

Revisiting Query Variants: The Advantage of Retrieval Over Generation of Query Variants for Effective QPP

Fangzheng Tian, Debasis Ganguly and Craig Macdonald

University of Glasgow, Scotland, United Kingdom

Abstract

Leveraging query variants (QVs), i.e., queries with potentially similar information needs to the target query, has been shown to improve the effectiveness of query performance prediction (QPP) approaches. Existing QV-based QPP methods generate QVs facilitated by either query expansion or non-contextual embeddings, which may introduce topical drifts and hallucinations. In this paper, we propose a method that retrieves QVs from a training set (e.g., MS MARCO) for a given target query of QPP. To achieve a high recall in retrieving queries with the most similar information needs as the target query from a training set, we extend the directly retrieved QVs (1-hop QVs) by a second retrieval using their denoted relevant documents (which yields 2-hop QVs). Our experiments, conducted on TREC DL'19 and DL'20, show that the QPP methods with QVs retrieved by our method outperform the best-performing existing generated-QV-based QPP approaches by as much as around 20%, on neural ranking models like MonoT5.

Keywords

Query Performance Prediction (QPP), Query-Variant-based QPP, Retrieval of Query Variants

1. Introduction

Query performance prediction (QPP) is the task of estimating the quality of retrieval results for a **target query** without relevance judgements. As neural ranking models are increasingly applied in IR, developing effective QPP methods for them has become an important task. However, the common unsupervised post-retrieval QPP baselines, such as NQC [1], usually perform poorly for neural rankers where the retrieval score distribution is not suitable for distinguishing retrieval quality based on statistical characteristics [2]. We argue that the approach of Shtok et al. [3] that leverages the QPP results of the queries with similar information needs as the target query (i.e. **query variants**, abbr. QVs), which is useful for improving the QPP baselines' effectiveness, can be potentially suitable for QPP in neural rankers.

To provide reliable signals for incorporating the QPP of the target query, **keeping the information needs relatively invariant across the QVs is a core task in QV-based QPP** [4]. The existing methods mainly use feedback or non-contextual embeddings to automatically generate QVs [2], which are prone to contain hallucinations and information drifts [5]. To tackle this difficulty, in this paper, we propose a novel QV retrieval method from an IR training set. According to our experiments, QPP using the QVs retrieved by our proposed methods shows better effectiveness than baselines.

The goal of retrieving QVs from a training set is to find the queries that are most similar to the target one, which will then be included in a candidate QV set for QPP. However, due to the inherent deficiency of representing an information need by a short query [4], directly retrieving QVs based on the target query may be incapable of getting a high recall in retrieving similar queries. To address this problem, we propose to use the relevant documents of the directly retrieved QVs (referred to as **1-hop QVs**), which are available as a part of the MS MARCO training set, to include more candidate QVs (referred to as **2-hop QVs**). Specifically, the relevant documents of the 1-hop QVs are formulated as queries to retrieve 2-hop QVs from the query index, which is a process similar to relevance feedback [6]. Since

ECIR'25: QPP++ Workshop, April 06–10, 2025, Lucca, Italy

*Corresponding author.

✉ f.tian.1@research.gla.ac.uk (F. Tian); debasis.ganguly@glasgow.ac.uk (D. Ganguly); craig.macdonald@glasgow.ac.uk (C. Macdonald)

✉ 0009-0000-3282-0220 (F. Tian); 0000-0003-0050-7138 (D. Ganguly); 0000-0003-3143-279X (C. Macdonald)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

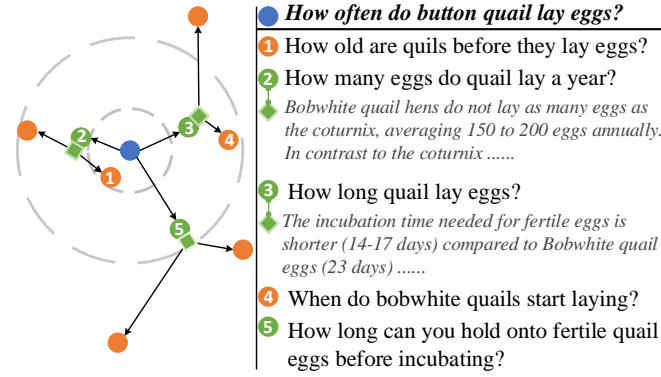


Figure 1: An visualisation of the idea behind QPP with *retrieved* QVs. For a target query (the blue point), the first step is to retrieve a list of *1-hop* QVs (the green points) from an index constructed from the training query set. In the second step, the relevant document associated with each *1-hop* candidate (the italicised texts) is used as a query to further retrieve *2-hop* QVs (the amber points). The combination of *1-hop* and *2-hop* QVs comprises the target query’s 2-hop neighbourhood, which defines the range of the final candidate QV set.

2-hop QVs implicitly contain relevant information from the 1-hop QVs, they can still be topically related to the information the target query’s information needs and act as useful references in a QV-based QPP model. The idea of the neighbourhood expansion from 1-hop QVs to 2-hop QVs is depicted in Figure 1.

The candidate QVs within the 2-hop neighbourhood (illustrated in Figure 1) comprise the final QV candidate set. In the next step, the candidate QVs are re-ranked according to the similarities between their top retrieved documents and the target query’s retrieval result (by means such as RBO [7]), which is a more effective indicator of the similarity between information needs than simply measuring similarity between queries. The top-ranked QVs after the re-ranking are subsequently used in the QV-based QPP. In our experiments, compared with existing QV-based QPP methods, the QPP estimations using the QVs retrieved by our proposed methodology largely enhance the QPP effectiveness for neural retrieval models. For instance, our QPP method based on retrieved QVs improves the QPP effectiveness on MonoT5 by around 20% in comparison to that of generated QVs. Therefore, the innovation in exploiting the resources provided by IR training sets to retrieve QVs that can be helpful for QV-based QPP is the main **contribution** of our work.

In the next section, we review the work related to QV-based QPP. Section 3 introduces the methodology of QV retrieval. Section 4 is about the settings and results of the experiments which test the QPP approaches based on the QVs retrieved by our proposed method. Section 5 concludes this paper with directions for future work.

2. Related Work

Unsupervised QPP methods. QPP models can be categorised into pre- and post-retrieval ones based on whether the retrieval results are available to a prediction model. In **pre-retrieval QPP** methods, a query’s performance is estimated by only using the information from the query and the collection statistics, and thus their prediction objective is either the specificity [8] or difficulty [9] of a query. It can be interpreted as the prediction of the expected retrieval quality which is not specific to any retrieval models. In contrast, **post-retrieval QPP** methods are capable of utilising useful signals from the content and the scores of the top-retrieved documents to predict the retrieval quality [10]. The underlying hypothesis common to various post-retrieval predictors is the separability of the top retrieved documents from the remaining ones: a high separability is a likely indicator of the relevance of the top-retrieved documents [11]. To avoid the effect of outliers, some QPP approaches randomly sample subsets of the top-retrieved documents and then aggregate the predictions over each subset [12].

QV-based QPP methods. In principle, any statistical estimation method can be improved with the availability of a large number of observation points [13]. In the context of QPP, since a post-retrieval estimator relies on the computation of statistical measures such as the variance of the retrieval score [1], the estimate for a query can be improved by leveraging information from the queries with information needs similar to the target query (referred to as query variants (QVs) in [4]). Usually, an unsupervised QV-based QPP method first generates QVs using relevance feedback [14] or word2vec embeddings [15], as done by WRIG [2]. Then, it uses each generated QV to retrieve *reference lists* of documents for QPP [3]. A base predictor, e.g., NQC, is applied to each reference list. The weighted average across those predictions is yielded as the final prediction.

Apart from QV generation, retrieving QVs has also been performed by Ebrahimi et al. [16] in their supervised QPP method. They concatenate the retrieved QVs and their estimated performances (based on the relevance judgements in the training set) to the target query and its retrieval results. In this way, neighbourhood information is encoded in the target query’s expression to help the cross-encoder to predict its performance. However, the training queries usually only have shallow annotated relevance judgments, which leads to the concatenated QVs’ performances being unreliable. Therefore, their training data might be unavoidably noisy, which can potentially influence QPP effectiveness.

Research Gaps. Although Ebrahimi et al. [16] also retrieve QVs from a training set, they did not deeply investigate how to obtain high-quality QVs. In our research, instead of merely using 1-hop QVs which are directly retrieved by the target query, we explore retrieving 2-hop QVs to extend the candidate QV set to improve the effectiveness of QV-based QPP. The details about our QV retrieval model follow in the next section.

3. Query Variant Retrieval based QPP Methodology

This section first provides a brief technical introduction to the QV-QPP model of [4] (Section 3.1), following which we describe the proposed QV retrieval component in detail (Section 3.2).

3.1. Smoothing Estimates from QVs

A post-retrieval QPP model ϕ takes a target query Q and its retrieval result $L_M(Q)$ obtained by a **target retriever** M as input. ϕ outputs a score to indicate the quality of $L_M(Q)$. Formally speaking, $\phi : L_M(Q) \mapsto \mathbb{R}$. In QV-based QPP models, the predictor function ϕ depends not only on a target query and its retrieved documents but also on a set of query variants (QVs) that contain similar information needs as Q . Formally speaking, $\phi^+ : L_M(Q), \mathcal{E}(Q) \mapsto \mathbb{R}$, where $\mathcal{E}(Q)$ denotes a set of variants of Q .

We develop our QV-based QPP framework based on the method proposed in [4], which aggregates the QPP estimates for Q and its QVs by linear combination. Formally, the predictor in [4] is defined as

$$\phi_M^+(L_M(Q), \mathcal{E}(Q)) = (1 - \lambda)\phi(L_M(Q)) + \lambda \sum_{Q' \in \mathcal{E}(Q)} \frac{\sigma(Q, Q')\phi(L_M(Q'))}{\bar{\sigma}(Q)}, \quad (1)$$

where σ is a function measuring the similarity of information needs between a pair of queries $\sigma(Q, Q')$, and $\bar{\sigma}(Q) = \sum_{Q' \in \mathcal{E}(Q)} \sigma(Q, Q')$, which is the average over all the QVs in $\mathcal{E}(Q)$. Equation (1) linearly combines the QPP estimations for the target query and the QVs (the same base predictor ϕ is used for both). The **relative importance** of the estimates about QVs is controlled by the parameter $\lambda \in [0, 1]$. The QPP about QVs is a weighted average across all the QVs in \mathcal{E} , where the similarity measure between a pair of queries $\sigma(Q, Q')$ is used to determine the relative contribution (weight) of each QV Q' . Following Zendel et al. [4], we use rank-biased overlap (RBO) similarity [7] as σ , as this QPP estimator puts a higher emphasis on the contribution from the QVs that contains similar information needs as the target query.

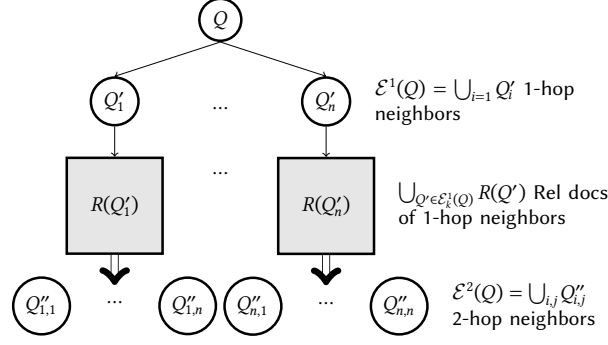


Figure 2: Schematic representation of the idea of collecting second-hop neighbourhood queries by extending the set of 1-hop neighbours $\mathcal{E}^1(Q)$ with the set of relevant documents $R(Q')$ for each first-hop query Q' .

In Equation (1), base predictor ϕ yields the QPP estimates for QVs based on their retrieval results obtained by the same retriever M as the target query. Our proposed QPP framework generalises Equation (1) by setting an **internal retriever** for the QVs:

$$\phi_{M'}^+(L_M(Q), \mathcal{E}(Q)) = (1 - \lambda)\phi(L_M(Q)) + \lambda \sum_{Q' \in \mathcal{E}(Q)} \frac{\sigma(Q, Q')\phi(L_{M'}(Q'))}{\bar{\sigma}(Q)}, \quad (2)$$

where M' is an internal retriever of ϕ^+ , which is allowed to be separate from the target retriever M . Therefore, when M is a neural ranker, M' can still be an efficient lexical retriever. Moreover, considering the base predictor ϕ (e.g. NQC) can be ineffective for neural retrievers [2], this separation can bring additional reliability to the QPP estimation for the target query. In our experiments, we set $M' \equiv BM25$.

3.2. Retrieving the Query Variants

In our proposed method, QVs in \mathcal{E} are retrieved from the training set. The key motivation for using retrieved QVs, instead of generating them based on relevance feedback or non-contextual embeddings, is that these queries are formulated by real users and, thereby, are more likely to represent real-life information needs and less likely to be hallucinatory.

1-hop QV retrieval. In the 1-hop QV-based QPP approach, the QV set in Equations (2) corresponds to $\mathcal{E}^1(Q)$, which is a set of top- k most similar queries directly retrieved from a query collection \mathcal{Q} by the target query Q . This makes $\mathcal{E}^1(Q)$ depending on a similarity function that, given a target query Q , is used to compute its k nearest neighbours, which is denoted as

$$\mathcal{E}^1(Q) \equiv \mathcal{N}_k(Q, \sigma) \subset \mathcal{Q}, \quad (3)$$

where \mathcal{N}_k is a neighbourhood of size k , and σ is a similarity function. In practice, we employ RBO as σ , similar to experiments conducted in [2, 4]. However, since the RBO similarities are the rank-biased overlap between the retrieval results of two queries, it can only be calculated once we have got the retrieval results for them. Therefore, the process of retrieving candidate QVs and then finding the k QVs most similar to Q is like retrieval and re-ranking in retrieving documents. Firstly, we obtain a candidate set of $n(> k)$ top-retrieved queries using an indexed query collection \mathcal{Q} . Then, we compute each candidate's RBO (with respect to the target query) to rerank the list and extract the top- k QVs with the largest RBO scores. This set forms the set of 1-hop QVs. As specific choices for retrieving the 1-hop candidates, we employ both a BM25 sparse index and a dense index of SBERT [17] embeddings.

2-hop QV retrieval. A limitation of retrieving 1-hop QVs is that the queries most similar to Q may not be easily retrieved based on Q itself. The root of this low recall in retrieving high-quality QVs is

the ambiguity of representing an information need by a short query. To alleviate this limitation, we leverage the associated relevant documents of each 1-hop query to expand the candidate QV set with 2-hop QVs, which has a bigger chance of retrieving more high-quality QVs.

Let us denote the set of relevant documents for each query $Q' \in \mathcal{Q}$ by $R(Q')$. We argue that this relevance information for a 1-hop QV Q' can further help to identify the actual information need of Q' , which, in turn, is potentially close to that of Q . In the light of this idea, we (1) construct a pseudo-query Q'_D from each document $D \in R(Q')$, then (2) execute Q'_D on the index of queries \mathcal{Q} to obtain another list of “2-hop” queries $\mathcal{E}(Q'_D)$. The idea is visualised in Figure 2, where the set of relevant documents for each first-level QV is used to retrieve the second-level ones (the bottom row of queries). Formally,

$$\mathcal{E}^2(Q) = \bigcup_{Q' \in \mathcal{N}_k(Q, \sigma)} \bigcup_{D \in R(Q')} \mathcal{N}_k(Q_D, \sigma), \quad (4)$$

where $\mathcal{E}^2(Q)$ denotes the set of 2-hop QVs. We then merge the $\mathcal{E}^1(Q)$ and $\mathcal{E}^2(Q)$ as the final candidate QV set from where we select the QVs for QPP:

$$\mathcal{E}^+(Q) = \mathcal{E}^1(Q) \cup \mathcal{E}^2(Q), \quad (5)$$

where $\mathcal{E}^+(Q)$ is the merged QV set containing both 1-hop and 2-hop retrieved QVs of Q . Similar to the 1-hop approach, we compute the RBO scores across the QVs in $\mathcal{E}^+(Q)$ and take the QVs with the highest- k RBO scores as the final nearest- k QVs. Those QVs can subsequently be used in the QPP framework as described in Equation (2).

4. Evaluation

4.1. Research Questions

The objective of our experiments is to confirm if the QVs retrieved by the proposed method can lead to more effective estimations of retrieval quality. Therefore, we raise the following research questions about the effectiveness of applying retrieved QVs for QPP and the necessity of using 2-hop QVs.

- **RQ-1:** *Are retrieved query variants from a training set more effective than query variants generated by non-contextual embedding or relevance feedback approaches (namely the ones used in Datta et al. [2])?*
- **RQ-2:** *Can the information from the relevant documents of 1-hop QVs be useful to yield a candidate QV set leading to better QPP estimates?*

4.2. Experiment Settings

Datasets. We conduct our experiments on the MS MARCO passage collection benchmark [18] with queries from TREC DL’19 [19] and DL’20 [20] test sets. The QV candidates are retrieved from the MS MARCO training set containing over 800K examples of query-relevant document pairs.

Target Retrievers. Since the objective of this research is enhancing QPP effectiveness for neural IR, we evaluate our proposed QPP approach and baselines on the following neural ranking models: (i) **BM25»MonoT5** [21]¹: After retrieving an initial candidate of 100 documents with BM25 (using a sparse index), we re-rank the top 100 results by MonoT5, a cross-encoder model. (ii) **BM25»BERT** [22]²: BERT is a contextualised language model that can be used to obtain embeddings of queries and documents utilising the classification token (CLS) [23]. We use a model that was fine-tuned on the MS MARCO Passage dataset to re-rank the top 100 BM25-retrieved passages. (iii) **TCT-ColBERT**[24]: A neural retriever trained by distilling knowledge from ColBERT [25]. We use this model to test the performance of our QPP methods for end-to-end neural retrieval.

¹Huggingface id castorini/monot5-base-msmarco

²Huggingface id capreolus/bert-base-msmarco

Internal Retriever (Retrieval of Documents with QVs). As described in Section 3.1, the internal retriever M' in Equation (2) (i.e., the model which is used to retrieve for the QVs) is always BM25 in our experiments.

Related Query (QV) Retrieval Models. QVs are retrieved from the MS MARCO training query set by: (i) **BM25**: A term-weighting, document length normalisation-based classic IR model. (ii) **Sentence-BERT** [17]: A CLS-pooling-based bi-encoder BERT model pre-trained on the MS MARCO training set triples, which is abbreviated as **SBERT** in the rest of this paper. For all the QV-based QPP methods, we re-rank the candidate QVs by their **RBO** [7] similarities with the target query and then take the top- k ones for the smoothing estimation in Equation (2).

Parameters and Settings. We employ **Kendall's τ** as the QPP effectiveness evaluation metric. Since different target IR measures can cause significant variations in QPP outcomes [26], we report our experiments with two target IR metrics, namely AP@100 and nDCG@10 - the latter being top-rank precision-focused. For computing AP, documents with a relevance label of 2 or higher are considered relevant [19].

We evaluate the experimented QPP methods with a 2-fold setup: in the first fold, hyper-parameters are trained on DL'19 and tested on DL'20, and the other way around in the second fold. The hyper-parameters trained in every QV-based QPP method (named with 'QV*') are: (i) k , the number of QVs used in the smoothing estimation, and (ii) λ , the relative importance of the QV-based QPP smoothing (see Equation (2) for more details).

4.3. Proposed QPP Methods and Baselines

As a common naming convention, we use the common prefix 'QV' to name the QV-based QPP approaches. As QPP baselines, we investigate the following approaches:

- **Predictors without QVs:** (i) **NQC** [1], which estimates the retrieval quality by measuring the skewness of retrieval score distribution in the top-ranked documents, and (ii) **UEF** [12], which samples subsets from the top-ranked documents for estimating the robustness of retrieval quality. We choose NQC as the base predictor for UEF, as described on [2]. Both NQC and UEF also serve as base predictors for the QV-based methods.
- **Predictors using generated QVs:** We employ the methods proposed in [4], where the QVs (set \mathcal{E} of Equation (2)) are generated by the methodology proposed in [2]. There are two methods with different ways of QV generation: (i) **QV-RLM**: generating QVs by query expansion via a Relevant Language Model (RLM), which is RM3 [14] in our implementation, (ii) generating QV by adding semantically similar terms by using the non-contextual embedding space of Word2Vec [15].
- **Supervised Predictors:** To compare our QV-based unsupervised QPP methods with their supervised counterpart, we implement the cross-encoder-based QPP method **BERTQPP** [27]. We also implement a variant of it, **BERTQPP-QV** [16], which considers QVs and their estimated query performance when encoding the target query.

Experimented proposed predictors with retrieved QVs. The key difference between our methods and the QV-based baselines (QV-RLM and QV-W2V) is the usage of retrieved QVs from a training set. We place an 'R' in the names of the methods with retrieved QVs following 'QV'. Additionally, we set the superscript of 'R' as '1' or '2' to distinguish between the methods with 1-hop or 2-hop neighbourhood considered when retrieving the QVs. According to whether the query retriever is the lexical BM25 or semantic SBERT, we test the following four variants of the proposed method: (1) **QV-R¹-BM25**: This method uses 1-hop query variants only, i.e., only uses the set \mathcal{E} of 1-hop neighbouring queries (see Equation (3)) to smooth the predictor. The 1-hop QVs are retrieved by BM25. (ii) **QV-R²-BM25**: Like QV-R¹-BM25, it also uses BM25 as the query retriever. However, it extends the candidate QV set

Table 1

Comparisons between our proposed QPP method (‘Ours’) and the baselines (‘BL’) along with the two supervised methods (‘Supervised’). QPP effectiveness is measured by Kendall’s τ with AP@100 and nDCG@10 as the target metrics. Reported Kendall’s τ values are averaged over a 2-fold train-test split on TREC DL’19 and DL’20, with λ is tunable while k is fixed to 1, as described in Equation (2). The best results for each metric with a particular base QPP model are underlined. Comparing the best results obtained with NQC and UEF as the base QPP model, the better ones are bold-faced. A ‘*’ alongside one of our proposed QPP approaches over a target metric indicates that the reported result is statistically significant (by Fisher’s z-test) over the corresponding base predictor.

ϕ	Type	Method	BM25»MonoT5		BM25»BERT		TCT-ColBERT	
			AP- τ	nDCG- τ	AP- τ	nDCG- τ	AP- τ	nDCG- τ
NQC	BL	NQC	0.1673	0.0274	0.1278	0.0391	0.3991	0.2618
		QV-W2V	0.2685	0.2041	0.2395	0.1520	0.3923	0.2521
		QV-RLM	0.3308	0.1848	0.3045	0.1460	0.3920	<u>0.2848</u>
	Ours	QV- R^1 -BM25	0.3520	0.1918	0.3669*	0.2081	0.3954	0.2611
		QV- R^1 -SBERT	0.3361	0.2000	0.3558*	0.2016	0.4177	0.2561
		QV- R^2 -BM25	0.3694*	0.2298	0.3678*	0.2111	0.3878	0.2539
		QV- R^2 -SBERT	0.4033*	0.2573*	<u>0.4250*</u>	<u>0.2517*</u>	0.4022	0.2614
	BL	UEF	0.1679	0.0341	0.1323	0.0399	0.3908	0.2732
		QV-W2V	0.2674	0.1969	0.2290	0.1583	0.3892	0.2718
		QV-RLM	0.3247	0.1940	0.2846	0.1594	0.3845	0.2998
UEF	Ours	QV- R^1 -BM25	0.3438	0.1963	0.3567*	0.2247	0.3701	0.2782
		QV- R^1 -SBERT	0.3481	0.2044	0.3791*	0.2159	<u>0.4080</u>	0.2732
		QV- R^2 -BM25	0.3597	0.2325	0.3621*	0.2079	0.3832	0.2801
		QV- R^2 -SBERT	<u>0.4026*</u>	<u>0.2540*</u>	0.4375*	0.2630*	0.3619	0.2633
	Supervised	BERTQPP	0.2277	0.1746	0.1827	0.1328	0.2238	0.1326
		BERTQPP-QV	0.2432	0.1728	0.2051	0.1459	0.2529	0.1514

with 2-hop QV selection as shown in Equation (5) and Figure 2. The superscript ‘2’ on ‘R’ indicates 2-hop QVs are used in this method. (iii) **QV- R^1 -SBERT**: This is similar to QV- R^1 -BM25 but employs an embedding-space (semantic) approach to retrieve candidate 1-hop QVs. In particular, we use SBERT³ as the query retriever. (iv) **QV- R^2 -SBERT**: This method employs the same query retriever as QV- R^1 -SBERT. The difference is that it additionally adds 2-hop neighbours into the candidate QV set. Above retrieval pipelines are implemented by PyTerrier [28]

4.4. Main Observations

We first present the main findings of our approach (relative to the baselines) with a single query variant, i.e., we use $k = 1$ in the general QV-QPP framework (Equation (2)). The performances of our QPP approach over a range of different numbers of k are reported in 4.5. Table 1 shows a comparison between the baselines and the proposed approaches when $k = 1$. In this table, the two groups of rows correspond to the two experimented base predictors (denoted as ϕ) - NQC and UEF. Each row in this table is either a baseline or a variant of the proposed retrieved-QV-based method. The results of supervised BERT-QV are listed at the bottom of this table for comparison. We highlight the following observations to answer RQ1 and RQ2.

RQ1: Retrieved QVs yield better QPP effectiveness compared to RLM and W2V-based generated QVs. In general, the baseline QPP methods with W2V- or RLM-generated QVs, are more effective than the respective base predictors and the supervised predictors, especially for BM25»MonoT5 and BM25»BERT where the retrieval scores are bounded by [0, 1]. For example, when the retriever is

³Huggingface id sentence-transformers/all-MiniLM-L6-v2

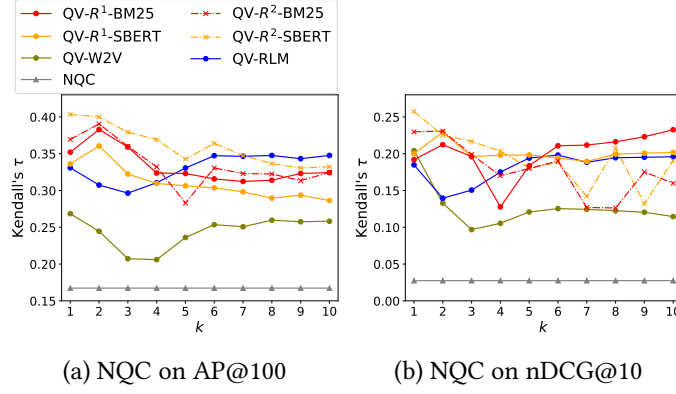


Figure 3: A complete spectrum of the results of Table 1 showing the effect of k for each tested QPP method. The base predictor is NQC.

BM25»MonoT5, compare the τ values for the AP@100 with NQC (0.1673) vs. QV-RLM (0.3308) in Table 1 (this also conforms to the observations reported in [2]). Our proposed retrieval-based QV approach outperforms the ones generated by either RLM and W2V. The difference between the proposed methods and QV-RLM (the best baseline method) is larger when the target metric is AP@100, e.g., with NQC as the base predictor, QV- R^1 -BM25 achieves a τ of 0.3669 outperforming QV-RLM (0.3045). The QV- R^1 -BM25 result is significantly better than NQC (Fisher’s z test at a 99% confidence interval).

RQ2: Utilising relevant information of 1-hop QVs to retrieve 2-hop QVs enhances QPP effectiveness. This is observed by comparing the results between these two pairs of QPP methods - 1-hop and 2-hop for each QV retrieval method. For instance, Table 1 shows that with NQC as the base predictor, the 2-hop method QV- R^2 -SBERT outperforms the 1-hop method QV- R^1 -SBERT (0.4033 vs. 0.3361). This improvement in correlation is at 80% confidence interval by Fisher’s z test.

4.5. Parameter Sensivity

We conduct experiments to investigate the relation between the two hyper-parameters: k - the number of QVs, and λ - the relative importance of the QVs in the final QPP estimate described in Equation (2).

Effect of varying k w.r.t. λ s in QPP based on retrieved QVs. Figure 3 shows the QV-based QPP results obtained with k values in $[1, 10]$. Similar to Table 1, the values of λ are optimised on the train splits of the 2-fold setup. The general trend observed in all the configurations of the retrieved QVs is that QPP effectiveness decreases with the number of QVs k . This is expected because as k increases, QVs with smaller RBO values will be used for QPP. The low-RBO QVs are more likely than the higher-ranked QVs to include information, which is unrelated to the intent of the input query, and hence degrading the QPP effectiveness.

Effect of varying both k and λ in QPP based on retrieved QVs. Figure 4 shows the joint effect of both the parameters k and λ on the retrieved-QV-based QPP models. To analyse k and λ separately, these results are not the average of a 2-fold setup with hyper-parameter tuning; instead, they show the QPP effectiveness of each combination of k and λ ($k \in \{1, 2, 3\}$, $\lambda \in \{0, 0.1, \dots, 0.9, 1.0\}$).

First, as a general trend, comparing the proposed methods with QV-W2V and QV-RLM in Figure 4, that QPP approaches with retrieved QVs are mostly better than the generated-QV-based QPP methods. The only exception appears in TREC DL’19, where the QPP with the top-3 RLM-generated QVs produces better results than the proposed QPP methods. In all other cases, the proposed methods perform better. Moreover, the QPP methods with 2-hop QVs are consistently better than the ones with 1-hop QVs, thus demonstrating the usefulness of the relevant documents in the training set.

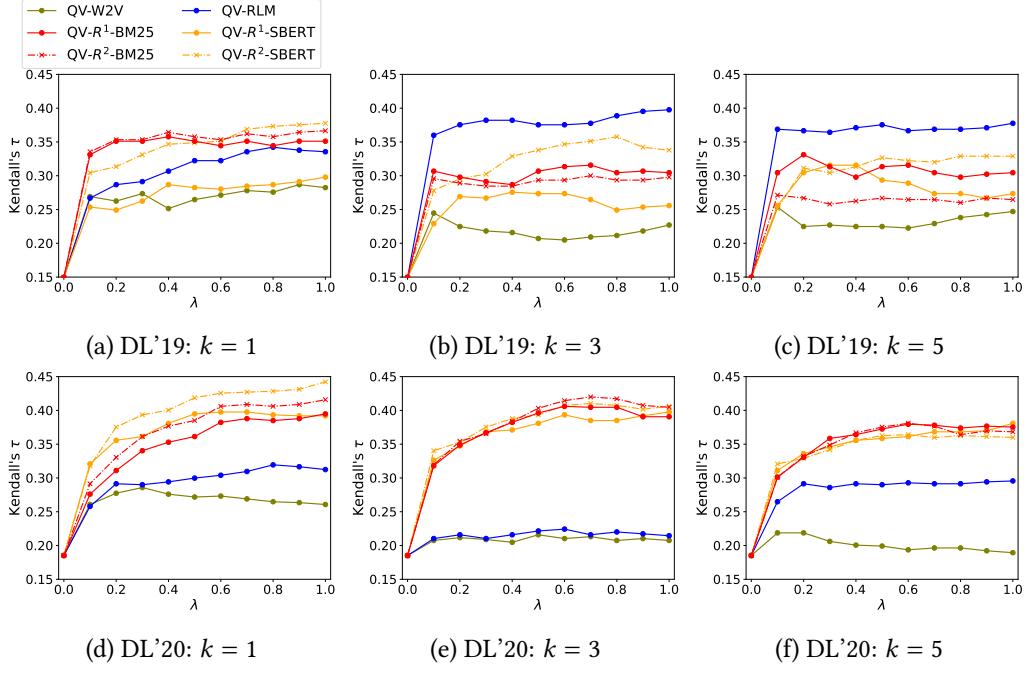


Figure 4: Effect of jointly varying both k (#QVs), and λ (relative importance of the QVs) on DL'19 and DL'20 test queries. The base estimator is NQC, and the target IR metric is AP@100.

To see what values of k work well, we can compare the plots in Figure 4 horizontally (e.g., a->b->c). For both DL'19 and DL'20 queries, we see that small k s work the best and $k = 1$ yields the optimal results. However, there is a considerable difference in the optimal value of λ in DL'19 and DL'20, which can be seen from the fact that the best values for DL'19 usually appear when λ is close to 0.3, whereas for DL'20, the highest values are concentrated on the right half of the λ -axis, where $\lambda > 0.5$.

5. Concluding Remarks

Different from existing QV-based QPP methods, which use manually formulated QVs or generated QVs based on relevance feedback or non-contextual embeddings, we explore the possibility of retrieving QVs within the 2-hop neighbourhood of the target query from a training query set. We propose to incorporate the retrieved QVs in the QPP for the target query and verify that our proposed methods are more effective than existing QV-based QPP approaches for neural rankers, achieving about 20% increase in AP- τ relative to the best-performing baseline.

A likely reason for this improvement is that the proposed 2-hop QV retrieval can effectively retrieve the queries which are most similar to the target query in terms of the information need from a training set. For this reason, our work provides an appealing option for improving QPP effectiveness in real-life scenarios where a query log with user interaction history is available. However, we notice that not every target query can find its query variants from a training set with a fixed number of queries. Therefore, in future, we would like to explore the possibility of integrating QV retrieval with LLM-based QV generation in order to obtain QVs with small topical drifts for every target query.

References

- [1] A. Shtok, O. Kurland, D. Carmel, F. Raiber, G. Markovits, Predicting query performance by query-drift estimation, *ACM Trans. Inf. Syst.* 30 (2012).
- [2] S. Datta, D. Ganguly, M. Mitra, D. Greene, A relative information gain-based query performance prediction framework with generated query variants, *ACM Trans. Inf. Syst.* 41 (2022).

- [3] A. Shtok, O. Kurland, D. Carmel, Query performance prediction using reference lists, *ACM Trans. Inf. Syst.* 34 (2016).
- [4] O. Zendel, A. Shtok, F. Raiber, O. Kurland, J. S. Culpepper, Information needs, queries, and query performance prediction, in: *proc. of SIGIR'19*, 2019, p. 395–404.
- [5] M. Gospodinov, S. MacAvaney, C. Macdonald, Doc2query–: When less is more, in: *Advances in Information Retrieval*, Springer Nature Switzerland, Cham, 2023, pp. 414–422.
- [6] V. Lavrenko, W. B. Croft, Relevance based language models, in: *proc. of SIGIR'01*, 2001, p. 120–127.
- [7] W. Webber, A. Moffat, J. Zobel, A similarity measure for indefinite rankings, *ACM Trans. Inf. Syst.* 28 (2010).
- [8] N. Arabzadeh, F. Zarrinkalam, J. Jovanovic, F. Al-Obeidat, E. Bagheri, Neural embedding-based specificity metrics for pre-retrieval query performance prediction, *IPM* 57 (2020) 102248.
- [9] D. Roy, D. Ganguly, M. Mitra, G. J. Jones, Estimating Gaussian Mixture Models in the local neighbourhood of Embedded Word Vectors for Query Performance Prediction, *IPM* 56 (2019) 1026–1045.
- [10] S. Cronen-Townsend, Y. Zhou, W. B. Croft, Predicting query performance, in: *proc. of SIGIR'02*, 2002, p. 299–306.
- [11] R. Cummins, J. Jose, C. O’Riordan, Improved query performance prediction using standard deviation, in: *proc. of SIGIR'11*, Association for Computing Machinery, 2011, p. 1089–1090.
- [12] A. Shtok, O. Kurland, D. Carmel, Using statistical decision theory and relevance models for query-performance prediction, in: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, Association for Computing Machinery, 2010, p. 259–266.
- [13] G. K. Jayasinghe, W. Webber, M. Sanderson, L. S. Dharmasena, J. S. Culpepper, Statistical comparisons of non-deterministic ir systems using two dimensional variance, *IPM* 51 (2015) 677–694.
- [14] N. A. Jaleel, J. Allan, W. B. Croft, F. Diaz, L. S. Larkey, X. Li, M. D. Smucker, C. Wade, Umass at TREC 2004: Novelty and HARD, in: *TREC 2004*, 2004, pp. 1–13.
- [15] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: *1st International Conference on Learning Representations*, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings, 2013, pp. 1–12.
- [16] S. Ebrahimi, M. Khodabakhsh, N. Arabzadeh, E. Bagheri, Estimating query performance through rich contextualized query representations, in: *Advances in Information Retrieval*, Springer Nature Switzerland, Cham, 2024, pp. 49–58.
- [17] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: *proc. of EMNLP-IJCNLP*, 2019, pp. 3982–3992.
- [18] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, et al., MS MARCO: A human generated machine reading comprehension dataset, *arXiv preprint arXiv:1611.09268* (2016).
- [19] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, E. Voorhees, Overview of the TREC 2019 Deep Learning track, *CoRR abs/2003.07820* (2020). [arXiv:2003.07820](https://arxiv.org/abs/2003.07820).
- [20] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, Overview of the TREC 2020 Deep Learning track, *CoRR abs/2003.07820* (2021). [arXiv:2102.07662](https://arxiv.org/abs/2102.07662).
- [21] R. F. Nogueira, Z. Jiang, R. Pradeep, J. Lin, Document ranking with a pretrained sequence-to-sequence model, in: *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online Event, 16–20 November 2020, volume EMNLP 2020 of *Findings of ACL*, Association for Computational Linguistics, 2020, pp. 708–718.
- [22] C. Li, A. Yates, S. MacAvaney, B. He, Y. Sun, PARADE: Passage Representation Aggregation for Document Reranking, *ACM Trans. Inf. Syst.* 42 (2023).
- [23] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: *proc. of NAACL-HLT'19*, 2019, pp. 4171–4186.
- [24] S.-C. Lin, J.-H. Yang, J. Lin, In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval, in: *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, 2021, pp. 163–173.
- [25] O. Khattab, M. Zaharia, ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT, in: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 2020, p. 39–48.
- [26] D. Ganguly, S. Datta, M. Mitra, D. Greene, An analysis of variations in the effectiveness of query performance prediction, *CoRR abs/2202.06306* (2022).
- [27] N. Arabzadeh, M. Khodabakhsh, E. Bagheri, BERT-QPP: contextualized pre-trained transformers for query performance prediction, in: *CIKM*, ACM, 2021, pp. 2857–2861.
- [28] C. Macdonald, N. Tonellotto, S. MacAvaney, I. Ounis, PyTerrier: Declarative experimentation in python from bm25 to dense retrieval, in: *proc. of CIKM '21*, 2021, p. 4526–4533.