Structure-aware Pre-Retrieval Performance Prediction on Query Affinity Graphs

Abstract. Pre-retrieval query performance prediction (QPP) estimates a query's effectiveness before retrieval, enabling tasks such as query routing. Prior pre-retrieval methods built on hand-crafted indicators treat queries independently and often scale poorly across collections and metrics; recent learning approaches often optimize relative difficulty without calibrated absolute estimates and still rely on single-query evidence, while neighborhood models assume a query behaves like the average of its nearest neighbors and cannot exploit multi-hop structure or handle heterogeneous neighborhoods. We propose a relational formulation that treats effectiveness as a smooth function over a query–query affinity graph structure, learning how information should propagate across related queries to produce context-aware predictions and extending to unseen queries by situating them within this graph structure. Our proposed method consistently outperforms classical and learning baselines, including higher rank correlations on MS MARCO Dev (Spearman $\rho = 0.409$) and DL-Hard (Spearman $\rho = 0.440$), and competitive results on TREC DL 2020 (higher Pearson and Spearman), indicating that our approach provides an effective basis for QPP.

1 Introduction

Retrieval systems often show non-uniform effectiveness over different queries [19]. Some are straightforward and achieve high nDCG or MRR, while others are ambiguous, rare, or underspecified and fare poorly under the same configuration [3, 9, 27, 4, 1, 22]. Anticipating this variation enables practical decisions such as routing hard queries to stronger rankers, selectively applying expansion or reranking, allocating compute adaptively, and designing risk-aware systems [24, 20]. *Query Performance Prediction (QPP)* aims to estimate how well a retrieval method will perform for a given query [8]. There are two classes of QPP methods, *post-retrieval* and *pre-retrieval* [8, 15]. Post-retrieval methods use evidence from the retrieved list and are often accurate but require running the pipeline [16]. Pre-retrieval methods avoid this cost and rely on query-only and collection-level signals. Classic pre-retrieval indicators include lexical statistics such as IDF [18] and ICTF [18], distributional or entropy measures such as SCS [17] and VAR [29], and term association signals such as PMI [14]. These signals are inexpensive and can be predictive on specific datasets, yet they treat queries largely in isolation, and their effectiveness varies across collections and query types [8, 13].

Learning-based *pre-retrieval* methods map pre-retrieval evidence to expected effectiveness [2, 5, 6]. Recent work on *disentangled QPP*, known as `CoDiR-QPP`, learns representations that separate difficulty-related factors and then orders queries by predicted difficulty instead of regressing absolute scores [21]. This shifts the focus from hand-crafted indicators to data-driven estimation that can

adapt to the target collection and metric. A complementary pre-retrieval approach, namely `QSD-QPP`, builds a local subspace around each query using nearest neighbors in an embedding space and interpolates historical effectiveness within that neighborhood [7]. This leverages the idea that similar queries carry informative performance signals, but it implicitly assumes that a query's effectiveness equals the average of its immediate neighbors [26]. That assumption can fail when neighborhoods are heterogeneous, when useful evidence resides beyond one hop neighbors, or when performance depends on broader relational patterns [12, 10]. Our approach differs by modeling how information should *propagate* over the network of queries rather than fixing it to a local average.

We adopt a relational view of *pre-retrieval* QPP that treats effectiveness as a function defined on a query–query *similarity graph*. The central assumption is *smoothness* at the scale of the graph, meaning that nearby queries should receive similar predictions while allowing changes across less connected regions. Instead of relying on local averaging, we learn how evidence should propagate through multiple hops so the estimate for a query reflects both its own features and the structure of its neighborhood. This provides a controlled form of *graph-based smoothing* that reduces variance on sparse or ambiguous queries and captures families of related intents that are not visible from a single hop [11]. The formulation is *inductive* because a new query is embedded, linked to its nearest neighbors, and evaluated by the same learned propagation, so predictions inherit information through local connectivity while remaining consistent with the learned structure.Our work's contribution in this paper can be enumerated as:

- We frame *pre-retrieval* QPP as learning a smooth effectiveness function over a query–query *similarity graph* and argue why *graph-based smoothing*, rather than local averaging, captures regularities that drive effectiveness across related queries.
- We instantiate this view with a simple and efficient method that constructs a k-nearest-neighbor query graph, performs learned *propagation* to obtain context-aware representations, and supports *inductive* prediction of performance for new queries.
- We present a comprehensive evaluation on MS MARCO Dev, TREC DL 2019 and 2020, and DL-Hard, showing consistent gains over classical *pre-retrieval* predictors and recent learning-based baselines, with ablations on encoder choice and graph architecture.

2 Proposed Approach

Task Definition. Pre-retrieval query performance prediction (QPP) assigns to each query a real-valued estimate of the effectiveness that a specified retrieval pipeline would achieve, before that pipeline is executed for the query. The problem is defined with respect to a fixed document collection, a fixed retrieval pipeline, and a fixed effectiveness metric, and the goal is to provide, for any query, an estimate of the resulting effectiveness score under that configuration.

More formally, let $\mathcal{C}$ be a document collection, $\mathcal{Q}$ a set of queries with relevance assessments $\mathcal{J}$, and R a retrieval pipeline that yields a ranking $\pi_{\mathsf{R}}(q)$

for any $q \in Q$. Let $\mathcal{M}$ be an effectiveness functional (e.g., MRR@k) mapping $(\pi_{\mathsf{R}}(q), \mathcal{J}(q))$ to a real value, i.e., $M_{\mathsf{R}}(q) \in \mathbb{R}$. A pre-retrieval QPP is a function

$$F: \ (C, q) \ \mapsto \ \widehat{M}_{\mathsf{R}}(q) \in \mathbb{R},$$

that, for any problem instance $(C, Q, \mathcal{J}, \mathsf{R}, \mathcal{M})$ and any $q \in Q$, produces an estimate $\widehat{M}_{\mathsf{R}}(q)$ of $M_{\mathsf{R}}(q)$ prior to computing $\pi_{\mathsf{R}}(q)$. The task is therefore to approximate the target mapping $q \mapsto M_{\mathsf{R}}(q)$ induced by the tuple $(C, \mathsf{R}, \mathcal{M}, \mathcal{J})$.

Approach Rationale. Pre-retrieval QPP predicts, for each query, the effectiveness of the retrieval method for that query before retrieval. Empirically, queries close in semantics or intent tend to yield similar effectiveness [23, 25], therefore, it might be possible to model the target function using neighborhoods of related queries rather than by isolated points. Treating queries as independent discards this structure and forces a predictor to discover regularities from sparse evidence. Considering a query *similarity graph* makes relational dependence explicit and reframes QPP as estimating a function that varies coherently along that graph, which better reflects how effectiveness behaves across families of related queries.

We adopt a *smoothness prior* on the query affinity graph so nearby queries are encouraged to receive similar predictions. *Graph based smoothing* approximates the conditional expectation of effectiveness given *retrieval independent* evidence. Learned propagation acts as data driven smoothing over multi-hop neighborhoods on the query affinity graph, aligning the influence of neighbors with effectiveness relevance and yielding predictions that change gradually under small perturbations of the query representation. As such, we predict each query's effectiveness as the value of a learned smooth function defined over a query–query relationship graph. The function is computed via propagation from the query's nearest neighbors, producing a context-aware effectiveness estimate.

Approach Formalization. We predict a query's effectiveness by exploiting how effectiveness co-varies across semantically related queries. Rather than treating each query in isolation, we place queries on a query affinity graph and learn a function whose values change coherently along this graph, i.e., the value at a node serves as the pre-retrieval effectiveness estimate for that query. Let $Q = \{q_1, \ldots, q_N\}$ be the query set and let $x_i \in \mathbb{R}^d$ denote the embedding of q_i from a pretrained encoder. We build an undirected weighted query similarity graph $G = (V, E, W)$ with $V = Q$, where edges connect k-nearest neighbors in embedding space. Weights use cosine similarity and are symmetrized:

$$w_{ij} = \begin{cases} \max\{0, \cos(x_i, x_j)\}, & \text{if } q_j \in \text{Top-}k(q_i) \text{ or } q_i \in \text{Top-}k(q_j), \\ 0, & \text{otherwise.} \end{cases}$$

Let $W = [w_{ij}]$, $D = \operatorname{diag}(W\mathbf{1})$, and define the symmetrically normalized adjacency with self-loops $\hat{A} = D^{-1/2}(W + I)D^{-1/2}$. The initial node features are $H^{(0)} = X = [x_1; \ldots; x_N] \in \mathbb{R}^{N \times d}$. We learn context-aware representations for queries in G through message passing:

$$H^{(\ell+1)} = \sigma\big(\hat{A}\, H^{(\ell)} W^{(\ell)}\big), \qquad \ell = 0, \ldots, L-1,$$

with trainable $W^{(\ell)}$ and nonlinearity σ. The final representation $H_i^{(L)}$ combines the intrinsic semantics of q_i with its multi-hop relational context. A linear readout maps $H_i^{(L)}$ to the pre-retrieval effectiveness estimate $\hat{M}_i = a^\top H_i^{(L)} + b$, where $a \in \mathbb{R}^{d_L}$ and $b \in \mathbb{R}$ are learned together with $\{W^{(\ell)}\}$. Parameters are fit on a labeled subset $Q_{\text{tr}} \subseteq Q$ by minimizing mean squared error, namely:

$$\mathcal{L} = \frac{1}{|Q_{\text{tr}}|} \sum_{q_i \in Q_{\text{tr}}} \left(\hat{M}_i - M_i\right)^2.$$

At inference and for an unseen query q_{new} with embedding x_{new}, we form edges to its k nearest neighbors using the same weighting and normalization, yielding the augmented operator $\hat{A}_{\text{aug}}$ and features $[X; x_{\text{new}}]$. Let $G_\Theta(A, X)$ denote the depth-L propagation that maps (A, X) to $H^{(L)}$ with parameters $\Theta = \{W^{(\ell)}\}_{\ell=0}^{L-1}$. The prediction evaluates the mapping on the augmented graph:

$$\hat{M}_{\text{new}} = a^\top \Big(G_\Theta\big(\hat{A}_{\text{aug}}, [X; x_{\text{new}}]\big) \Big)_{\text{new}} + b,$$

which provides a query context-aware effectiveness estimate consistent with the learned graph geometry, without modifying Θ, a, or b.

3 Experiments and Results

3.1 Experimental Setup

Architecture and implementation details follow a two-layer *graph convolutional network* with 128 hidden units and ReLU activation, implemented in PyTorch Geometric and trained with Adam at a learning rate of 2×10^{-4} for up to 20 epochs, with dropout 0.2 and layer normalization for regularization. Mini-batch training uses neighborhood sampling via `NeighborLoader` so each batch aggregates k hop messages within the sampled subgraph. All runs execute on a single NVIDIA RTX 6000 Ada GPU with 16 CPU cores and 50 GB RAM, and we compare three encoders for the initial query vectors, *DistilBERT*, *E5 Mistral 7B Instruct*, and *Contriever MSMARCO*, keeping all other settings fixed.

Datasets comprise the MS MARCO passage collection with 8.8M passages and over 500K queries with judgments for graph construction and training. Evaluation uses MS MARCO Dev with 6,980 queries, TREC Deep Learning 2019 with 43 queries, TREC Deep Learning 2020 with 53 queries, and DL Hard with 50 queries. Evaluation queries are held out during training and are used only for inference on the learned model.

Evaluation metrics measure agreement between predicted and actual effectiveness using Pearson r, Kendall τ, and Spearman ρ. Effectiveness is measured on BM25, using MRR@10 on MS MARCO Dev and nDCG@10 on TREC DL 2019, TREC DL 2020, and DL Hard, and all correlations are computed between predicted pre-retrieval scores and their actual effectiveness.

Baselines include classical pre-retrieval predictors based on lexical and collection statistics such as SCQ, SCS, VAR, PMI, IDF, ICTF, and information-content features, as well as graph or centrality inspired variants such as clustering coefficient and degree centrality. We also compare to three learning-based

Table 1: Correlation between predicted and actual effectiveness metrics. All reported correlations of our approach are statistically significant at $p < 0.01$.

	MS MARCO Dev			TREC DL 2019			TREC DL 2020			DL-Hard		
Model	*p*	*k*	*s*	*p*	*k*	*s*	*p*	*k*	*s*	*p*	*k*	*s*
SCQ [29]	0.012	0.011	0.014	0.298	0.116	0.162	0.226	0.076	0.132	0.264	0.127	0.179
SCS [17]	0.074	0.037	0.049	0.370	0.194	0.287	0.401	0.272	0.397	0.109	0.106	0.140
VAR [29]	0.074	0.062	0.083	0.197	0.107	0.152	0.107	0.059	0.077	0.053	0.016	0.035
PMI [14]	0.019	0.017	0.023	0.090	0.097	0.017	0.017	0.040	0.056	0.014	0.022	0.031
IDF [18]	0.125	0.116	0.154	0.311	0.158	0.245	0.395	0.245	0.353	0.075	0.111	0.125
ICTF [18]	0.112	0.114	0.152	0.309	0.153	0.240	0.287	**0.345**	0.330	0.073	0.107	0.115
CC [5]	0.080	0.065	0.085	0.103	0.099	0.055	0.034	0.106	0.026	0.098	0.103	0.141
DC [5]	0.102	0.107	0.144	0.065	0.095	0.053	0.035	0.091	0.035	0.112	0.123	0.165
IEF [5]	0.085	0.094	0.104	0.156	0.187	0.166	0.056	0.064	0.081	0.189	0.140	0.191
MRL [28]	0.028	0.024	0.032	0.325	0.245	0.373	0.003	0.009	0.006	0.013	0.049	0.057
CoDiR-QPP [21]	-	0.260	0.385	-	0.227	0.311	-	0.265	0.384	-	0.221	0.335
QSD-QPP [7]	0.018	0.172	0.227	0.096	0.100	0.144	0.323	0.203	0.298	**0.401**	0.241	0.341
Ours	**0.278**	**0.287**	**0.409**	**0.386**	**0.289**	**0.413**	**0.436**	0.272	**0.414**	0.122	**0.285**	**0.440**

methods: `MRL` [28], a supervised regression baseline that combines standard pre-retrieval cues to predict effectiveness; `CoDiR-QPP` [21], which estimates relative difficulty via pairwise ordering and therefore reports rank correlations; and `QSD-QPP` [7], which interpolates historical effectiveness within a nearest-neighbor query subspace. All baselines are re-run under the same retrieval backend and evaluation protocol for comparability, with hyperparameters following the settings reported in the respective papers.

Code and Data. We make our repository publicly available for reproducibility purposes `https://github.com/paper007submission-creator/KG.git`.

3.2 Findings

We structure the analysis around three research questions, namely (**RQ1**) does the proposed graph-based pre-retrieval QPP outperform lexical and learning baselines on MS MARCO Dev, TREC DL 2019/2020, and DL-Hard in terms of Pearson r, Spearman ρ, and Kendall τ? (**RQ2**) How sensitive is performance to the choice of encoder backbone (DistilBERT, E5-Mistral-7B-Instruct , Contriever-MSMARCO)? (**RQ3**) What is the impact of the message-passing architecture (GCN, GAT, GraphSAGE) on accuracy and stability?

Findings on RQ1. As shown in Table 1, across MS MARCO Dev, TREC DL 2019/2020, and DL-Hard, our proposed approach exhibits the strongest overall agreement with ground-truth effectiveness. It leads on MS MARCO Dev and TREC DL 2019 across Pearson r, Kendall τ, and Spearman ρ. On TREC DL 2020 it remains competitive, attaining higher r and ρ while ICTF achieves the best τ. On DL-Hard it delivers the highest rank correlations ($\tau = 0.285$, $\rho = 0.440$) and outperforms recent learning methods (e.g., +0.105 in ρ over *CoDiR–QPP)*, indicating better utility for ordering query difficulty. As such, the results show that our proposed approach outperforms lexical and learning baselines overall, with strong and stable rank correlations across benchmarks. While ICTF attains the best Kendall τ on TREC DL 2020, it does not show comparably strong performance on other datasets or metrics, and hence our method shows the most *consistent* strong performance compared to the other baselines.

Table 2: Correlation results for different encoders. Best results in **bold**.

	MS MARCO Dev			TREC DL 2019			TREC DL 2020			DL-Hard		
Model	p	k	s	p	k	s	p	k	s	p	k	s
DistilBERT	0.278	0.287	0.409	**0.386**	**0.289**	**0.413**	**0.436**	**0.272**	**0.414**	0.122	0.285	0.440
Mistral	**0.383**	**0.350**	**0.495**	0.351	0.229	0.340	0.258	0.170	0.244	0.332	0.288	**0.445**
Contriever	0.279	0.294	0.421	0.246	0.212	0.314	0.343	0.272	0.384	**0.382**	**0.291**	0.417

Table 3: Correlation results for different GNN architectures. Best is in **bold**.

	MS MARCO Dev			TREC DL 2019			TREC DL 2020			DL-Hard		
Model	p	k	s	p	k	s	p	k	s	p	k	s
GCN	0.278	0.287	0.409	**0.386**	**0.289**	**0.413**	**0.436**	0.272	**0.414**	0.122	**0.285**	**0.440**
GAT	**0.390**	**0.318**	**0.451**	0.228	0.187	0.253	0.323	0.248	0.370	**0.366**	0.278	0.419
GraphSAGE	0.251	0.311	0.355	0.184	0.118	0.174	0.366	**0.263**	0.380	0.135	0.130	0.183

Findings on RQ2. According to the results in Table 2, across different encoders, the strongest overall performance comes from *DistilBERT*, which reports consistently high correlations on TREC DL 2019 (r = 0.386, τ = 0.289, ρ = 0.413) and TREC DL 2020 (r = 0.436, ρ = 0.414). *Mistral* peaks on MS MARCO Dev (r = 0.383, τ = 0.350, ρ = 0.495) and attains the highest ρ on DL-Hard (0.445), but is less stable across datasets. *Contriever* remains reliable on TREC DL 2020 and DL-Hard (e.g., DL-Hard r = 0.382, τ = 0.291), indicating that corpus-aligned contrastive training can help on harder queries. Overall, the results suggest that a consistent semantic space that preserves local neighborhoods matters more than model size, with compact distilled embeddings offering the best balance of accuracy and stability.

Findings on RQ3. Based on Table 3, across different graph architectures, *GCN* offers the best balance of accuracy and stability where it shows strong performance on TREC DL 2019 (r = 0.386, τ = 0.289, ρ = 0.413), on TREC DL 2020 in Pearson and Spearman (r = 0.436, ρ = 0.414), and on DL-Hard in rank correlations (τ = 0.285, ρ = 0.440). *GAT* shows its best on MS MARCO Dev (r = 0.390, τ = 0.318, ρ = 0.451) and remains competitive elsewhere, suggesting benefits from learning neighbor importance, but it does not match GCN's cross-dataset consistency. *GraphSAGE* underperforms on most benchmarks, with a single strong Kendall correlation on TREC DL 2020 (τ = 0.263), indicating that sampling-based aggregation is less effective on our dense k-NN query graphs. These results support the hypothesis that normalized uniform aggregation aligns well to estimate a smooth effectiveness function over the query graph.

4 Concluding Remarks

We presented a relational formulation of *pre-retrieval* query performance prediction that treats effectiveness as a smooth function on a query–query affinity graph and operationalizes it via learned propagation with a linear readout. Across MS MARCO Dev, TREC DL 2019/2020, and DL-Hard, our approach consistently outperforms classical indicators and recent learning baselines. The ablations on encoder choice and graph architecture show that the gains stem from leveraging relational structure rather than model size or architectural complexity. The model is inductive, supporting unseen queries by linking them to the learned graph, and provides stable rank correlations on difficult benchmarks.

References

1. Arabzadeh, N., Bigdeli, A., Hamidi Rad, R., Bagheri, E.: Quantifying ranker coverage of different query subspaces. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2298–2302 (2023)
2. Arabzadeh, N., Khodabakhsh, M., Bagheri, E.: Bert-qpp: Contextualized pre-trained transformers for query performance prediction. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. p. 2857–2861. CIKM '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3459637.3482063
3. Arabzadeh, N., Mitra, B., Bagheri, E.: Ms marco chameleons: Challenging the ms marco leaderboard with extremely obstinate queries. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 4426–4435 (2021)
4. Arabzadeh, N., Yan, X., Clarke, C.L.A.: Predicting efficiency/effectiveness trade-offs for dense vs. sparse retrieval strategy selection. CoRR **abs/2109.10739** (2021), https://arxiv.org/abs/2109.10739
5. Arabzadeh, N., Zarrinkalam, F., Jovanovic, J., Al-Obeidat, F., Bagheri, E.: Neural embedding-based specificity metrics for pre-retrieval query performance prediction. Information Processing Management **57**(4), 102248 (2020). https://doi.org/10.1016/j.ipm.2020.102248
6. Arabzadeh, N., Zarrinkalam, F., Jovanovic, J., Bagheri, E.: Neural embedding-based metrics for pre-retrieval query performance prediction. In: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II. p. 78–85. Springer-Verlag, Berlin, Heidelberg (2020). https://doi.org/10.1007/978-3-030-45442-5_10
7. Bigdeli, A., Ebrahimi, S., Arabzadeh, N., Salamat, S., SeyedSalehi, S., Khodabakhsh, M., Zarrinkalam, F., Bagheri, E.: Query performance prediction using neural query space proximity. ACM Trans. Intell. Syst. Technol. (Sep 2025). https://doi.org/10.1145/3762197
8. Carmel, D., Yom-Tov, E.: Estimating the Query Difficulty for Information Retrieval, Synthesis Lectures on Information Concepts, Retrieval, and Services, vol. 2. Morgan & Claypool Publishers (2010). https://doi.org/10.2200/S00237ED1V01Y201006ICR016
9. Carmel, D., Yom-Tov, E., Darlow, A., Pelleg, D.: What makes a query difficult? In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 390–397 (2006)
10. Chawla, N.V., Karakoulas, G.: Learning from labeled and unlabeled data: An empirical study across techniques and domains. Journal of Artificial Intelligence Research **23**, 331–366 (Mar 2005), http://dx.doi.org/10.1613/jair.1509
11. Chong, Y., Qin, T., Tang, Q., Wei, S., Xu, M., Tang, X., Zhang, Z.: Graph-based semi-supervised learning: A review. Neurocomputing **408**, 216–230 (2020). https://doi.org/10.1016/j.neucom.2020.04.113
12. Eswaran, D., Kumar, S., Faloutsos, C.: Higher-order label homogeneity and spreading in graphs (2020), https://arxiv.org/abs/2002.07833
13. Faggioli, G., Formal, T., Marchesin, S., Clinchant, S., Ferro, N., Piwowarski, B.: Query performance prediction for neural ir: Are we there yet? In: European Conference on Information Retrieval. pp. 232–248. Springer (2023)

14. Hauff, C.: Predicting the effectiveness of queries and retrieval systems. In: SIGIR Forum. vol. 44, p. 88 (2010)
15. Hauff, C., Hiemstra, D., de Jong, F.: A survey of pre-retrieval query performance predictors. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008. pp. 1419–1420 (2008). https://doi.org/10.1145/1458082.1458311
16. Hauff, C., Hiemstra, D., de Jong, F.: A survey of pre-retrieval query performance predictors. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management. p. 1419–1420. CIKM '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1458082.1458311
17. He, B., Ounis, I.: Inferring query performance using pre-retrieval predictors. In: String Processing and Information Retrieval, 11th International Conference, SPIRE 2004, Padova, Italy, October 5-8, 2004, Proceedings. pp. 43–54 (2004). https://doi.org/10.1007/978-3-540-30213-1_5
18. Kwok, K.L.: A new method of weighting query terms for ad-hoc retrieval. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'96, August 18-22, 1996, Zurich, Switzerland (Special Issue of the SIGIR Forum). pp. 187–195 (1996). https://doi.org/10.1145/243199.243266
19. Meng, C., Arabzadeh, N., Aliannejadi, M., de Rijke, M.: Query performance prediction: From ad-hoc to conversational search. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 2583–2593. SIGIR '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3539618.3591919
20. Meng, C., Arabzadeh, N., Aliannejadi, M., de Rijke, M.: Query performance prediction: From ad-hoc to conversational search. arXiv preprint arXiv:2305.10923 (2023)
21. Salamat, S., Arabzadeh, N., Seyedsalehi, S., Bigdeli, A., Zihayat, M., Bagheri, E.: A contrastive neural disentanglement approach for query performance prediction. Mach. Learn. **114**(4) (Feb 2025). https://doi.org/10.1007/s10994-025-06752-x
22. Saleminezhad, A., Arabzadeh, N., Beheshti, S., Bagheri, E.: Context-aware query term difficulty estimation fornbsp;performance prediction p. 30–39 (2024). https://doi.org/10.1007/978-3-031-56066-8_4
23. Saleminezhad, A., Arabzadeh, N., Rad, R.H., Beheshti, S., Bagheri, E.: Robust query performance prediction for dense retrievers via adaptive disturbance generation. Mach. Learn. **114**(3) (Feb 2025). https://doi.org/10.1007/s10994-024-06659-z
24. Sugiura, A., Etzioni, O.: Query routing for web search engines: architecture and experiments. Comput. Netw. **33**(1), 417–429 (Jun 2000). https://doi.org/10.1016/S1389-1286(00)00059-1
25. Tian, F., Ganguly, D., Macdonald, C.: Revisiting query variants: The advantage of retrieval over generation of query variants for effective qpp (2025), `https://arxiv.org/abs/2510.02512`
26. Widmann, N., Verberne, S.: Graph-based semi-supervised learning for text classification. In: Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval. p. 59–66. ICTIR '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3121050.3121055
27. Yom-Tov, E., Fine, S., Carmel, D., Darlow, A.: Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 512–519 (2005)

28. Zamani, H., Bendersky, M.: Multivariate representation learning for information retrieval. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 163–173. SIGIR '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3539618.3591740
29. Zhao, Y., Scholer, F., Tsegay, Y.: Effective pre-retrieval query performance prediction using similarity and variability evidence. In: Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings. pp. 52–64 (2008). https://doi.org/10.1007/978-3-540-78646-7_8