

Query-specific Variable Depth Pooling via Query Performance Prediction

Debasis Ganguly
University of Glasgow
Glasgow, United Kingdom
debasis.ganguly@glasgow.ac.uk

Emine Yilmaz
University College London
London, United Kingdom
emine.yilmaz@ucl.ac.uk

ABSTRACT

Due to the massive size of test collections, a standard practice in IR evaluation is to construct a ‘pool’ of candidate relevant documents comprised of the top- k documents retrieved by a wide range of different retrieval systems - a process called depth- k pooling. A standard practice is to set the depth (k) to a constant value for each query constituting the benchmark set. However, in this paper we argue that the annotation effort can be substantially reduced if the depth of the pool is made a variable quantity for each query, the rationale being that the number of documents relevant to the information need can widely vary across queries. Our hypothesis is that a lower depth for queries with a small number of relevant documents, and a higher depth for those with a larger number of relevant documents can potentially reduce the annotation effort without a significant change in IR effectiveness evaluation. We make use of standard query performance prediction (QPP) techniques to estimate the number of potentially relevant documents for each query, which is then used to determine the depth of the pool. Our experiments conducted on standard test collections demonstrate that this proposed method of employing query-specific variable depths is able to adequately reflect the relative effectiveness of IR systems with a substantially smaller annotation effort.

CCS CONCEPTS

• Information systems → Query intent; Information retrieval query processing.

KEYWORDS

IR Model Evaluation, Depth Pooling, Query Performance Prediction

ACM Reference Format:

Debasis Ganguly and Emine Yilmaz. 2023. Query-specific Variable Depth Pooling via Query Performance Prediction. In *Proceedings of ACM Conference (Conference’17)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

The most widely used approach used in evaluating quality of retrieval systems is based on constructing test collections via the

Cranfield paradigm [6], which assumes that relevance judgments for each query are complete. However, due to the cost of obtaining relevance judgments, it is often impractical to obtain relevance judgments for all documents in a collection.

A commonly used approach to reduce the need for extensive judging effort in test collection construction is depth- k pooling, which is based on constructing a pool of documents that consists of the top- k documents retrieved by various systems and then obtaining relevance judgments for these documents, assuming that the rest of the documents are non-relevant. Most existing test collections, such as the ones constructed by TREC are constructed using depth-100 pools, i.e., using a value of $k = 100$ [24]. However, depth-100 pools still tend to be quite large and hence significant research has been devoted to reducing the number of judgments needed in constructing test collections [5, 25]. In order to reduce the pool depth, some recent test collections are, instead, constructed using much shallower depths, such as depth-10 pools used by the recent Deep Learning Track collections [9].

Most previous work assumes that a constant depth (k) should be used across all the queries in the test collection. However, some queries may contain more relevant documents than the others, and using the same depth across all queries could lead to wasting a significant proportion of annotation budget on those queries where fewer judgments could have been sufficient [25, 26].

Some previous work based on active learning proposed approaches employed deeper depths for systems that are more likely to retrieve a higher number of relevant documents [7, 19, 22] - a process which often leads to different rank cutoffs for different systems. Optimising the resultlist presentation to search system users motivated a similar thread of work that involves chopping off a ranked list of documents at different cut-off points based on the statistics of their score distributions [2, 18]. More recently, supervised learning via neural networks has been applied to address this problem [4].

While previous work has investigated cutting off ranked lists at variable depths to reduce the information finding effort of search engine users [2, 4, 7, 19, 22], we, in contrast, use the concept of variable depths for reducing the assessment effort. The rationale behind the idea is that some queries have a smaller number of documents in the collection that are relevant to their corresponding information needs, whereas for some other queries this number may be substantially larger. Our hypothesis is that a lower depth for the former class of queries and a higher depth for the latter can potentially reduce the annotation effort without a significant change in the relative evaluation of different retrieval systems.

In particular, to estimate the number of potentially relevant documents for each query we make use of a standard query performance prediction (QPP) based approach (specifically, NQC [23] in this paper). This number of relevant documents estimated for a query

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference’17, July 2017, Washington, DC, USA

© 2023 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

in the top-documents retrieved by an IR system is then used to determine the number of documents that contribute to the pool for that particular query and IR system combination.

2 PROPOSED METHODOLOGY

2.1 A Review of QPP

QPP approaches can broadly be categorized into the pre-retrieval and post-retrieval types. A pre-retrieval estimator uses aggregated collection-level statistics (e.g., maximum or average of the inverse document frequencies of the query terms) as an estimated performance measure of a query [16, 17, 28]. A post-retrieval estimator, on the other hand, makes use of the information from the set of top-retrieved documents to estimate the quality of the retrieved list. In general, the QPP score for a post-retrieval estimator ϕ is a function of the query and the set of top-retrieved documents, i.e.,

$$\phi : Q \times M^{(k)}(Q) \mapsto \mathbb{R}, \quad (1)$$

where $M^{(k)}(Q)$ denotes the set of top- k documents retrieved for query Q with a model M . From hereon, $M^{(k)}(Q)$ is abbreviated as $M^{(k)}$, the query being understood from the context.

Various evidences extracted from the top-retrieved documents have been shown to be useful for different post-retrieval QPP estimation methods, such as the KL divergence between the language model of the top-retrieved documents and the collection model as in Clarity [11], the aggregated values of the information gains of each top-retrieved document with respect to the collection as in WIG (Weighted Information Gain) [29], the skewness of the retrieval status values (RSVs) measured with variance as in NQC (Normalized Query Commitment) [23], ideas based on the clustering hypothesis for a pairwise document similarity matrix [15], topology of the embedded word vectors [21] and even supervised approaches using neural networks [12, 14].

In our work, as an unsupervised QPP approach we employ NQC (Normalized Query Commitment) [23], which is a simple yet effective post-retrieval QPP method [13, 27] (we leave the investigation with other QPP approaches as future work). NQC predicts the retrieval effectiveness of a query using the variance of the document scores, the rationale being that a query with a well-defined information need is likely to lead to a more non-uniform (heavy-tailed) distribution of the RSVs. Formally speaking, the generic ϕ function of Equation 1 takes the form

$$\phi_{\text{NQC}}(Q, M^{(k)}) \stackrel{\text{def}}{=} \frac{\sqrt{\frac{1}{k} \sum_{i=1}^k (P(D_i|Q) - \bar{P}(D|Q))^2}}{P(Q|C)}, \quad (2)$$

where $P(D_i|Q)$ denotes the RSV of the document D_i for Q , $\bar{P}(D|Q)$ denotes the mean of the RSVs, and $P(Q|C)$ denotes the similarity of Q to the collection, which is computed by aggregating collection statistics (e.g., idf) over the query terms.

Although for our experiments we specifically use the NQC method, our proposed method of variable depth pooling strategy (to be discussed in the next section) is a general one allowing application of any other QPP model as a concrete realisation of ϕ (Equation 2).

2.2 Depth Estimation using QPP

Given a set of n queries $Q = \{Q_1, \dots, Q_n\}$, a standard depth- k pooling process first involves employing a number of different retrieval systems (models), say $M = \{M_1, \dots, M_p\}$ to construct a pool of the top- k documents retrieved with each M_i ($i = 1, \dots, p$). Formally speaking, this pool of depth k , $\mathcal{P}_k(Q)$ for query $Q \in Q$ is constructed as

$$\mathcal{P}_k(Q) = \cup_{i=1}^p M_i^{(k)}, \quad (3)$$

where $M_i^{(k)}$ denotes the top- k documents retrieved with model M_i .

The key idea now is to make this depth k a function of Q itself rather than it being a constant across all queries. We propose to make this integer-valued depth of a query a function of the generic form of the real-valued QPP estimate ϕ (which also depends on $M_i^{(k)}$) as shown in Equation 1, and denote this integer-valued depth function as $\zeta(Q, M_i) \in \mathbb{Z}^+$.

One important point to note is that we make the variable depth a function of the query and of the retrieval model. As a next step after computing the depth $\zeta(Q, M_i)$ for a 'query and system' combination (Q, M_i) , we use this depth to determine the number of top documents from $M_i^{(\zeta(Q, M_i))}$ to add to the pool. In other words, we obtain a more generic version of Equation 3 as

$$\mathcal{P}_{\zeta(Q)} = \cup_{i=1}^p M_i^{(\zeta(Q, M_i))}. \quad (4)$$

We now explore two different ways by which this variable depth of a query-model combination may depend on the QPP estimator. Each of these two choices of $\zeta(Q, M)$ has its own set of supporting arguments; more details follow.

2.2.1 Inverse Linear Dependence. The first choice for $\zeta(Q, M)$ is a linearly inverse proportional function, the intuition for which is that the higher the value of the estimate - the higher is the likelihood of the ranked list (as retrieved by M) to contain a higher proportion of relevant documents towards the top ranks. This, in turn, means that a smaller depth for such a query is likely to be adequate to include an adequate set of potentially relevant documents in the constructed pool for a robust evaluation of IR systems.

On the other hand, a relatively low value of the QPP estimate for a query potentially indicates that more documents should perhaps be included in the pool by employing a higher depth value for that query. Formally speaking, using the generic notation of the QPP function of Equation 1, the depth of a query Q is then

$$\zeta(Q, M_i) = d_{\min} + \lfloor (1 - \phi(Q, M_i^{(d_{\max})})) (d_{\max} - d_{\min}) \rfloor, \quad (5)$$

where the parameters $d_{\min}$ denotes the minimum depth, and $d_{\max}$ denotes the maximum depth ($d_{\min}, d_{\max} \in \mathbb{Z}^+$, i.e., they both are positive integers).

Normalized values of the QPP $\phi(Q, M_i^{(d_{\max})})$ estimates ensure that the depth of a query is an integer between the integer bounds $d_{\min}$ and $d_{\max}$. Note that, in particular, for computing the QPP estimates themselves we use $k = d_{\max}$ (the maximum depth), and we apply max-normalization for the QPP estimates.

2.2.2 Linear Dependence. The argument for this choice of $\zeta(Q, M)$ is that a higher value of $\phi(Q, M^{(k)})$ is likely to indicate that a higher number of potentially relevant documents for Q exists in the collection. This, in turn, means that one may consider probing

at higher depths to collect those candidates for assessment for a more comprehensive evaluation of IR systems. Similarly, a smaller estimate for $\phi(Q, M^{(d_{max})})$ means that it is not worthwhile to use a high depth for Q because the candidates collected from lower down the ranked lists of such IR systems may end up in the ground-truth set indicating wasted manual effort. Formally,

$$\zeta(Q, M_i) = d_{min} + \lfloor \phi(Q, M_i^{(d_{max})})(d_{max} - d_{min}) \rfloor, \quad (6)$$

where the only difference of Equation 6 with that of 5 is that in the former an increase in the $\phi(Q, M_i^{(d_{max})})$ increases the depth to a higher integer value within the bounds $[d_{min}, d_{max}]$ instead of decreasing it as is the case for the latter.

3 EVALUATION

3.1 Experiment Details

3.1.1 Research Questions. We conduct experiments to investigate the following two research questions.

- **RQ-1:** Is an NQC-based variable depth pooling strategy beneficial to reduce annotation effort without causing significant changes in the relative system ranks?
- **RQ-2:** Which depth selection function (linear or inverse linear - Equations 5 or 6) turns out to be the more effective of the two?

3.1.2 Datasets. Our retrieval experiments are conducted on two standard datasets used for the ad-hoc IR task, namely the TREC Robust [24] and the TREC DL datasets [20]. While the former is comprised of news articles, the latter is a collection of passages accumulated with Bing queries. The set of relevant documents comprising the ground-truth of the TREC Robust dataset was constructed via depth-100 pooling [24]. On the other hand, in TREC DL a combination of depth-10 pooling and an active learning based strategy [1] was used to compile the ground-truth [8, 10]. We leave out TREC 6 topic sets from our experiments for consistency with the remaining topic sets, the ground-truths of which do not include the congressional records (CR). Table 1 summarises the datasets used for our experiments.

3.1.3 Setup. For each topic set used in our experiments, we make use of the officially submitted runs as downloaded from the TREC archive¹. We conduct our experiments on each topic set separately so as to compute the effect of the relative changes in the systems (officially submitted runs) in each.

In each experiment, the value of the minimum depth (d_{min} of Equations 5 and 6) was set to 10% of the true depth used to construct the pool of the respective datasets, i.e., $0.1 \times 100 = 10$ for TREC Robust, and $0.1 \times 10 = 1$ for TREC DL. Similarly, the value of the maximum depth (d_{max}) was set to half the value of the true depths, i.e., $100/2 = 50$ and $10/2 = 5$ for TREC Robust and TREC DL datasets, respectively. As the QPP estimate $\phi(Q, M^{(k)})$, we use the standard unsupervised QPP approach - NQC.

3.1.4 Pooling Methods Investigated. As baselines, we employ the standard procedure of constant-depth pooling (CDP) (Equation 3). Since our proposed methodology uses depths that varies across queries, for a fair comparison we compare our proposed approach with several CDP baselines, as enumerated below.

Table 1: Summary of the datasets used in our experiments. The columns ‘ $|\bar{Q}|$ ’ and ‘ $\#Rel$ ’ denote the average number of query terms and average number of relevant documents, respectively. The column p denotes the number of official runs submitted, all of which is used to construct the pools (Equation 3) for each topic set.

Collection	#Docs	Topics	#Topics	$ \bar{Q} $	$\#Rel$	p
Robust (disks 4,5 - CR)	528,155	TREC 7	50	2.42	93.48	103
		TREC 8	50	2.38	94.56	129
MS MARCO Passage	8,841,823	DL'19	43	5.40	58.16	37
		DL'20	54	6.04	30.85	59

- **CDP-Max** involves setting $k = d_{max}$ in Equation 3, where d_{max} is the upper bound of the depth used in the VDP approach (d_{max} in Equations 5 and 6). This method thus represents an apex-line (AL) scenario with a larger pool size, thus implying a larger effort for relevance assessments.
- **CDP-Min** is a baseline which sets $k = d_{min}$ in Equation 3 thus implying that this represents the lower end of the spectrum with a much smaller pool size.
- **CDP-Avg** is a baseline with the depth of the pool being set to the closest integer of the mid-point of the interval $[d_{min}, d_{max}]$, i.e., setting $k = \lfloor (d_{max} - d_{min})/2 \rfloor$ in Equation 3. This baseline yields a pool that is expected to be of a size similar to those obtained by the VDP-based methods.

As variants of our proposed methodology of variable-depth pooling (VDP), we explore the following.

- **VDP-IL:** this denotes variable-depth pooling by means of an inverse linear dependence (Equation 5).
- **VDP-L:** this denotes variable-depth pooling with a linear dependence (Equation 6)².

3.1.5 Evaluation Metrics. As per the standard practice of a simulated pooling setup [3], the pool of documents obtained with each method is a subset of the existing relevance assessments. This allows provision to compute the quality of a pool by comparing the correlation of the relative system ranks measured via the ground-truth induced on the subset as against the entire existing pool of the respective datasets.

A smaller pool is considered to be of good quality if the relative system ranks measured via an IR metric (e.g., AP) on this smaller set of ground-truth does not change substantially the metrics as measured on the larger pool. In particular, as correlation measures between IR models we employ Pearson’s r and Kendall’s τ . We employed mean average precision (MAP) to induce an order on the different officially submitted runs (systems). As per the standard practice, AP on the TREC DL dataset considered graded judgments higher than or equal to 2 as relevant [8, 10].

In addition to the relative rank stability of systems, we also report the recall or **coverage**, measured as the fraction of relevant documents found in a depth restricted pool averaged across all the

¹<https://trec.nist.gov/results/>

²The implementation of all the methods investigated is available at <https://github.com/gdebasis/vardepthpooling>

Table 2: A comparison of QPP-based VDP with CDP approaches (including the oracle case, denoted as ‘AL’ or apex-line, shown in green) on the TREC Robust topic sets. The best results along each column of the non-oracle results are bold-faced. A higher value of all the metrics except $|\overline{\mathcal{P}}|$ (average pool size) indicates a more effective pooling strategy.

Set	Type	Pool	Avg. Depth	Evaluation Metrics				
				P-r	K- τ	C	$ \overline{\mathcal{P}} $	PNC
TREC 7	BL	CDP-Min	10	0.9897	0.9261	0.3988	187.66	0.0762
		CDP-Avg	30	0.9988	0.9714	0.6711	484.00	0.1086
	Ours	VDP-L	16.36	0.9985	0.9760	0.7021	579.52	0.1104
		VDP-IL	42.72	0.9986	0.9718	0.6467	414.50	0.1073
	AL	CDP-Max	50	0.9996	0.9886	0.8223	759.00	0.1240
TREC 8	BL	CDP-Min	10	0.9922	0.9215	0.4052	239.54	0.0740
		CDP-Avg	30	0.9987	0.9683	0.6596	620.96	0.1026
	Ours	VDP-L	22.90	0.9978	0.9680	0.6827	721.00	0.1037
		VDP-IL	36.14	0.9991	0.9714	0.6600	540.00	0.1049
	AL	CDP-Max	50	0.9997	0.9864	0.8201	959.62	0.1194

queries of a benchmark topic set. Formally,

$$C = \frac{1}{|R_{\max}(\mathcal{Q})|} \sum_{Q \in \mathcal{Q}} \sum_{D \in \mathcal{P}_{\zeta(Q)}} \mathbb{I}(\text{Rel}(D, Q) = 1), \quad (7)$$

where $\mathcal{P}_{\zeta(Q)}$ is defined in Equation 4, $\text{Rel}(D, Q) = 1$ if a document D is judged as relevant to Q , $\mathcal{Q}$ is a set of benchmark queries, $\mathbb{I}(\cdot)$ denotes the indicator function, and $R_{\max}(\mathcal{Q})$ represents the total number of relevant documents known for a collection, e.g., the ones obtained by employing $\zeta(Q) = 100 \forall Q \in \mathcal{Q}$, i.e., the true depth used to compile the ground-truth of the TREC Robust topic sets.

We also measure the **average pool size** $|\overline{\mathcal{P}}|$, as the number of unique documents occurring in a depth restricted pool - again averaged over the queries. Note that this measure is related to the assessment effort. Formally,

$$|\overline{\mathcal{P}}| = \frac{1}{Q} \sum_{Q \in \mathcal{Q}} \left| \bigcup_{i=1}^p M_i^{\zeta(Q)} \right|, \quad (8)$$

Since a high coverage (Equation 7) and a low average pool size (Equation 8) indicate an effective pooling strategy, for the sake of convenient comparisons we combine these two measures into a single metric. Since the average pool size per query is much larger than the recall values (bounded in $[0, 1]$), we compute the ratio after taking a log of the average pool size, akin to the tf-idf combination where the document frequencies being much larger than the tf's are used with a log transformation. Formally, we define Pool-size Normalized Coverage (**PNC**) as $C / \log |\overline{\mathcal{P}}|$, a higher value of which indicates a better coverage obtained with a low average pool size.

3.2 Results

Tables 2 and 3 present the results of our experiments on the TREC Robust and the TREC DL datasets. We observe the following trends in the results. First, our proposed variable-depth pooling (VDP) approaches outperform the constant depth pooling approaches with the depth being set to minimum and average values of the depth range, as can be seen from the better correlation values measured

Table 3: Evaluation on TREC DL topic sets, the organization of the table being identical to that of Table 2.

Set	Type	Pool	Avg. Depth	Evaluation Metrics				
				P-r	K- τ	C	$ \overline{\mathcal{P}} $	PNC
TREC DL'19	BL	CDP-Min	1	0.9022	0.6336	0.2229	8.65	0.1033
		CDP-Avg	3	0.9559	0.7147	0.4703	20.46	0.1558
	Ours	VDP-L	3.37	0.9686	0.8559	0.5398	24.76	0.1682
		VDP-IL	1.67	0.9241	0.7297	0.2814	10.83	0.1181
	AL	CDP-Max	5	0.9850	0.9399	0.6542	30.67	0.1911
TREC DL'20	BL	CDP-Min	1	0.9760	0.8656	0.2448	12.27	0.0976
		CDP-Avg	3	0.9944	0.9299	0.4878	29.48	0.1442
	Ours	VDP-L	3.87	0.9960	0.9334	0.5740	37.50	0.1584
		VDP-IL	1.16	0.9866	0.9030	0.3161	15.75	0.1146
	AL	CDP-Max	5	0.9977	0.9568	0.6671	45.24	0.1750

with r and τ . Moreover, these high correlations are observed with better coverage and PNC, which answers **RQ-1** in affirmative. CDP-Max, the apex-line setting (shown as the green rows in the tables) yields better results at the cost of higher annotation effort ($|\overline{\mathcal{P}}|$).

Second, in relation to **RQ-2**, it can be observed that there is no clear winner between the VDP-L (Equation 6) and VDP-IL (Equation 5) variants. While the linear dependence method works better than its inverse-linear counterpart for 3 topic-sets (TREC 7 and the two TREC DL sets as seen by the higher rank correlation, coverage and PNC), the inverse linear works slightly better for TREC 8.

Third, it can be seen that VDP approaches are more suitable in cases where the depth range used for VDP is comprised of smaller values, as is the case for the TREC DL ($[1, 5]$) vs. TREC Robust ($[10, 50]$); this can be seen from the better improvements in the Kendall's τ rank correlations of systems on the TREC DL topic sets compared to the TREC Robust ones (compare the relative gains of the VDP approaches with the CDP-Avg ones). Lastly, the VDP methods consistently yield better values of PNC (Pool-size normalized coverage) in comparison to the baseline CDP approaches. This means that a higher number of relevant documents can be found with reduced manual assessment effort.

Concluding Remarks. In this initial investigation of employing variable depth-pooling (VDP) strategies for constructing ground-truth relevance data, our experiments demonstrated encouraging trends. Specifically, we observe that a standard unsupervised QPP method, such as NQC, leads to satisfactory results in terms of high correlation of relative system ranks (as measured with respect to larger pool sizes). We also observed that our proposed method leads to higher coverage at the expense of smaller average pool-size. These observations indicate stable evaluation results with minimized annotation effort. There are a number of ways in which we can extend this initial exploration. First, it would be interesting to compare the relative effects of different QPP methods on VDP. It would also be interesting to see if the use of query variants, such as [13, 27], can further optimise the depth prediction of VDP.

Acknowledgement. The first author thanks Procheta Sen (University of Liverpool) for suggesting this idea during an informal conversation.

REFERENCES

- [1] Mustafa Abualsaud, Nimesh Ghelani, Haotian Zhang, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. 2018. A System for Efficient High-Recall Retrieval. In *Proc. of SIGIR'18*. 1317–1320. <https://doi.org/10.1145/3209978.3210176>
- [2] Avi Arampatzis, Jaap Kamps, and Stephen Robertson. 2009. Where to Stop Reading a Ranked List? Threshold Optimization Using Truncated Score Distributions. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Boston, MA, USA) (*SIGIR '09*). Association for Computing Machinery, New York, NY, USA, 524–531. <https://doi.org/10.1145/1571941.1572031>
- [3] Javed A. Aslam and Emine Yilmaz. 2007. Inferring document relevance from incomplete information. In *CIKM*. ACM, 633–642.
- [4] Dara Bahri, Yi Tay, Che Zheng, Donald Metzler, and Andrew Tomkins. 2020. Choppy: Cut Transformer for Ranked List Truncation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) (*SIGIR '20*). Association for Computing Machinery, New York, NY, USA, 1513–1516. <https://doi.org/10.1145/3397271.3401188>
- [5] Ben Carterette, James Allan, and Ramesh K. Sitaraman. 2006. Minimal test collections for retrieval evaluation. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, Efthimis N. Efthimiadis, Susan T. Dumais, David Hawking, and Kalervo Järvelin (Eds.). ACM, 268–275. <https://doi.org/10.1145/1148170.1148219>
- [6] Cyril Cleverdon. 1967. The Cranfield tests on index languages devices. (1967). <http://www3.interscience.wiley.com/journal/114214228/abstract>
- [7] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. 1998. Efficient Construction of Large Test Collections. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) (*SIGIR '98*). Association for Computing Machinery, New York, NY, USA, 282–289. <https://doi.org/10.1145/290941.291009>
- [8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 Deep Learning Track. In *Proc. 29th Text REtrieval Conference, TREC 2020 (NIST Special Publication, Vol. 1266)*.
- [9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. arXiv:2003.07820 [cs.IR]
- [10] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. arXiv:2003.07820 [cs.IR]
- [11] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting Query Performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR '02*). Association for Computing Machinery, New York, NY, USA, 299–306.
- [12] Suchana Datta, Debasis Ganguly, Derek Greene, and Mandar Mitra. 2022. Deep-QPP: A Pairwise Interaction-based Deep Learning Model for Supervised Query Performance Prediction. In *Proc. of WSDM'22*. 201–209.
- [13] Suchana Datta, Debasis Ganguly, Mandar Mitra, and Derek Greene. 2022. A Relative Information Gain-Based Query Performance Prediction Framework with Generated Query Variants. *ACM Trans. Inf. Syst.* 41, 2, Article 38 (dec 2022), 31 pages.
- [14] Suchana Datta, Sean MacAvaney, Debasis Ganguly, and Derek Greene. 2022. A 'Pointwise-Query, Listwise-Documents' based Query Performance Prediction Approach. In *SIGIR*. ACM, 2148–2153.
- [15] Fernando Diaz. 2007. Performance Prediction Using Spatial Autocorrelation. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR '07*). Association for Computing Machinery, New York, NY, USA, 583–590.
- [16] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. 2008. A Survey of Pre-Retrieval Query Performance Predictors. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (*CIKM '08*). Association for Computing Machinery, New York, NY, USA, 1419–1420.
- [17] Ben He and Iadh Ounis. 2004. Inferring Query Performance Using Pre-retrieval Predictors. In *String Processing and Information Retrieval*. Springer Berlin Heidelberg, Berlin, Heidelberg, 43–54.
- [18] Yen-Chieh Lien, Daniel Cohen, and W. Bruce Croft. 2019. An Assumption-Free Approach to the Dynamic Truncation of Ranked Lists. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval* (Santa Clara, CA, USA) (*ICTIR '19*). Association for Computing Machinery, New York, NY, USA, 79–82. <https://doi.org/10.1145/3341981.3344234>
- [19] David E. Losada, Javier Parapar, and Álvaro Barreiro. 2016. Feeling Lucky? Multi-Armed Bandits for Ordering Judgements in Pooling-Based Evaluation. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing* (Pisa, Italy) (*SAC '16*). Association for Computing Machinery, New York, NY, USA, 1027–1034.
- [20] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. In *Proc. of CoCo@NIPS*.
- [21] Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth J. F. Jones. 2016. Word Vector Compositionality based Relevance Feedback using Kernel Density Estimation. In *Proc. of CIKM 2016*. 1281–1290.
- [22] Mark Sanderson. 2010. Test Collection Based Evaluation of Information Retrieval Systems. *Found. Trends Inf. Retr.* 4, 4 (2010), 247–375. <http://dblp.uni-trier.de/db/journals/ftir/ftir4.html#Sanderson10>
- [23] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting Query Performance by Query-Drift Estimation. *ACM Trans. Inf. Syst.* 30, 2, Article 11 (2012).
- [24] Ellen M Voorhees and Donna K Harman. 2000. *The eighth text retrieval conference (TREC-8)*. Technical Report.
- [25] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. 2008. A simple and efficient sampling method for estimating AP and NDCG. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong (Eds.). ACM, 603–610.
- [26] Emine Yilmaz, Manisha Verma, Nick Craswell, Filip Radlinski, and Peter Bailey. 2014. Relevance and Effort: An Analysis of Document Utility. In *Proc. of CIKM'14*. 91–100.
- [27] Oleg Zendel, Anna Shtok, Fiana Raiber, Oren Kurland, and J. Shane Culpepper. 2019. Information Needs, Queries, and Query Performance Prediction. In *Proc. of SIGIR '19*. Association for Computing Machinery, New York, NY, USA, 395–404.
- [28] Ying Zhao, Falk Scholer, and Yohannes Tsegay. 2008. Effective Pre-Retrieval Query Performance Prediction Using Similarity and Variability Evidence. In *Proc. ECIR'08*. Springer-Verlag, Berlin, Heidelberg, 52–64.
- [29] Yun Zhou and W. Bruce Croft. 2007. Query Performance Prediction in Web Search Environments. In *Proc. of SIGIR'07*. 543–550.