Query Performance Prediction for Pseudo-Feedback-Based Retrieval

Haggai Roitman
IBM Research – Haifa
haggai@il.ibm.com

Oren Kurland
Technion – Israel Institute of Technology
kurland@ie.technion.ac.il

ABSTRACT

The query performance prediction task (QPP) is estimating retrieval effectiveness in the absence of relevance judgments. Prior work has focused on prediction for retrieval methods based on surface level query-document similarities (e.g., query likelihood). We address the prediction challenge for pseudo-feedback-based retrieval methods which utilize an initial retrieval to induce a new query model; the query model is then used for a second (final) retrieval. Our suggested approach accounts for the presumed effectiveness of the initially retrieved list, its similarity with the final retrieved list and properties of the latter. Empirical evaluation demonstrates the clear merits of our approach.

ACM Reference Format:
Haggai Roitman and Oren Kurland. 2019. Query Performance Prediction for Pseudo-Feedback-Based Retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331369

1 INTRODUCTION

The query performance prediction task (QPP) has attracted much research attention [2]. The goal is to evaluate search effectiveness with no relevance judgments. Almost all existing QPP methods are (implicitly or explicitly) based on the assumption that retrieval is performed using (only) document-query surface-level similarities [2]; e.g., standard language-model-based retrieval or Okapi BM25.

We address the QPP challenge for a different, common, retrieval paradigm: pseudo-feedback-based retrieval [3]. That is, an initial search is performed for a query. Then, top-retrieved documents, considered pseudo relevant, are utilized to induce a query model (e.g., expanded query form) that is used for a second (final) retrieval.

Thus, in contrast to the single-retrieval setting addressed in almost all prior work on QPP, here the effectiveness of the final result list presented to the user depends not only on the retrieval used to produce it (e.g., properties of the induced query model), but also on the initial retrieval using which the query model was induced. A case in point, if the initial retrieval is poor, it is highly unlikely that the final result list will be effective regardless of the query-model induction approach employed. Accordingly, our novel approach for QPP for pseudo-feedback-based retrieval accounts for the presumed effectiveness of the initially retrieved list, its association with the final retrieved list and properties of the latter.

Empirical evaluation shows that the prediction quality of our approach substantially transcends that of state-of-the-art prediction methods adopted for the pseudo-feedback-based retrieval setting — the practice in prior work on QPP for pseudo-feedback-based retrieval [6, 14, 15].

2 RELATED WORK

In prior work on QPP for pseudo-feedback-based retrieval, existing predictors were applied either to the final retrieved list [6, 14] or to the initially retrieved list [15]. We show that our prediction model, which incorporates prediction for both lists and accounts for their association, substantially outperforms these prior approaches.

The selective query expansion task (e.g., [1, 5, 13]) is to decide whether to use the pseudo-feedback-based query model, or stick to the original query. In contrast, we predict performance for a list retrieved using the query model.

In several prediction methods, a result list retrieved using a pseudo-feedback-based query model is used to predict the performance of the initially retrieved list [15, 20]. In contrast, our goal is to predict the effectiveness of the final result list; to that end, we also use prediction performed for the initial list.

3 PREDICTION FRAMEWORK

Suppose that some initial search is applied in response to a query q over a document corpus $\mathcal{D}$. Let D_{init} be the list of the k most highly ranked documents. Information induced from the top documents in D_{init} is used for creating a new query model (e.g., expanded query) used for a second retrieval; i.e., these documents are treated as pseudo relevant. We use D_{scnd} to denote the result list, presented to the user who issued q, of the k documents most highly ranked in the second retrieval.

Our goal is to predict the effectiveness of D_{scnd}. To this end, we appeal to a recently proposed query performance prediction (QPP) framework [16]. Specifically, the prediction task amounts to estimating the relevance likelihood of D_{scnd}, $p(D_{scnd}|q, r)$, where r is a relevance event[1].

We can use *reference document lists* to derive an estimate for $p(D_{scnd}|q, r)$ [16]:

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331369

[1] This is post-retrieval prediction which relies on analyzing the retrieved list. The relevance of a retrieved list is a notion that generalizes that for a single document [16]. At the operational level, a binary relevance judgment for a list can be obtained by thresholding any list-based evaluation measure.

$$\hat{p}(D_{scnd}|q,r) \stackrel{def}{=} \sum_{L} \hat{p}(D_{scnd}|q,L,r)\hat{p}(L|q,r); \quad (1)$$

L is a document list retrieved for q; herein, $\hat{p}$ is an estimate for p. The underlying idea is that strong association (e.g., similarity) of D_{scnd} with reference lists L (i.e., high $\hat{p}(D_{scnd}|q,L,r)$) which are presumably effective (i.e., high $\hat{p}(L|q,r)$) attests to retrieval effectiveness.

It was shown that numerous existing post-retrieval prediction methods can be instantiated from Equation 1 where a single reference list is used. Similarly, here we use D_{init} as a reference list:

$$\hat{p}(D_{scnd}|q,r) \approx \hat{p}(D_{scnd}|q,D_{init},r)\hat{p}(D_{init}|q,r). \quad (2)$$

That is, by the virtue of the way D_{scnd} is created — i.e., using information induced from D_{init} — we assume that D_{init} is the most informative reference list with respect to D_{scnd}'s effectiveness. A case in point, an expanded query constructed from a poor initial list (i.e., which mostly contains non-relevant documents) is not likely to result in effective retrieval.

3.1 Instantiating Predictors

Equation 2 can be instantiated in various ways, based on the choice of estimates, to yield a specific prediction method. To begin with, any post-retrieval predictor, $\mathcal{P}$, can be used to derive $\hat{p}(D_{init}|q,r)$ [16].

For $\hat{p}(D_{scnd}|q,D_{init},r)$ in Equation 2, we use logarithmic interpolation:

$$\hat{p}(D_{scnd}|q,D_{init},r) \stackrel{def}{=} \hat{p}^{[\mathcal{P}]}(D_{scnd}|q,r)^{\alpha}\hat{p}(D_{scnd}|D_{init},r)^{(1-\alpha)}; \quad (3)$$

α ($\in [0,1]$) is a free parameter. The estimate $\hat{p}^{[\mathcal{P}]}(D_{scnd}|q,r)$ corresponds to the predicted effectiveness of D_{scnd}, where the prediction, performed using the post-retrieval predictor $\mathcal{P}$, ignores the knowledge that D_{scnd} was produced using information induced from D_{init}.

The estimate $\hat{p}(D_{scnd}|D_{init},r)$ from Equation 3, of the association between D_{scnd} and D_{init}, is usually devised based on some symmetric inter-list similarity measure $sim(D_{scnd},D_{init})$ [16]. However, as Roitman [11] has recently suggested, a more effective estimate can be derived by exploiting the *asymmetric co-relevance* relationship between the two lists (cf., [10]); that is, $\hat{p}(D_{scnd}|D_{init},r)$ is the likelihood of D_{scnd} given that a relevance event has happened and D_{init} was observed:

$$\hat{p}(D_{scnd}|D_{init},r) \stackrel{def}{=} \hat{p}(D_{scnd}|D_{init}) \sum_{d\in D_{scnd}} \hat{p}(d|D_{scnd}) \frac{\hat{p}(d,r|D_{init})}{\hat{p}(d|D_{init})\hat{p}(r|D_{init})}; \quad (4)$$

d is a document. Following Roitman [11], we use $\hat{p}(D_{scnd}|D_{init}) \stackrel{def}{=} sim(D_{scnd},D_{init})$. Similarly to some prior work [7, 11], for $\hat{p}(r|D_{init})$ we use the entropy of the centroid (i.e., the arithmetic mean) of the language models of documents in D_{init}. We further assume that $\hat{p}(d|D_{scnd})$ and $\hat{p}(d|D_{init})$ are uniformly distributed over D_{scnd} and D_{init}, respectively. Finally, to derive $\hat{p}(d,r|D_{init})$, we follow Roitman [11] and use the corpus-based regularized cross entropy (CE) between a relevance model, $R[_{D_{init}}]$, induced from D_{init}, and a language model, $p_d(\cdot)$, induced from d:

$$\hat{p}(d,r|D_{init}) \stackrel{def}{=} CE(R[_{D_{init}}]||p_d(\cdot)) - CE(R[_{D_{init}}]||p_{\mathcal{D}}(\cdot)); \quad (5)$$

$\hat{p}_{\mathcal{D}}(\cdot)$ is a language model induced from the corpus. Further details about language model induction are provided in Section 4.1.

4 EVALUATION

4.1 Experimental setup

4.1.1 Datasets. We used for evaluation the following TREC corpora and topics: WT10g (451-550), GOV2 (701-850), ROBUST (301-450, 601-700) and AP (51-150). These datasets are commonly used in work on QPP [2]. Titles of TREC topics were used as queries. We used the Apache Lucene[2] open source search library for indexing and retrieval. Documents and queries were processed using Lucene's English text analysis (i.e., tokenization, lowercasing, Porter stemming and stopping). For the retrieval method — both the initial retrieval and the second one using the induced query model — we use the language-model-based cross-entropy scoring (Lucene's implementation) with Dirichlet smoothed document language models where the smoothing parameter was set to 1000.

4.1.2 Pseudo-feedback based retrieval. Let $c_x(w)$ denote the occurrence count of a term w in a text (or text collection) x; let $|x| \stackrel{def}{=} \sum_{w\in x} c_x(w)$ denote x's length. Let $p_x^{[\mu]}(w) \stackrel{def}{=} \frac{c_x(w)+\mu p_{\mathcal{D}}(w)}{|x|+\mu}$ denote x's Dirichlet-smoothed language model, where $p_{\mathcal{D}}(w) \stackrel{def}{=} \frac{c_{\mathcal{D}}(w)}{|\mathcal{D}|}$. For a query q and a set of pseudo-relevant documents $F \subseteq D_{init}$, $p_F(\cdot)$ denotes a pseudo-feedback-based query model.

We use three state-of-the-art pseudo-feedback-based (PRF) query-model induction methods. All three incorporate query anchoring as described below. The first is the *Relevance Model* [8] (**RM**):

$$p_F(w) \stackrel{def}{=} \sum_{d\in F} p_d^{[0]}(w)p_q^{[\mu]}(d), \quad (6)$$

where $p_q^{[\mu]}(d) \stackrel{def}{=} \frac{p_d^{[\mu]}(q)}{\sum_{d'\in F} p_{d'}^{[\mu]}(q)}$ and $p_d^{[\mu]}(q) \stackrel{def}{=} \sum_{w\in q} c_d(w)\log p_d^{[\mu]}(w)$.

The second is the *Generative Mixed Model* [19] (**GMM**) which is estimated using the following EM algorithm iterative update rules: $t^{(n)}(w) \stackrel{def}{=} \frac{(1-\beta)p_F^{(n-1)}(w)}{(1-\beta)p_F^{(n-1)}(w)+\beta p_{\mathcal{D}}(w)}$, $p_F^{(n)}(w) \stackrel{def}{=} \frac{\sum_{d\in F} c_d(w)t^{(n)}(w)}{\sum_{w'\in V}\sum_{d\in F} c_d(w')t^{(n)}(w')}$.

The third is the *Maximum-Entropy Divergence Minimization Model* [9] (**MEDMM**): $p_F(w) \propto \exp\left(\frac{1}{\gamma}\sum_{d\in F}\hat{p}_q^{[\mu]}(d)\log p_d^{[0]}(w) - \frac{\nu}{\gamma}p_{\mathcal{D}}(w)\right)$.

We applied query anchoring [8, 9, 19] to all three models: $p_{F,\lambda}(w) \stackrel{def}{=} \lambda p_q^{MLE}(w) + (1-\lambda)p_F(w)$; $p_q^{MLE}(w)$ is the maximum likelihood estimate of w with respect to q and $\lambda \in [0,1]$.

We used the n most highly ranked documents in the initial retrieval for query-model induction (i.e., inducing $p_F(\cdot)$). Then, a second query q_f was formed using the l terms w assigned the highest $p_F(w)$. We resubmitted q_f to Lucene[3] to obtain D_{scnd}.

4.1.3 Baseline predictors. As a first line of baselines, we use **Clarity** [4], **WIG** [20] and **NQC** [17], which are commonly used post-retrieval QPP methods [2]. These baselines are also used for $\mathcal{P}$ in Eq. 3. **Clarity** [4] is the divergence between a language model induced from a retrieved list and that induced from the corpus.

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

[3] Expressed in Lucene's query parser syntax as: $w_1\hat{}p_F(w_1)\ w_2\hat{}p_F(w_2)\ldots w_l\hat{}p_F(w_l)$.

Table 1: Prediction quality. Boldface: best results per basic QPP method and query-model induction method. Underlined: best results per query-model induction method. '$*$' marks a statistically significant difference between PFR-QPP and either the second best predictor (in case PFR-QPP is the best) or the best predictor (in case PFR-QPP is not).

	WT10g			GOV2			ROBUST			AP		
Method	RM	GMM	MEDMM	RM	GMM	MEDMM	RM	GMM	MEDMM	RM	GMM	MEDMM
ListSim	.442	.532	.337	.490	.432	.410	.543	.528	.436	.537	.343	.407
NQC($D_{scnd}|q_f$)	.293	.228	.182	**.599**	.545	.353	.653	.637	.622	.655	.617	.454
NQC($D_{scnd}|q$)	.071	.051	.092	.437	.418	.283	.475	.492	.620	.574	.479	.530
NQC($D_{init}|q$)	.483	.397	**.424**	.486	.414	.414	.635	.605	.602	.550	.536	.502
RefList(NQC)	**.535**	.531	.415	.517	.486	.457	.654	.631	.621	.607	.530	.572
PFR-QPP(NQC)	.513*	**.557***	.410	.596	**.549**	**.550***	**.671***	**.661***	**.642***	**.670***	**.640***	**.650***
Clarity($D_{scnd}|q_f$)	.292	.325	.316	.230	.157	.130	.450	.393	.409	.313	.408	.339
Clarity($D_{scnd}|q$)	.327	.227	.368	.278	.200	.084	.412	.350	.349	.236	.350	.270
Clarity($D_{init}|q$)	.363	.350	.314	.282	.261	.264	.452	.441	.401	.320	.456	.308
RefList(Clarity)	**.481**	**.567**	.388	.480	.469	.414	.582	.575	.535	.589	.519	.511
PFR-QPP(Clarity)	.408*	.557*	**.398***	**.615***	**.497***	**.490***	**.589***	**.607***	**.566***	**.652***	**.585***	**.651***
WIG($D_{scnd}|q_f$)	.270	.307	**.388**	.263	.301	.448	.424	.361	.381	.159	.281	.285
WIG($D_{scnd}|q$)	.253	.105	.153	.583	.424	.276	.651	.455	.430	.414	.281	.226
WIG($D_{init}|q$)	.237	.221	.224	.562	.498	.498	.649	.618	.578	.554	.614	.505
RefList(WIG)	.338	.384	.311	.581	.562	.480	.660	.638	.637	.639	.580	.608
PFR-QPP(WIG)	**.370***	**.466***	.353*	**.630***	**.603***	**.575***	**.665***	**.682***	**.648***	**.650***	**.634***	**.643***
WEG($D_{scnd}|q_f$)	.231	.205	.331	.585	.548	.432	.661	.656	**.693**	.627	**.562**	.575
WEG($D_{scnd}|q$)	.141	.134	.239	.513	.504	.390	.566	.571	.674	.560	.491	.575
WEG($D_{init}|q$)	.353	.311	.313	.532	.470	.409	.635	.619	.616	.526	.474	.518
RefList(WEG)	.443	.483	.371	.527	.481	.427	.654	.633	.632	.580	.467	.555
PFR-QPP(WEG)	**.456***	**.575***	**.436***	**.660***	**.562***	**.481***	**.688***	**.664***	.688*	**.675***	.552*	**.664***

WIG [20] and **NQC** [17] are the corpus-regularized[4] mean and standard deviation of retrieval scores in the list, respectively. We further compare with the *Weighted Expansion Gain* (**WEG**) [6] method – a WIG alternative which regularizes with the mean score of documents at low ranks of the retrieved list instead of the corpus.

We use three variants of each of the four predictors described above. The first two directly predict the effectiveness of the final retrieved list D_{scnd} using either (i) the original query q (denoted $\mathcal{P}(D_{scnd}|q)$), or (ii) the query q_f (denoted $\mathcal{P}(D_{scnd}|q_f)$) which was induced from D_{init} as described above (cf., [15, 20]). The third variant (denoted $\mathcal{P}(D_{init}|q)$) is based on predicting the performance of D_{scnd} by applying the predictor to D_{init} as was the case in [15].

To evaluate the impact of our inter-list association measure in Eq. 4, we use two additional baselines. The first, denoted **ListSim** [16], uses $sim(D_{scnd}, D_{init})$ to predict the performance of D_{scnd}. The second, denoted **RefList**($\mathcal{P}$) [7, 16], treats D_{init} as a pseudo-effective list of D_{scnd} and estimates D_{scnd}'s performance by:

$$\hat{p}_{RefList}(r|D_{scnd}, q) \stackrel{def}{=} sim(D_{scnd}, D_{init})\hat{p}^{[\mathcal{P}]}(D_{init}|q, r),$$

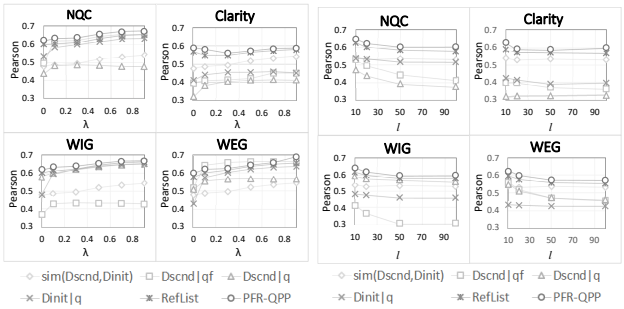
where $\mathcal{P}$ is one of the four basic QPP methods described above. There are two important differences between our proposed method and **RefList**. First, we use the query q in the list association measure in Eq. 3. Second, we use an asymmetric co-relevance measure between the two lists in Eq. 4 compared to the symmetric one used in **RefList**.

4.1.4 Setup. Hereinafter, we refer to our proposed QPP method from Eq. 2 as **PFR-QPP**: Pseudo-Feedback based Retrieval QPP. **PFR-QPP**($\mathcal{P}$) is a specific predictor instantiated using the base predictor $\mathcal{P}$. We predict for each query the effectiveness of the 1000 documents (i.e., $k = 1000$) most highly ranked in the final result list D_{scnd}. Prediction quality is measured using the Pearson correlation between the ground truth AP@1000 (according to TREC's relevance judgments) and the values assigned to the queries by a prediction method.

Most prediction methods described above incorporate free parameters. Following the common practice [2], we set $m \leq k$ – the number of documents in a given list (i.e., either D_{scnd} or D_{init}) used for calculating a given predictor's value; with $m \in \{5, 10, 20, 50, 100, 150, 200, 500, 1000\}$. We applied term clipping with l terms ($l \in \{10, 20, 50, 100\}$) to the relevance model used in **Clarity** and **PFR-QPP**. Following [16], we realized the **ListSim**, **RefList** and **PFR-QPP** baselines using *Rank-Biased Overlap* (*RBO*(p)) [18] as our list-similarity measure $sim(\cdot)$ (with $p = 0.95$, further following [16]). For our **PFR-QPP** method, we set $\alpha \in \{0, 0.1, 0.2, \ldots, 0.9, 1.0\}$. Query models are induced from the $n = 50$ top documents. For the **GMM** and **MEDMM** models, we set their free-parameters to previously recommended values, i.e., **GMM** ($\beta = 0.5$) [19] and **MEDMM** ($\gamma = 1.2, \nu = 0.1$) [9]. Unless stated otherwise, the query anchoring and clip-size parameters in all models were fixed to $\lambda = 0.9$ and $l = 20$, respectively. The prediction quality for other (λ, l) settings is studied in Section 4.2.

Following [12, 17], we trained and tested all methods using a 2-fold cross validation approach. Specifically, in each dataset, we generated 30 random splits of the query set; each split had two folds. We used the first fold as the (query) train set. We kept the second fold for testing. We recorded the average prediction quality

[4] To this end, the corpus is treated as one large document.



(a) Query anchoring (λ). **(b) Number of terms (l).**

Figure 1: Sensitivity to free-parameter values of the relevance model used for query-model induction.

over the 30 splits. Finally, we measured statistically significant differences of prediction quality using a two-tailed paired t-test with $p < 0.05$ computed over all 30 splits.

4.2 Results

Table 1 reports the prediction quality of our method and the baselines. We can see that in the vast majority of cases, our **PFR-QPP** approach statistically significantly outperforms the baselines.

Applying basic QPP methods. We first compare the three variants of the four basic QPP methods. We observe that, in most cases, utilizing the PRF-induced query q_f for predicting the performance of the final list D_{scnd} ($\mathcal{P}(D_{scnd}|q_f)$), yields better prediction quality than using the original query q ($\mathcal{P}(D_{scnd}|q)$). In addition, predicting the performance of D_{scnd} by applying the base predictor to the initially retrieved list D_{init} ($\mathcal{P}(D_{init}|q)$) yields high prediction quality — sometimes even higher than applying the predictor to D_{scnd}. These findings provide further support the motivation behind **PFR-QPP**: integrating prediction for the initially retrieved list and the final retrieved list and accounting for their asymmetric co-relevance relation.

PFR-QPP vs. reference-list based alternatives. First, in line with previous work [7, 15, 16], the high prediction quality of **ListSim** and **RefList** in our setting shows that the similarity between the two lists is an effective performance indicator. Moreover, combining prediction for the performance of the initial list with its similarity with the final list (i.e., **RefList**) yields prediction quality that transcends in most cases that of using only the similarity (i.e., **ListSim**). Finally, our **PFR-QPP** method which uses prediction for both the initial and final lists, and accounts for their asymmetric co-relevance relationship, outperforms both **ListSim** and **RefList** in most cases, and often to a statistically significant degree.

Sensitivity to query-model induction tuning. Using the ROBUST dataset and the relevance model (**RM**), Figure 1 reports the effect on prediction quality of varying the value of the query anchoring parameter (λ; while fixing $l = 20$) and the number of terms used after clipping (l; while fixing $\lambda = 0$) in the query model, and hence, in q_f. As can be seen, decreasing λ or increasing l decreases the prediction quality of all methods. With reduced query anchoring or when using more terms, the induced queries (q_f) tend to become more "verbose", with less emphasis on the original query q. Indeed, a recent study showed that existing QPP methods are less robust for long queries [12]. Finally, we see that for any value of λ and l, **PFR-QPP** outperforms the baselines.

5 CONCLUSIONS

We addressed the QPP task for pseudo-feedback-based retrieval, where the final retrieved list depends on an initially retrieved list – e.g., via a query model induced from the latter and used to produce the former. Our approach accounts for the predicted effectiveness of each of the two lists as well as to their asymmetric co-relevance relation. Empirical evaluation showed that our approach significantly outperforms a variety of strong baselines.

ACKNOWLEDGEMENT

We thank the reviewers for their comments. This work was supported in part by the Israel Science Foundation (grant no. 1136/17)

REFERENCES

[1] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. Query difficulty, robustness, and selective application of query expansion. In *Proceedings of ECIR*, pages 127–137, 2004.
[2] David Carmel and Oren Kurland. Query performance prediction for ir. In *Proceedings of SIGIR*, pages 1196–1197, New York, NY, USA, 2012. ACM.
[3] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, January 2012.
[4] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *Proceedings of SIGIR*, pages 299–306, New York, NY, USA, 2002. ACM.
[5] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. A framework for selective query expansion. In *Proceedings of CIKM*, pages 236–237, New York, NY, USA, 2004. ACM.
[6] Ahmad Khwileh, Andy Way, and Gareth J. F. Jones. Improving the reliability of query expansion for user-generated speech retrieval using query performance prediction. In *CLEF*, 2017.
[7] 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*, pages 823–832, New York, NY, USA, 2012. ACM.
[8] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of SIGIR*.
[9] Yuanhua Lv and ChengXiang Zhai. Revisiting the divergence minimization feedback model. In *CIKM '14*.
[10] Fiana Raiber, Oren Kurland, Filip Radlinski, and Milad Shokouhi. Learning asymmetric co-relevance. In *Proceedings of ICTIR*, pages 281–290, 2015.
[11] Haggai Roitman. Enhanced performance prediction of fusion-based retrieval. In *Proceedings of ICTIR*, pages 195–198, New York, NY, USA, 2018. ACM.
[12] Haggai Roitman. An extended query performance prediction framework utilizing passage-level information. In *Proceedings of ICTIR*, pages 35–42, New York, NY, USA, 2018. ACM.
[13] Haggai Roitman, Ella Rabinovich, and Oren Sar Shalom. As stable as you are: Re-ranking search results using query-drift analysis. In *Proceedings of HT*, pages 33–37, New York, NY, USA, 2018. ACM.
[14] Harrisen Scells, Leif Azzopardi, Guido Zuccon, and Bevan Koopman. Query variation performance prediction for systematic reviews. In *Proceedings of SIGIR*, pages 1089–1092, New York, NY, USA, 2018. ACM.
[15] Anna Shtok, Oren Kurland, and David Carmel. Using statistical decision theory and relevance models for query-performance prediction. In *Proceedings of SIGIR*, pages 259–266, New York, NY, USA, 2010. ACM.
[16] 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.
[17] 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.
[18] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4):20:1–20:38, November 2010.
[19] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*.
[20] Yun Zhou and W. Bruce Croft. Query performance prediction in web search environments. In *Proceedings of SIGIR*, pages 543–550, New York, NY, USA, 2007. ACM.