnt**, which appears to be (very) important in Web corpora. This strongly supports the merits of considering content diversification aspects in the design of QPP methods for Web corpora [18].

7 CONCLUSIONS

Our empirical evaluation has served as a solid evidence on the effectiveness of our proposed discriminative calibrated mean estimator. By re-designing several of previously suggested QPP methods according to our approach, we were able to significantly improve their prediction quality. We also demonstrated the merits of using additional and diverse calibration features within our approach, which resulted in a prediction quality which outperformed that of several strong alternative baselines.

[5] Features are ordered according to their relative contribution and were selected using a sequential forward selection approach.

REFERENCES

[1] Javed A. Aslam and Virgil Pavlu. Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In *Proceedings of ECIR'07*.

[2] Leif Azzopardi and Vishwa Vinay. Retrievability: An evaluation measure for higher order information access tasks. In *Proceedings of CIKM '08*.

[3] Niranjan Balasubramanian, Giridhar Kumaran, and Vitor R. Carvalho. Predicting query performance on the web. In *Proceedings of SIGIR '10*.

[4] David Carmel and Oren Kurland. Query performance prediction for ir. In *Proceedings of SIGIR '12*.

[5] Kevyn Collins-Thompson and Paul N. Bennett. Predicting query performance via classification. In *Proceedings of ECIR'2010*.

[6] William S. Cooper, Fredric C. Gey, and Daniel P. Dabney. Probabilistic retrieval based on staged logistic regression. In *Proceedings of SIGIR '92*.

[7] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *Proceedings of SIGIR '02*.

[8] Ronan Cummins. Document score distribution models for query performance inference and prediction. *ACM Trans. Inf. Syst.*, 32(1):2:1–2:28, January 2014.

[9] Ronan Cummins, Joemon Jose, and Colm O'Riordan. Improved query performance prediction using standard deviation. In *Proceedings of SIGIR '11*.

[10] Keshi Dai. *Modeling Score Distributions for Information Retrieval*. PhD thesis, Boston, MA, USA, 2012. AAI3542649.

[11] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. A survey of pre-retrieval query performance predictors. In *Proceedings of CIKM '08*.

[12] Claudia Hauff, Vanessa Murdock, and Ricardo Baeza-Yates. Improved query difficulty prediction for the web. In *Proceedings of CIKM '08*.

[13] Evangelos Kanoulas, Virgil Pavlu, Keshi Dai, and Javed A Aslam. Modeling the score distributions of relevant and non-relevant documents. In *Conference on the Theory of Information Retrieval*, pages 152–163. Springer, 2009.

[14] Oren Kurland, Anna Shtok, Shay Hummel, Fiana Raiber, David Carmel, and Ofri Rom. Back to the roots: A probabilistic framework for query-performance prediction. In *Proceedings of CIKM '12*.

[15] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of SIGIR '01*.

[16] Donald Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Inf. Retr.*, 10(3):257–274, June 2007.

[17] David William Miller et al. Executive decisions and operations research. 1963.

[18] A. M. Ozdemiray and Ismail S. Altingovde. Query performance prediction for aspect weighting in search result diversification. *In Proceedings of CIKM '14*.

[19] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web, 1999.

[20] Joaquín Pérez-Iglesias and Lourdes Araujo. Standard deviation as a query hardness estimator. In *Proceedings of SPIRE'10*.

[21] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR '98*.

[22] Fiana Raiber and Oren Kurland. On identifying representative relevant documents. In *Proceedings of CIKM '10*.

[23] Fiana Raiber and Oren Kurland. Query-performance prediction: Setting the expectations straight. In *Proceedings of SIGIR '14*.

[24] Fiana Raiber and Oren Kurland. Using document-quality measures to predict web-search effectiveness. In *Proceedings of ECIR'13*.

[25] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009.

[26] Haggai Roitman. An enhanced approach to query performance prediction using reference lists. In *Proceedings of SIGIR '17*.

[27] Haggai Roitman, Shai Erera, and Bar Weiner. Robust standard deviation estimation for query performance prediction. In *Proceedings of ICTIR '17*.

[28] Anna Shtok, Oren Kurland, and David Carmel. Using statistical decision theory and relevance models for query-performance prediction. In *Proceedings of SIGIR '10*.

[29] Anna Shtok, Oren Kurland, and David Carmel. Query performance prediction using reference lists. *ACM Trans. Inf. Syst.*, 34(4):19:1–19:34, June 2016.

[30] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. Predicting query performance by query-drift estimation. *ACM Trans. Inf. Syst.*, 30(2):11:1–11:35, May 2012.

[31] Yongquan Tao and Shengli Wu. Query performance prediction by considering score magnitude and variance together. In *Proceedings of CIKM '14*.

[32] Yun Zhou and W. Bruce Croft. Query performance prediction in web search environments. In *Proceedings of SIGIR '07*.