Projection-Displacement based Query Performance Prediction for Embedded Space of Dense Retrievers

SUCHANA DATTA, University College Dublin, Ireland
GUGLIELMO FAGGIOLI, University of Padua, Italy
NICOLA FERRO, University of Padua, Italy
DEBASIS GANGULY, University of Glasgow, UK
CRISTINA IOANA MUNTEAN, ISTI-CNR, Italy
RAFFAELE PEREGO, ISTI-CNR, Italy
NICOLA TONELLOTTO, University of Pisa, Italy

Recent advances in representation learning allow neural Information Retrieval (IR) systems to use learned dense representations for queries and documents to effectively handle semantics, language nuances, and vocabulary mismatch problems. In contrast to traditional IR systems that rely on word matching, dense IR models exploit query/document similarities in dense latent spaces but need substantial training data and come with increased computational demands. Thus, it would be beneficial to predict how a system will perform for a given query to decide whether a dense IR model is the best option or alternatives should be used. Traditional Query Performance Predictors (QPP) are designed for lexical IR approaches and hence they perform sub-optimally when applied to (dense) neural IR systems. Therefore, there has been a renewed interest in QPP to make it more effective for (dense) neural IR models. While the results of the new QPP methods are generally encouraging, there is ample room for improvement in terms of absolute performance and stability. We argue that by using features that are more aligned with the inner rationale underneath dense IR models, we can improve the performance of QPP. In this respect, we propose the Projection-Displacement based QPP (PDQPP) that, exploiting the geometric properties of dense IR models, projects queries and retrieved documents onto sub-spaces defined by pseudo-relevant documents and considers the changes in retrieval scores in such sub-spaces as proxy for retrieval incoherence. Minor score changes suggest coherent retrieval, while significant alterations indicate semantic divergence and potentially poor performance. Results over a wide range of experiment settings on both traditional (TREC Robust) and neural-oriented (TREC Deep Learning) test collections show that PDQPP mostly outperforms the state-of-the-art QPP baselines.

ACM Reference Format:
Suchana Datta, Guglielmo Faggioli, Nicola Ferro, Debasis Ganguly, Cristina Ioana Muntean, Raffaele Perego, and Nicola Tonellotto. 2024. Projection-Displacement based Query Performance Prediction for Embedded Space of Dense Retrievers. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 29 pages. https://doi.org/10.1145/ZZZZZZ.ZZZZZZZ

1 INTRODUCTION

The advent of pretrained Large Language Models (LLMs) has accelerated the development of supervised Information Retrieval (IR) models that use them as foundation models, the parameters of which are then fine-tuned on examples of relevant and non-relevant documents for queries [37, 39, 40, 43, 71, 76]. The parameters of a fine-tuned bi-encoder

Manuscript submitted to ACM

model produce dense vector representations (embeddings) of documents and queries [43, 53]. A bi-encoder model encodes the query and the documents separately in the latent embedding space. This has the advantage that it is possible to embed documents beforehand and only compute the query representation at run-time. Conversely, cross-encoders embed queries and documents jointly, thus requiring re-indexing of the documents and the query at runtime. Dense end-to-end IR models operate by conducting an approximate nearest neighbour search on an indexed embedding space of document and query vectors [37, 39, 40, 53, 71, 76].

While dense representations are more effective in bridging the semantic gap between queries and documents, they are also more computationally expensive. Recognizing which queries benefit from using dense models and which ones can be managed with traditional lexical approaches would allow us to reduce query latency and save computational resources [17, 45]. A second major drawback of dense IR models is the need for vast training data. In particular, if the training set does not contain enough examples for a specific query type, we might observe low performance for such queries. In this sense, understanding which queries are likely to fail may help collect aimed annotations to improve the performance of the dense models on such queries [10, 31]. Consequently, developing effective Query Performance Prediction (QPP) approaches can potentially help design adaptive pipelines for IR systems, where only a subset of queries on which lexical models do not perform well may be routed to more computationally expensive rankers [17, 45] (e.g., dense end-to-end models). Furthermore, QPP estimates may also be used to select queries for deeper relevance assessments to help develop more effective rankers [32].

Most classical QPP approaches leverage discrete term statistics and operate on sparse retrieval pipelines [61, 63, 79]. Off-the-shelf application of these classical QPP approaches on neural ranking models (NRMs) has been shown not to produce sufficiently effective results mainly because these approaches do not factor in term semantics [18, 25, 26].

This paper focuses on improving the QPP effectiveness for end-to-end dense rankers. It has been recently shown that the use of query variants (i.e., alternative formulations of the information need of a query) plays an important role in improving QPP effectiveness [18, 75], mainly because the ranked list of documents retrieved with these variants provide additional sources of information about the retrieval quality of the original top-retrieved list itself. Existing works on generating query variants operate in the discrete term space, e.g., reformulating a query 'five stages of grief' to a more specific version 'five stages of grief in sports' by adding terms. While such variants can be used for QPP estimates via methods such as [18, 75], the variant generation process does not take into account the topology of the embedded space itself. [Comment: R1.1] In this paper, we address the problem from a different angle, experimenting with solutions that operate considering perturbations directly in the embedding space, in line with [3]. We show how this allows us to devise an effective QPP approach for dense NRMs.

The main idea of our method is conceptually similar to aggregating the relative changes in QPP estimates measured across the variants [18]. However, instead of generating variants in the discrete term space and embedding them as dense vectors, we measure these relative changes across the embedded vector representations of the top-retrieved documents. In other words, a top-retrieved document in our proposed method acts as a proxy for a query variant vector. Specifically, our proposed QPP estimator Projection Displacement Query Performance Predictor (PDQPP), projects both the query and the retrieved documents on the subspaces defined by a set of pivot vectors constituted of top-k ranked documents. Our method then aggregates the relative changes in the similarities between the projected vectors and the original ones for each query document pair.

Indeed, the pseudo-relevant documents provide us with an unsupervised way to describe different facets of the topic underlying a query. Suppose there are no major changes in the retrieval scores when we project the query and documents to the subspaces identified by each pseudo-relevant document. In that case, we can hypothesize the retrieval

is of good quality. In contrast, significant changes in retrieval scores suggest that different pseudo-relevant documents define semantically quite different spaces, which, in turn, indicates a possibly incoherent retrieval, potentially indicating low performance.

Our research question can be formalized as follows: *can we employ the projection displacement to exploit topological properties of the latent embedding space of a dense IR model, to devise a query performance predictor that achieves state-of-the-art effectiveness?*

To answer this question, we conducted extensive experimentation, relying on both traditional experimental collections (TREC Robust) and neural-oriented ones (TREC Deep Learning), considering several dense IR retrieval models (ANCE, Contriever, TAS-B, [Comment: 1.6] and MiniLM-l12) and a range of state-of-the-art QPP approaches. Our experiments show that our proposed predictor – PDQPP – is often the top-performing approach or, at least, in the top-performing group. Moreover, it delivers very stable performance across experimental collections and IR models, different from current state-of-the-art approaches, which suffer from performance variability under various operating conditions.

The paper is organized as follows: Section 2 summarizes the relevant literature; Section 3 introduces the projection displacement QPP (PDQPP); Sections 4 and 5 present the experiment setup and results; Section 6 concludes the paper with future directions.

2 RELATED WORK

Dense IR. Traditionally, IR systems relied on lexical signals, such as the presence of the query terms within the documents. However, the emergence of neural models transformed how we represent and match queries and documents. Dense IR approaches are traditionally divided into three main categories: bi-encoders, cross-encoders, and late-interaction models [77].

Bi-encoders (a.k.a dual-encoders) are models that use two separate (but possibly identical) neural networks to represent documents and queries [77]. In the most typical scenario, a placeholder token, such as the [CLS] token [52], is appended to the text (i.e., the query or the document) and the string is fed to a transformer architecture. The latent representation of the placeholder token is then used to represent the text. To compute the similarity between the query and a document, the inner product between the representation of the two is used. This has the major advantage of allowing to precompute the representation of all documents. At runtime, it is sufficient to compute the query representation and its inner product with the representation of all the documents. In recent years, several such models have been released, e.g., STAR [76], ANCE [71], Contriever [39], TAS-B [37]. In this work, we focus on this category of models as they allow us to represent in the same latent space separately queries and documents. More in detail, we focus on symmetric bi-encoders that use the same neural network to encode queries and documents.

Traditional cross-encoders jointly represent documents and queries [77]. To do so, such models concatenate a placeholder token to the query and the document, obtaining a final string with the format "`[CLS]` ⟨query⟩ `[SEP]` ⟨document⟩", where the special token [SEP] indicates where the query finished and the document begins. The string is fed to a transformer architecture that produces a contextual representation of each token. The representation of the `[CLS]` token is further fed to a fully connected layer that outputs the probability that the query is relevant to the token. The model needs access to the query to obtain the representation mentioned above. This requires computing a new representation for the documents every time a new query is received. Therefore, cross-encoders are mostly used to operate on small sets of documents, such as for reranking.

Late-interaction models, such as ColBERT [40], require computing and storing a contextual representation of each term of the query and documents. For what concerns documents, such representation can be computed beforehand

and stored in an efficient index structure. At runtime, the contextual vector representation of each query term is matched with the most similar document term representation for each document. The QPP proposed in this paper focuses on approaches that employ a single representation for the query or the documents. Conversely, late-interaction models employ multiple vectors (i.e., on for each word) to represent queries and documents. How to adapt PDQPP for late-interaction models is left as a future work.

Dense IR systems produce smaller but denser representations than those produced by the traditional lexical IR approaches. Indeed, classical solutions are based on representations whose dimensionality ranges from tens of thousands to hundreds of thousands of dimensions: the number of terms in the vocabulary considered by the IR. Vice-versa, dense IR systems learn representations whose dimensionality falls within hundreds to thousands.

Traditional QPP. Depending on the features they rely upon, traditional QPPs are divided into pre- and post-retrieval predictors [7, 35, 36]. The former relies on signals that can be derived without considering the ranked list of documents produced in response to the query. Such signals are, for example, the collection frequency of terms appearing in the query [47, 78]. On the other hand, post-retrieval predictors infer their predictions by taking the ranked list of documents in response to the query as input. Depending on which aspects are considered to compute the prediction, there are three main classes of post-retrieval predictors: coherence-, score-, and robustness-based. Coherence-based predictors rely on measuring how strongly documents retrieved are clustered together: the most well-known representative of this class of approaches is Clarity [13]. PDQPP, the predictor proposed in this paper is a member of this class. Score-based predictors employ heuristics computed on the retrieval score of the retrieved documents, some examples include Weighted Information Gain (WIG) [79], Normalized Query Commitment (NQC) [63], and Score Magnitude and Variance (SMV) [65]. Finally, robustness-based predictors compare the original ranking of documents with one produced by introducing noise in the query, the index, or documents, e.g., the Utility Estimation Framework (UEF) [61], the Reference Lists framework [56, 62], and Robust Standard Deviation (RSD) [58].

Traditional QPPs were meant and designed to operate on lexical IR methods, such as BM25 [55] or the probabilistic language models, that relied on the presence of the same terms in both queries and documents to determine the relevance of a document. With the advent of Neural IR and semantic matching-based IR systems, it was highlighted the need for novel QPPs explicitly designed to cooperate with such novel IR systems [26]. In this regard, we recognize two novel classes of QPPs: those that employ semantic signals but are aimed at predicting the performance of lexical IR systems, and those explicitly designed to cooperate with Neural IR models.

Semantic QPPs for lexical IR systems. The advent of word embeddings fostered the development of QPP models that exploit them to compute their predictions. NeuralQPP, proposed by Zamani et al. [74], uses Deep Learning to integrate three diverse signals: the query text, the retrieval scores, and aspects related to the distribution of the terms. On the same line, Roy et al. [59] show that, by utilizing the semantic similarity aspect of word embedding, it is possible to estimate the local neighbourhood of a query using Gaussian Mixture Models. Roy et al. observe that the spatial properties of such a neighbourhood correlate with system performance. Similarly, Arabzadeh et al. [5, 6] propose a set of measures derived from neural embeddings that allow for quantifying the term specificity. They observe that the presence of specific terms in a query suggests more effective retrieval. Khodabakhsh and Bagheri [41] propose three neural features based on dense word representations: Neural Matching, Neural Aggregated Matching, and Neural Distance. These features combine the embeddings of query and document tokens to capture the semantic relationships occurring between them. The authors use the matching signals provided by such features to encode semantic aspects within classic predictors. Differently, Datta et al. [16] proposes using the interaction between query and document

terms as signals for QPP. Specifically, they employ 3D convolutional neural networks with shared parameters to train an end-to-end pairwise predictor, called Deep-QPP.

Arabzadeh et al. [1] introduce BERT-QPP, one of the first methods harnessing LLMs for QPP. Specifically, they fine-tune BERT [20] by utilizing BM25's performance on each training query and the BERT representation of the first retrieved document as supervision to train a QPP. Several subsequent works build upon BERT-QPP. Similarly, Chen et al. [9] extend BERT-QPP, introducing a groupwise approach enabling the query performance prediction using signals from multiple queries simultaneously.

Arabzadeh et al. [4] also utilize LLMs to create a predictor for conversational search. They leverage BERT to construct a document graph and cluster documents. If multiple clusters exist for a document, they identify the user's information need by posing clarifying questions to determine the cluster containing relevant documents. Subsequently, they test this approach using BM25. While these methods lean towards Neural IR models, their primary application remains associated with lexical IR approaches. This leads to a discrepancy between the query/document representations utilized for ranking and prediction phases, with the former relying more on lexical aspects and the latter emphasizing semantic information. [Comment: 1.3] More recently, Saleminezhad et al. [60] explore the role of semantic representations for a pre-retrieval predictor designed for lexical IR systems. More precisely, Saleminezhad et al. proposed a three-step QPP model that predicts whether the query terms are "useful" (i.e., they are likely to improve the performance of the query), or "harmful" (i.e., they cause a performance drop for the query). First, by using T5 [50], they generate several query formulations. Some of them perform better than the original query, and others worse. Then, they assign a label to each term in the generated queries. This label is 1 if the term appears in a query that performs better than the original, -1 if the query performs worse and 0 if the term appears in both the original and generated query. Finally, using contextualized word embeddings and a linear regression model, they learn how to predict automatically this label. By applying this predictor to the query terms, Saleminezhad et al. [60] determine if such terms improve the final performance of the query. Notice that, since the method operates exclusively on the query itself, it represents an example of a pre-retrieval predictor. Saleminezhad et al. [60] test their approach on BM25. [Comment: 1.3] On the same line [21] exploit a fine-tuned transformer to address the task of predicting the performance of a lexical model. In particular, they combine information on the query, the retrieved documents, and historical queries for which the performance is known to predict how the query will perform. On a different line, Khodabakhsh et al. [42] develop a pre-retrieval model based on BERT. In this case, Khodabakhsh et al. [42] first expand the relevance judgement on the training set by employing an expensive model (duoBERT [49]) to construct pseudorelevance labels. Using the newly built labels, they compute a performance metric in training and use it as supervision to fine-tune a BERT model to predict it given the query representation. All the predictors mentioned above are evaluated on IR systems that rely on lexical matching, thus are hindered when used to predict the performance of IR systems that exploit semantic matching [26]. Given that most of these predictors are designed for and tested on lexical IR models, they do not align with the focus of this paper, which instead addresses QPP predictors tailored for dense IR models.

QPP for Neural IR. [Comment: 1.3] The research community has recently invested some effort in devising QPP models specific for Neural IR systems, as observed and testified in the community events, such as workshops [22] and tutorials [2] on the topic. Among QPPs explicitly designed to work the best with neural IR systems, Hashemi et al. [34] introduce Non-Factoid Question Answering QPP (NQAQPP), a methodology incorporating retrieval scores, query lexical features, and both query and answer lexical features within a deep neural network framework for addressing Non-Factoid Question Answering. Hashemi et al. is also one of the early works evaluating the effectiveness of QPP

on neural IR models. They specifically evaluate it on BM25, aNMM [73], and Conv-KNRM [15], noting a substantial gap in predictive accuracy between BM25 and neural IR models, attributed to distinct score distributions generated by neural models. In a recent investigation, Faggioli et al. [26] scrutinize the capability of traditional QPP techniques in predicting the performance of neural IR systems and, through a series of experiments, they find a significant decline in the performance of current QPP models when applied to neural IR systems. This trend persists even when employing BERT-QPP as a predictive model for neural IR. Similarly, Datta et al. [18, 19] observe the diminished effectiveness of prior QPP methods when employed for neural IR compared to lexical IR. In response, they propose Weighted Relative Information Gain-based model (WRIG), a statistical approach employing probabilistic combinations of retrieval scores for multiple query formulations. To demonstrate the efficacy of their approach, they utilize WRIG to predict performance in BM25, four variations of DRMM [33], and the initial stage neural IR model, ColBERT [40]. Singh et al. [64] propose a novel QPP that employs an auxiliary pairwise ranker (DuoT5) as an unsupervised QPP model to measure how often the ranking produced by the IR system agrees with the pairwise comparison of the auxiliary model. Similarly to [18, 19], Singh et al. test the performance of the proposed model on multiple neural IR models, both considered end-to-end retrieval as well as reranking. Faggioli et al. [24] utilize the geometric characteristics of dense representations for performance prediction in conversational search, by devising the Reciprocal Volume (RV) predictor which consists of computing the volume on the axes-aligned bounding box containing the top-k retrieved documents and the query. More recently, Arabzadeh et al. [3] proposed a strategy explicitly designed to be applied for dense IR systems. The predictor proposed by Arabzadeh et al., called DenseQPP (DQPP), is based on measuring the similarity between the original ranked list and the ranked list obtained after perturbing the query with appositely crafted Gaussian noise. Faggioli et al. [23] propose a novel framework, called Dense-Centroid (DC) framework, to adapt traditional predictors to the dense IR systems. They start by noticing that classical predictors require regularizing predictions by the retrieval score that the corpus would achieve in response to the query. This score cannot be computed for dense models, as it would require feeding the entire corpus to the dense IR system and obtaining its representation. Therefore, they propose to use, as a proxy representation of the corpus, the centroid of the documents. More concretely, in their approach, the dot product between the original query and the centroid is used as a regularization factor within the classical QPPs. As these approaches share similar characteristics with that of our proposed approach PDQPP, in Section 3.6, we provide a detailed comparison between PDQPP and the above-mentioned state-of-the-art predictors.

3 PROPOSED METHODOLOGY

In this section, we describe our proposed methodology of projection-based QPP estimation.

3.1 Notations and Core concepts

In this section, we introduce the notations for embedded query and document vectors and outline the concept of vector projection, an essential component of our proposed predictor.

Embedded documents and queries. Since we aim to predict QPP for dense neural ranking models, we introduce the notations that will be useful to understand how our methodology works on the space of embedded vectors obtained via a bi-encoder-based neural representation model [39, 72]. Let ϕ be a bi-encoder-based supervised neural representation model, which has learned the parameterised representations of queries and documents from a training dataset. Embeddings of the textual representation of a query Q and that of a document $D \in \mathcal{D}$ ($\mathcal{D}$ denotes a document corpus) are then denoted, respectively, as $\mathbf{q}$ and $\mathbf{d}$, where both $\mathbf{q}$ and $\mathbf{d} \in \mathbb{R}^p$. The retrieval score of a document for a

query Q is then usually obtained by computing a dot product between the embedded representations of the query and a document from a candidate set, i.e., $\pi(Q, D) \stackrel{\text{def}}{=} \mathbf{q} \cdot \mathbf{d}$, where $D \in \mathcal{D}_k$ denoting a candidate set of k documents obtained via approximate nearest neighbour search on the embedded space.

Vector projections. We now introduce the concept of *projection* and discuss how it plays an important role in our proposed predictor. Informally speaking, a projection of a vector $\mathbf{d}$ onto another vector $\mathbf{v}$ leads to aligning $\mathbf{d}$ in the direction of $\mathbf{v}$ and also changes its magnitude. The standard notation to denote the projected vector is $\mathbf{d_v}$ (read as $\mathbf{d}$ projected onto $\mathbf{v}$), and is defined as

$$\mathbf{d_v} = \left(\frac{\mathbf{d} \cdot \mathbf{v}}{||\mathbf{v}||}\right)\hat{\mathbf{v}}, \tag{1}$$

where $||\mathbf{v}||$ denotes any norm (e.g., L^2) of the vector $\mathbf{v}$, and $\hat{\mathbf{v}}$ denotes the unit vector along $\mathbf{v}$, i.e., $\hat{\mathbf{v}} = \mathbf{v}/||\mathbf{v}||$. Note that the quantity within parenthesis is a scalar, and hence the projected vector $\mathbf{d_v}$ is a scaled version of $\hat{\mathbf{v}}$.

Projection displacement. We now introduce the concept of *projection displacement*, which represents how much the *similarity* between a pair of vectors (in terms of their dot product) or the angular distance between them (in terms of the cosine inverse of their dot product) changes when both are projected onto a different vector. Formally, we define the projection displacement of a pair of vectors $(\mathbf{q}, \mathbf{d})$ given a third vector $\mathbf{v}$ as

$$\delta_{\mathbf{v}}(\mathbf{q}, \mathbf{d}) = \mathbf{q} \cdot \mathbf{d} - \mathbf{q_v} \cdot \mathbf{d_v}, \tag{2}$$

where the notation $\mathbf{x_v}$, as per Equation 1, denotes the projection of $\mathbf{x}$ onto $\mathbf{v}$. The projection displacement of Equation 2 represents the relative gain (or loss) of the estimated similarity between two vectors $\mathbf{q}$ and $\mathbf{d}$ when a different frame of reference ($\mathbf{v}$) is used to estimate this similarity.

3.2 Relative Changes in Retrieval Scores

In this section, we discuss the idea of projection displacement under the specific context of embedded documents and query vectors. Revisiting Equation 2 with an assumption that $\mathbf{q}$ refers to the embedding of a query Q and $\mathbf{d}$ refers to that of a document D, projection displacement can be interpreted as the relative change in the similarity between the query and the document when a different frame of reference is used. In terms of retrieval using an NRM, this affects the relative rank of the document D.

To better understand the characteristics of projction displacement within the specific context of dense retrievers, let us revisit Equation 2 and express it in terms of the angles between the vectors. Substituting the identity $\mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}||||\mathbf{y}|| \cos(\mathbf{x}, \mathbf{y})$ into Equation 2, we see that the dot product between a pair of vectors $\mathbf{q}$ and $\mathbf{d}$ when projected on an arbitrary vector $\mathbf{v}$ can be expressed as

$$\begin{aligned}
\mathbf{q_v} \cdot \mathbf{d_v} &= \frac{||\mathbf{q}||||\mathbf{v}|| \cos(\mathbf{q}, \mathbf{v})}{||\mathbf{v}||}\hat{\mathbf{v}} \cdot \frac{||\mathbf{d}||||\mathbf{v}|| \cos(\mathbf{d}, \mathbf{v})}{||\mathbf{v}||}\hat{\mathbf{v}} \\
&= ||\mathbf{q}|| \cos(\mathbf{q}, \mathbf{v})\hat{\mathbf{v}} \cdot ||\mathbf{d}|| \cos(\mathbf{d}, \mathbf{v})\hat{\mathbf{v}} \\
&= ||\mathbf{q}|| \cos(\mathbf{q}, \mathbf{v})||\mathbf{d}|| \cos(\mathbf{d}, \mathbf{v}) \cos(\hat{\mathbf{v}}, \hat{\mathbf{v}}) \\
&= ||\mathbf{q}||||\mathbf{d}|| \cos(\mathbf{q}, \mathbf{v}) \cos(\mathbf{d}, \mathbf{v}).
\end{aligned} \tag{3}$$

The last step is derived from the fact that both $\mathbf{q_v}$ and $\mathbf{d_v}$ are vectors along the same direction, and hence $\cos(\hat{\mathbf{v}}, \hat{\mathbf{v}}) = 1$.

Equation 3 expresses the similarity between a query and a document vector projected along the same direction as a product of their norms and their angles with the axis of projection, which when substituted into Equation 2 yields the

expression for projection displacement as

$$\begin{aligned}\delta_{\mathbf{v}}(\mathbf{q},\mathbf{d}) &= \mathbf{q}\cdot\mathbf{d} - \mathbf{q_v}\cdot\mathbf{d_v} \\ &= ||\mathbf{q}||||\mathbf{d}||\cos(\mathbf{q},\mathbf{d}) - ||\mathbf{q}||||\mathbf{d}||\cos(\mathbf{q},\mathbf{v})\cos(\mathbf{d},\mathbf{v}) \\ &= ||\mathbf{q}||||\mathbf{d}||\Big(\cos(\mathbf{q},\mathbf{d}) - \cos(\mathbf{q},\mathbf{v})\cos(\mathbf{d},\mathbf{v})\Big).\end{aligned} \tag{4}$$

The formulation of projection displacement in Equation 4 allows relating it to QPP estimation. As a boundary case realise that $\delta_{\mathbf{v}}(\mathbf{q},\mathbf{d}) = 0$ if $\mathbf{v} = \mathbf{q}$ or $\mathbf{v} = \mathbf{d}$, e.g., if $\mathbf{v} = \mathbf{q}$ then $\cos(\mathbf{q},\mathbf{v}) = 1$ and $\cos(\mathbf{q},\mathbf{v}) = \cos(\mathbf{q},\mathbf{d})$, as a result $\delta_{\mathbf{v}}(\mathbf{q},\mathbf{d}) = ||\mathbf{q}||||\mathbf{d}||(\cos(\mathbf{q},\mathbf{d}) - \cos(\mathbf{q},\mathbf{d})) = 0$.

By a similar argument, if the projection axis $\mathbf{v}$ is close to either the query or the document, i.e., $|1 - \cos(\mathbf{q},\mathbf{v})| < \epsilon$ for a sufficiently small $\epsilon \in \mathbb{R}^+$, it is easy to see that $\delta_{\mathbf{v}}(\mathbf{q},\mathbf{d}) \to 0$. In other words, projection axes $\mathbf{v}$ close to either the query or the document induces small projection displacements.

3.3 Choosing the Projection Vectors

Till now, we have defined the projection displacement (Equations 2 and 4) in a generic way for an arbitrary vector $\mathbf{v}$. We now consider the situation when this vector $\mathbf{v}$ corresponds to an alternative formulation of the same information need as expressed by the embedding $\mathbf{q}$ of a query Q. In such a case, $\mathbf{q_v}\cdot\mathbf{d_v}$ can be interpreted as the similarity between the query and a document D (embedded as $\mathbf{d}$) in this transformed space of an alternative representation of the information need.

According to the *Clusering Hypothesis* [67], if a document D is relevant to the query Q, then we expect their representation to be similar. Furthermore, assume V represents a piece of information highly related to Q, such as a reformulation or the response to Q. If D is relevant to a query Q, it is also likely to be relevant (and hence likely to yield a high similarity score) to a query variant V [8, 18, 75].

In the context of QPP, this means that for a query and a relevant document pair (Q, D), the projection displacements or the relative changes in the retrieval scores for a different way of expressing the information need (i.e., V) should be small. This is the key idea of our proposed QPP estimator which measures the relative stability of the retrieval scores of top-retrieved documents along different projection vectors. [Comment: 3.1] Explicitly, our research hypothesis is that, assuming we were able to identify the different – latent – ways of representing an information need, we could use this information to estimate the expected performance of the IR system. In particular, small changes in the ranked list across such latent representations of the information need suggest it is stable with uniform latent ways of expressing it and thus stable retrieval, while major perturbations indicate a highly faceted information need for which the retrieval was less satisfactory.

While previous QPP approaches, such as [18, 75], have leveraged manually created or automatically generated query variants for discrete text, it is inconvenient to generate such variants in the embedded space of vectors. For QPP on dense vectors, we propose to make use of the top-retrieved documents themselves as the different axes for computing the projection displacements.

[Comment: 2.8] To mimic this behaviour, instead of using query variations, we propose to induce perturbations in the retrieval space. More in detail, we use the documents as "pivot documents" to change the space where retrieval occurs. More specifically, we take the pivot document, we project the query and the other documents on it using the projection operator defined above (Eq. 1), and we measure the displacement (Eq. 2) that occurs in the new projection space induced by the pivot document. This document is called a "pivot" since it modifies the retrieval space while

remaining fixed. Therefore, we now define more formally the fundamental component of our proposed QPP predictor, that uses the *projection displacement deviation* (PDD) for a pivot document (say D) over a set of top-ranked k documents. Formally, given a query embedding $\mathbf{q}$, a set of top-retrieved documents $\mathcal{D}_k$ for the query and the embedding $\mathbf{d}$ of a pivot document $D \in \mathcal{D}_k$, we define PDD as the standard deviation of the projection displacement values for each top-ranked document when projected along the pivot, i.e.,

$$\text{PDD}(\mathbf{q}, \mathbf{d}, \mathcal{D}_k) = \sqrt{\frac{\sum_{i=1}^{k} (\delta_{\mathbf{d}}(\mathbf{q}, \mathbf{d_i}) - \mu)^2}{k}}, \text{ where } \mu = \frac{\sum_{j=1}^{k} \delta_{\mathbf{d}}(\mathbf{q}, \mathbf{d_j})}{k}. \tag{5}$$

Intuitively, we expect the function $\text{PDD}(\mathbf{q}, \mathbf{d}, \mathcal{D}_k)$ to yield a small value if the pivot document is topically aligned with the query and every other document in the top-retrieved set. This is likely to happen if the pivot document is relevant to the query.

For under-specified queries, but also in the case of unsuccessful retrieval, the top-retrieved set of documents likely corresponds to different aspects of information need. In such a situation, selecting a pivot document that corresponds to a particular aspect of information need may lead to larger PDD values due to the presence of other documents corresponding to a different aspect. This also means that PDD concerning a pivot top-ranked document (Equation 5) can potentially act as a component to define an effective query performance estimator for dense vector spaces of queries and documents because a small value of this quantity is indicative of a likely well-specified query and vice-versa.

3.4 PDD-based QPP predictor

With the PDD definition (Section 3.3) and its geometric illustration (Section 3.5) we now formulate the predictor in terms of the PDD values. The key idea behind the proposed predictor is to aggregate the evidence for PDD values along several top-retrieved documents, which is similar to the idea of aggregating QPP estimates over multiple query variants [18, 75].

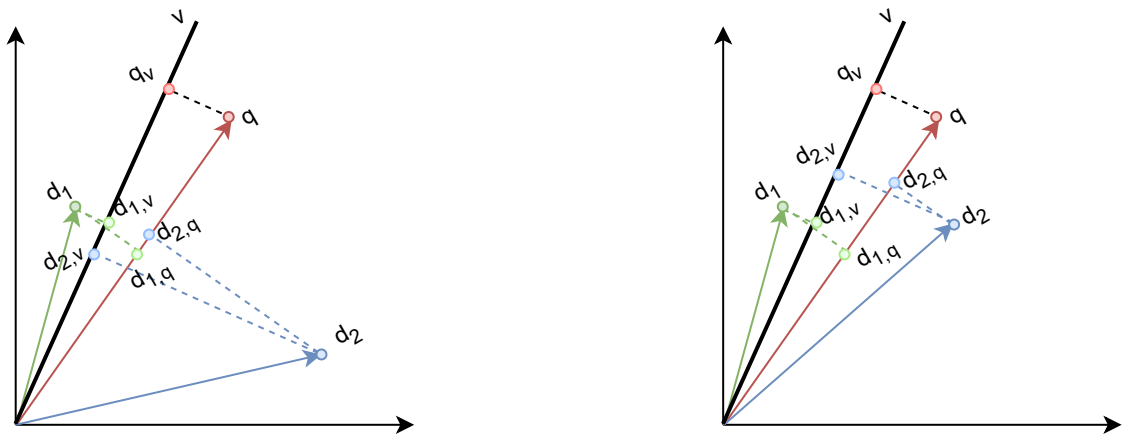
While the PDD values indicate the standard deviation of the topical alignment of the top-retrieved documents, it is potentially useful to scale these values relative to the similarities between the query and the document vectors in the embedded space, i.e., the retrieval scores. This scaling is likely to help calibrate these values over a range of different queries and potentially leads to an effective comparison between the QPP estimates.

Since our predictor, which we call **PDQPP**, aggregates the scaled PDD values over multiple pivots, we introduce an additional parameter to allow provision for how many documents to consider for this aggregation. Formally speaking, we call $\bar{l}$ the mean of the retrieval scores of the top-l retrieved document, i.e., $\bar{l} = \frac{1}{l} \sum_{D \in \mathcal{D}_l} \mathbf{q} \cdot \mathbf{d}$. Then, PDQPP is defined as:

$$\text{PDQPP}(Q) = -\frac{\sum_{j=1}^{k} \text{PDD}(\mathbf{q}, \mathbf{d}_j, \mathcal{D}_h)}{k \cdot \sqrt{\frac{1}{l} \sum_{i=1}^{l} (\mathbf{q} \cdot \mathbf{d}_i - \bar{l})^2}}, \tag{6}$$

where the numerator represents the PDD values (Equation 5) computed for k pivots over a set of h top-ranked documents (h and k being two different parameters), whereas the denominator corresponds to the scaling factor of average similarity values between the query and a set of top-l ranked documents (again the parameter l is different from k and h).

The predictor is an additive inverse of these aggregated displacement values (minus sign at the front of Equation 6) because the higher the displacements the higher is the likelihood that the query itself is under-specified and the top documents potentially correspond to different aspects of information need, some of which could be non-relevant thus degrading the retrieval effectiveness of such queries.



(a) An under-specified query.

(b) A well-specified query.

Fig. 1. A 2D visualisation of the local geometry of the embedding spaces of two queries and their top-retrieved documents. *a)* It is easy to find a pivot vector for which some of the document embeddings will be well-aligned whereas others will not. This will lead to a large PDD value (Equation 5). *b)* document vectors are well-aligned to the pivot vector, which means that the PDD values will be smaller. While v can be any possible direction, we observe empirically that the best results are achieved when w is aligned with pseudo-relevant documents vectors

The generic form of our proposed predictor has three hyper-parameters to control the sizes of the top-retrieved sets for three different computation purposes - i) a top-set of h documents to compute the PDD values with respect to a particular pivot document (Equation 5), ii) k, which specifies how many pivot documents to consider for aggregating the PDD values, and iii) l, the number of documents considered to compute the standard deviation of the retrieval scores.

3.5 A Geometric Illustration

Dense vector representations of the top-retrieved documents addressing different aspects of the information need are likely to be aligned along different subspaces while all of these are still similar to the query subspace. However, this means that a document addressing a specific aspect of the information need is likely to be dissimilar to another on a different aspect.

We provide here an illustrative example using a query that contains the polysemous word "bank", such as "Where is the closest bank?". The word "bank" might refer to the "financial institution" or "the land alongside a river", among many other meanings. Therefore, in response to the query, the retrieval system has retrieved documents concerning both financial institutions and geographic structures. Assume now we can somehow disambiguate the meanings of bank by transforming the projection space. If we could move to the "financial" projection space, we would observe documents concerning financial institutions to be close to the query, as in this new space the word "bank" refers to the financial institution, while documents regarding river sides would be demoted. Vice-versa, if we were to move on the "geography" projection space, we would observe documents concerning river banks to be close to the query. Depending on which meaning we attribute to the query, the ranking of documents dramatically changes. This is a clear indication

of a complex query for which the IR system is likely to fail – not even a human being would be able to answer the query "Where is the closest bank?" without asking further questions!

Consider now a more specific query, such as "Where is the closest financial institution?". In this case, we could assume that our IR system will retrieve almost exclusively documents where the word bank refers to the financial meaning. Thus, regardless of the space we consider, we will observe the documents to be close to the query.

Our example assumes we are capable of doing two types of geometric operations: i) we are able to define subspaces that represent the different semantic meanings of the query ii) we can change our projection space to reflect different semantic aspects. The second operation is handled using the projection operator defined in Eq. 1.

Conversely, to address the first operation, Equation 6 employs the first top k documents as pivot documents. We assume that each of these documents conveys a specific semantic meaning and defines a subspace characterized by a latent semantic. These subspaces might be very similar if the pivot documents have similar semantics (e.g., they refer to closely related topics), or might be very different if different documents refer to completely unrelated subjects. Considering our example again, when it comes to the query "Where is the closest bank", the top-k documents could for example focus on different meanings of the word bank. Therefore, depending on which document is used as the pivot, we will observe differences in ranking when things are projected onto such pivot. This hints at a weak retrieval. Vice-versa, when we consider the second query, "Where is the closest financial institution?", if the top-k retrieved documents have similar meanings, then, when using each of them as a pivot document, we will observe a relatively stable ranking.

Figure 1 visualises the idea in two dimensions. Figure 1b shows the embeddings of the top-retrieved documents for an under-specified query, where the angles between the top-retrieved documents can be large if they represent different topics, whereas, for a well-specified query (Figure 1a), it is likely that all the top-retrieved documents are likely to be similar to each other (and also to the query).

In terms of the project displacement deviation (as defined in Equation 5), choosing any direction as the pivot document for computing PDD values over the embeddings of Figure 1b is likely to lead to a large value because there potentially will be documents that are not well aligned with the pivot direction $\mathbf{v}$. On the other hand, the PDD values for the embeddings in Figure 1a are likely to be small because each top-retrieved document will potentially be aligned well with any pivot vector.

Considering the bank example, Figure 1a could represent the situation where the query is "Where is the closest bank?". In line with our example, d_1 is a document about financial institutions, while d_2 regards river banks. If our pivot document v concerns financial aspects, then when we project the query and the documents on the subspace defined by v, we observe d_1 being closer to the query in the subspace than d_2 – i.e., when projected on v, d_1 is closer to q than d_2. This is the opposite of what happens when considering the default situation (i.e., d_1 and d_2 projected on the query). Thus, our change or reference space induces a switch between d_1 and d_2 in the ranking. Vice-versa, Figure 1b represents the scenario where all the retrieved documents are closely related. Then, when we observe the projection on v of d_1 and d_2, we do not notice any switch in their ranking.

3.6 PDQPP vs. other existing predictors

After presenting our predictor, we now discuss how PDQPP differs from existing predictors, while still resembling them in certain ways. This will be useful to see how PDQPP generalises some predictors seeking to mitigate their limitations.

PDQPP vs. Score Standard Deviation (SD)-based predictors. Several classical predictors [51, 57, 63], as well as DC predictors [25], employ the retrieval score standard deviation to produce predictions. The rationale is that a high variance indicates that the IR system scored much higher on the documents retrieved in the top positions within a ranked list as compared to lower positions. This utilises the hypothesis that such high scores are reflective of the relevance of the top-ranked documents. Our predictor PDQPP uses the same signal (denominator of Equation 6) with a different underlying objective - which is to normalise the projection displacement values to produce its predictions.

The major improvement obtained by our predictor PDQPP over score standard deviation-based ones (results later in Section 5) can most likely be attributed to the additional factor incorporated as the projection displacement deviation. While an existing score standard deviation-based predictor can only compute how topically distinct a set of a very-top list of documents is from the ones that follow it, such predictors cannot predict the topical coherence of the top-retrieved set - a coherent set potentially indicating better quality retrieval.

PDQPP vs. the UEF estimator. The UEF framework for QPP estimation [61] relies on using pseudo-relevant documents to expand a query, retrieve a new set of documents, and compare the original ranked list with the one obtained from the expanded query. Conceptually, the UEF framework and PDQPP share several common characteristics. The UEF framework, in a sense, transforms a query into the reference space induced by the pseudo-relevant documents and then estimates how this transformed representation affects the ranking of the documents. Our proposed PDQPP operates in a similar but more explicit manner in that the query (and the documents) are explicitly projected within the pseudo-relevant space. Furthermore, similar to UEF, the projection displacement measures the (dis-)similarity between the results of the query in the original space as against the ones induced by the pseudo-relevant documents (Equation 5). The major difference is that by explicitly relying on the geometrical representation of the various elements, PDQPP better suits the end-to-end dense IR models.

PDQPP and DQPP. DQPP [3] projects the top-ranked documents on a subspace obtained by a perturbed version of the query, and then computes the robustness of the ranking of documents relative to this change. It can therefore be argued that both DQPP and PDQPP models project information on a different space, and hence estimate the robustness of an IR model relative to this transformed representation. While DQPP obtains this directly by comparing the two retrieved lists, PDQPP on the other hand, achieves this using the projection displacement operator. Moreover, PDQPP offers an advantage over DQPP in the sense that the new subspace where documents and queries are projected is not random. Instead, this reference subspace aligns with the pseudo-relevant documents, thus allowing provision to leverage the latent semantics of these documents for query performance estimation.

PDQPP and WRIG. WRIG computes the relative changes in the QPP estimates with reference to a set of query variants with the idea that a large increase potentially indicates that the original query itself was under-specified (poor retrieval quality), whereas a large decrease suggests that the original query itself was well-specified (effective retrieval quality) [18]. The idea of transforming a query via projection onto a reference subspace relates to that of leveraging information from variants in WRIG. While WRIG uses the relative gains computed via a base QPP estimator, our predictor PDQPP, instead, uses deviations of projection displacements.

Table 1. Evaluation (nDCG@10) of the dense IR models on the respective test collections subsequently used for our QPP experiments.

Topic set	ANCE	Contriever	TAS-B	MiniLM-l12
DL '19	0.645	0.676	0.716	0.673
DL '20	0.646	0.671	0.684	0.684
DL Hard	0.328	0.376	0.376	0.344
Robust '04	0.362	0.499	0.453	0.386

4 EXPERIMENT SETTINGS

4.1 Datasets and Models

Dense Neural Models. In our experimental analysis, we consider three dense retrieval models: ANCE[1] [71], Contriever[2] [39], and TAS-B[3] [37]. We use the model weights fine-tuned on the MS MARCO collection and publicly available on the huggingface repository. All the models that we experimented with use 768 dimensional embeddings for documents and queries.

Dataset. As benchmark datasets, we employ the following four collections: TREC Deep Learning '19 (DL '19) [12], TREC Deep Learning '20 (DL '20) [11], Deep Learning Hard (DL Hard) [44], and TREC Robust '04 (Robust '04) [68]. DL '19, DL '20, and DL Hard datasets constitute 43, 54, and 50 queries, respectively, with depth pooled relevance assessments (depth 10). The underlying task is ad-hoc passage retrieval on MS MARCO corpus, which contains over 8M passages [48]. As a part of the experiment setup, all the dense IR systems were fine-tuned on the MS MARCO training set of pairs of queries and relevant passages. The respective topic sets of DL '19, DL '20 and DL Hard, the predictions are in-domain in nature.

Additionally, to evaluate the QPP effectiveness for the neural models for out-domain ranking predictions, we employ Robust '04, constituted of disks 4 and 5 (minus congressional records) of the Tipster collection. The Robust '04 collection uses a deeper pool (depth 100) for relevance assessments, as a result of which recall plays a crucial role in determining a query's performance. It thus offers a different evaluation setting as compared to MS MARCO passage collection.

[Comment: 2.7] To favor reproducibility, the code (including the baselines) as well as the data (runs and prediction scores) are publicly available on GitHub [4].

4.2 Baselines and Evaluation Measures

Since our proposed QPP estimator is an unsupervised approach, we employ a wide range of existing unsupervised predictors as baselines for a fair comparison. More specifically, we consider two different categories of QPP models - i) those that are agnostic of an IR model, and ii) those that are explicitly designed to operate on embedding spaces of dense IR models.

IR Model agnostic QPP approaches. As QPP baselines that can work on both sparse and dense retrievers (i.e., agnostic QPP) we employ the following:

[1]https://huggingface.co/sentence-transformers/msmarco-roberta-base-ance-firstp
[2]https://huggingface.co/facebook/contriever
[3]https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b
[4]Link to be released upon acceptance

- **SD** [51] is an approach that predicts the performance using the standard deviation of the retrieval scores of the first top-k retrieved.
- [Comment: 1.2] $n(\sigma_{\%})$ [14] that considers as prediction score the standard deviation of the retrieval scores of the documents whose retrieval score is at least $n\%$ of the score of the highest retrieval score.
- **Clarity** [13] computes the Kullback–Leibler (KL) divergence between the language model of the entire corpus and the one of the top-k retrieved documents. Clarity operates under the assumption that observing a large KL divergence indicates a well-characterized and coherent set of top-k documents, which hints at a good retrieval.
- **NQC** [63] is the standard deviation of the retrieval scores of the first top-k retrieved documents, regularized by the retrieval score of the entire corpus.
- **RSD** [58] iterates over the retrieved list of documents, computing at each position the unbiased standard deviation of the scores reweighed by the WIG score of the ranking list up to that position and sums all these values.
- **SMV** [65] Combines NQC and WIG by taking into consideration both the magnitude and the variance of the retrieval scores of the top-k documents.
- **WIG** [79] is the average retrieval score of the first top-k retrieved documents, regularized by the retrieval score of the entire corpus.
- **UEF Framework** [61] The UEF framework operates by reweighing any of the aforementioned predictors (Clarity, NQC, SMV and WIG) by the similarity between the original retrieved list of documents and the list of documents retrieved after rewriting the query via Pseudo-Relevance Feedback (PRF).
- **WRIG** [18] is a variant-based predictor that computes the changes in the QPP estimates as obtained from a base predictor on a set of query variants relative to the original query. As suggested in [18], we employed NQC as the baseline predictor of WRIG. Additionally, we worked with a set of query variants automatically generated by skipgram embeddings [46] as suggested in [18]. Notice that WRIG was observed to supersed reference lists based methods [75].

Dense IR-based approaches. This class of QPP models are explicitly formulated to operate with dense IR models. As baselines we use the following:

- **DC** framework [25] instantiates traditional predictors (e.g., WIG [79], NQC [63], SMV [65]) by considering the centroid of all documents as an approximated corpus representation. Among the DC class of predictors, we consider DCWIG, DCNQC, and DCSMV as suggested in [25].
- **RV** [24] predictor correlates the IR system performance with the volume of the n-parallelepiped encompassing the top-k documents retrieved. [Comment: 2.4] More in detail, we experimented with both the reciprocal volume (RV) and the discounted matryoshka (DM) predictors described in [24] but, since in our scenario the latter appears to be less effective, we report it only the former.
- **DQPP** [3] introduces a small calibrated noise to a query's dense representation, and then as the QPP score, measures the similarity between the original ranked list of documents and the one obtained with the perturbed query.

[Comment: 1.4; RW: 2.2] Our choice of not including supervised baselines stems from three major reasons. First, the proposed model is a post-retrieval unsupervised QPP. Therefore, in our experiments, we employ 17 baselines drawn from this specific family of models. Secondly, our prediction targets are supervised dense models explicitly fine-tuned on MSMARCO training queries. Several supervised QPP [1, 19, 21] were also trained on MSMARCO training queries being one of the richest training data sources. Thus, the same dataset is used to train both the model that produces the IR performance (i.e., the IR system) and the model to predict it (i.e., the QPP). We are not aware of any work, at the current

time, showing that this does not represent a source of bias. Indeed, most of the works focusing on supervised QPP models [1, 16, 19, 21, 74], predict the performance of BM25 or other lexical models, ensuring that no bias is introduced. Finally, Arabzadeh et al. [3], showed that when predicting the performance of a dense IR model, supervised QPPs are on a par, if not worse, than classical unsupervised predictors – except when used on the MSMARCO dataset, further hinting at the hypothesis that they might be biased toward supervised models.

QPP Evaluation Metrics. As evaluation metrics for QPP, we follow the standard protocol of reporting the correlation of predicted QPP estimates and a target metric (measured with Pearson's ρ), and also the rank correlation between the ideal ordering of query performance as obtained by a target metric vs. the predicted ordering obtained via QPP scores (measured with Kendall's τ) [29]. As the target metric, we employ nDCG@10 following previous work [3, 25, 26], and being the official evaluation metric of TREC DL [11, 12]. In addition, we also employ a recently proposed error-based metric - scaled Mean Absolute Rank Error (sMARE) [27, 28], smaller values of which indicate better performance. To provide a consistent interpretation across the metrics, we report the values of one minus the sMARE scores (the range of sMARE values is in $[0, 1]$), which we call $\overline{\text{sMARE}}$.

4.3 Hyper-parameter tuning

For each predictor, we validate the hyper-parameters using the commonly adopted 2-fold validation strategy [18, 24, 63, 74, 75]. Specifically, this commonly used validation strategy involves randomly splitting a set of queries into two partitions, one used as a 'training set' for tuning parameters (for supervised approaches) or hyper-parameters (for unsupervised approaches), and the other partition is used as a 'test set' to evaluate the model performance. The roles of the two partitions are then switched, and the average performance over the two folds is then used as an evaluation measure. Evaluation measures collected this way are then aggregated over 30 random 2-fold splits of the data.

Recall that the hyper-parameters of our proposed method are the three cut-off values k, h and l, denoting the number of top documents to used to aggregate PDD values, the number of ones used as pivots for computing PDD values and the number of documents used to compute the scaling factor based on retrieval scores, respectively (see Equation 6). For a tractable choice of the number of experiments, we set the value of k to 5, which means that PDD values are aggregated over 5 documents. Later, in Section 5.3, we analyze the sensitivity of PDQPP to the number of documents used as pivots. The other two cut-offs in PDQPP, namely h and l, were optimised via grid search over the training splits from the set $\{5, 10, 50, 100, 250, 500\}$.

For a fair comparison, the hyper-parameter k (the number of top-documents used for estimation cut-off) of all the other baseline predictors were also optimised over the training folds. The baseline DQPP involves an additional parameter - the standard deviation of the Gaussian noise used to perturb query vectors. This parameter was validated in the range $[0.01, 0.09]$ with a step of 0.01 following the implementation in the repository provided by Arabzadeh et al.[5].

5 RESULTS

5.1 Comparison with other predictors

As a sanity-checking step, we first report in Table 1 the nDCG@10 (our QPP target metric) values for the various datasets used for each IR model considered in our experiments. It can be seen that the results are consistent with existing numbers reported in the literature [37, 39, 71], which, in turn, shows that our retrieval setup is at par with previous findings.

[5]https://github.com/Narabzad/Dense-QPP

Table 2. Performance of PDQPP compared to the baselines in predicting ANCE's nDCG@10 in terms of Kendall's τ, Pearson's ρ and $\overline{\text{sMARE}}$ (1 − sMARE). For each dataset and IR model, we highlight in bold the best method and underline the runner-up. The postfix '*' indicates QPP models that are statistically at par with the best. The last column reports the effectiveness index (EI), which is the number of times a QPP model either is the winner or ends up being a 'star' competitor (i.e., statistically indistinguishable from the best method).

	DL '19			DL '20			DL Hard			Robust '04			EI
	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	
	ANCE												
SD	0.374*	0.539	0.788*	0.261	0.346	0.745	0.329*	0.396	0.774*	0.401	0.503	0.794	4
$n(\sigma)$	0.319	0.462	0.771	0.187	0.236	0.717	0.288	0.369	0.758	0.369	0.490	0.784	0
Clarity	0.118	0.219	0.703	0.070	0.117	0.685	0.325	**0.510***	0.762	0.093	0.136	0.696	1
NQC	0.371	0.538	0.787	0.260	0.345	0.746	0.334*	0.397	0.775*	0.401	0.503	0.794	2
SMV	0.348	0.500	0.774	0.267	0.331	0.747	0.303	0.318	0.760	0.369	0.448	0.784	0
RSD	0.312	0.441	0.760	0.303	0.445	0.764	**0.380***	0.432	**0.787***	0.406	0.545*	0.793	3
WIG	0.336	0.481	0.757	0.273	0.454*	0.749	0.164	0.248	0.698	**0.426***	**0.561***	**0.800***	4
UEFClarity	0.106	0.131	0.697	0.100	0.110	0.680	0.106	0.157	0.695	0.212	0.278	0.737	0
UEFNQC	0.219	0.329	0.733	0.156	0.282	0.696	0.156	0.191	0.704	0.270	0.293	0.755	0
UEFSMV	0.193	0.300	0.721	0.168	0.260	0.696	0.141	0.163	0.695	0.263	0.291	0.754	0
UEFWIG	0.185	0.207	0.705	0.148	0.211	0.687	0.033	0.046	0.669	0.257	0.330	0.750	0
WRIG	0.330	0.522	0.773	0.229	0.497*	0.742	0.064	0.142	0.687	0.162	0.239	0.710	1
DCNQC	0.353	0.533	0.787	0.242	0.326	0.741	0.331*	0.390	0.771*	0.405	0.505	0.795	2
DCSMV	0.342	0.482	0.771	0.248	0.343	0.736	0.284	0.316	0.760	0.399	0.481	0.795	0
DCWIG	**0.416***	0.548	**0.803***	0.268	0.385	0.736	0.160	0.212	0.713	0.207	0.295	0.725	2
RV	0.200	0.297	0.722	0.288	0.339	0.747	0.039	0.113	0.664	0.238	0.347	0.733	0
DenseQPP	0.349	0.525	0.770	0.157	0.271	0.721	0.163	0.224	0.711	0.225	0.333	0.734	0
BERTQPP-bi	0.105	0.212	0.704	0.078	0.111	0.675	0.260	0.352	0.751	0.126	0.184	0.702	0
BERTQPP-ce	0.154	0.258	0.723	0.098	0.129	0.691	0.346*	0.500*	0.769	0.380	0.484	0.788	2
PDQPP	0.378*	**0.603***	0.788*	**0.396***	**0.519***	**0.787***	0.299	0.397	0.763	0.389	0.510	0.789	6

Tables 2, 3, and 4 report the effectiveness of the proposed PDQPP model in comparison to the different baseline models. The best results obtained for a particular collection are bold-faced, whereas the second-best ones are underlined. We append an asterisk to the approaches that are statistically indistinguishable from the best-performing approach. In particular, for the significance testing we employed ANOVA with Tukey's honestly significant difference (HSD) test with significance level $\alpha = 0.05$ [66].

In addition, to provide further insights into the relative performance of the QPP models, we report the number of times a particular method turns out to be a winner by outperforming other approaches or being statistically indistinguishable from the best-performing model. We call this count the Effectiveness Index (EI) of a model and report its values in the last column of Tables 2 to 4 (higher values of this number indicating better effectiveness). Intuitively speaking, it does not over-penalise a model for not yielding the best results. Instead, it rewards the runner-up model for being statistically indistinguishable from the best one thus factoring in the variational effects of random 2-fold splits - the commonly used setup of QPP experiments [18, 24, 29, 63, 74, 75], as well as the well-known problem of the intrinsic variability of the QPP measurements (see Section 5.2).

In Tables 2, 3, and 4 we see that with a few exceptions, the proposed PDQPP model outperforms the current state of the art models, or is at par with the best approach. With a few exceptions, PDQPP can easily outperform agnostic QPPs (upper part of the Tables). This phenomenon is unsurprising as previous work showed the diminished effectiveness of classic QPP models in dealing with dense and semantic-based IR systems [25, 26]. Furthermore, PDQPP makes use of

Table 3. Performance of PDQPP compared to the baselines in predicting Contriever's nDCG@10 in terms of Kendall's τ, Pearson's ρ and $\overline{\text{sMARE}}$ (1 − sMARE). For each dataset and IR model, we highlight in bold the best method and underline the runner-up. The postfix '*' indicates QPP models that are statistically at par with the best. The last column reports the effectiveness index (EI), which is the number of times a QPP model either is the winner or ends up being a 'star' competitor (i.e., statistically indistinguishable from the best method).

	DL '19			DL '20			DL Hard			Robust '04			EI
	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	
	Contriever												
SD	0.278*	0.457*	0.756*	0.108	0.214	0.695	0.246*	0.206	0.753*	**0.307***	0.380	**0.769***	8
$n(\sigma)$	0.322*	0.448	0.764*	0.067	0.062	0.700	0.243*	0.359*	**0.755***	0.277	0.370	0.765*	6
Clarity	0.124	0.172	0.701	0.005	0.020	0.662	0.225*	0.348*	0.725	0.112	0.142	0.698	2
NQC	0.271	0.429	0.748*	0.082	0.170	0.695	0.239*	0.212	0.750*	0.269	0.349	0.759	3
SMV	0.255	0.424	0.748*	0.051	0.136	0.689	0.191	0.165	0.733	0.254	0.318	0.754	1
RSD	0.220	0.354	0.728	0.198	0.288	0.734	0.268*	0.326*	0.755*	0.238	0.370	0.737	3
WIG	0.227	0.388	0.736	0.116	0.226	0.695	0.121	0.236	0.706	0.236	0.352	0.738	0
UEFClarity	0.142	0.203	0.708	0.080	0.099	0.695	-0.103	-0.168	0.642	0.193	0.276	0.730	0
UEFNQC	0.221	0.286	0.726	0.115	0.153	0.705	0.013	-0.069	0.691	0.236	0.306	0.745	0
UEFSMV	0.227	0.282	0.729	0.110	0.132	0.700	-0.015	-0.087	0.680	0.235	0.297	0.745	0
UEFWIG	0.135	0.211	0.705	0.014	-0.070	0.678	-0.126	-0.185	0.640	0.197	0.265	0.730	0
WRIG	0.272*	0.338	0.731	0.117	0.298	0.703	-0.064	0.097	0.648	0.104	0.157	0.700	1
DCNQC	0.286*	0.440	**0.761***	0.134	0.256	0.711	0.237*	0.222	0.752*	0.256	0.349	0.758	4
DCSMV	0.261	0.432	0.745	0.155	0.259	0.720	0.199	0.198	0.749*	0.241	0.345	0.748	2
DCWIG	**0.323***	**0.512***	0.752*	0.264*	0.406*	0.744*	0.100	0.151	0.687	0.189	0.282	0.716	6
RV	0.127	0.238	0.723	0.235	0.297	0.731	-0.101	-0.125	0.637	0.276	0.394*	0.751	1
DenseQPP	0.181	0.232	0.719	0.103	0.258	0.695	0.100	0.159	0.707	0.233	0.280	0.738	0
BERTQPP-bi	0.088	0.083	0.702	0.069	0.114	0.687	**0.282***	**0.383***	0.746*	-0.013	-0.031	0.663	3
BERTQPP-ce	0.138	0.188	0.720	0.045	0.092	0.678	0.264*	0.378*	0.749*	0.136	0.188	0.706	3
PDQPP	0.280*	0.458*	0.748*	**0.288***	**0.411***	**0.756***	0.229*	0.349*	0.751*	0.294*	**0.404***	0.762	11

topological characteristics of the embedded space in an explicit manner via leveraging subspace projections, which is the likely reason for its superior performance. Indeed, dense-IR based approaches are a more effective comparison with DCNQC, DCWIG or DQPP being particularly effective, depending on the collection/predicted system. IR system-wise, PDQPP is the most effective in predicting the retrieval performance for Contriever (Table 3) and TAS-B (Table 4). As can be seen from the Tables, for both these models PDQPP turns out to be the best or indistinguishable from the best in 11 out of 12 setups. No other baseline approach exhibits this high consistency in predicting the retrieval performance for Contriever and TAS-B.

PDQPP appears to be slightly less effective on ANCE (Table 2), where it belongs to the top-tier of predictors only 6 times out of 12. Notice that it is still the predictor with the highest EI. Furthermore, as per our observations on ANCE, the best baseline predictor depends heavily on the collection considered: for DL '19, we observe good high performance for DCWIG, for DL Hard and Robust '04 the most effective baselines are RSD and WIG. If we inspect the results collection-wise, we notice that PDQPP is particularly effective on DL '19,DL '20, and Robust '04.

5.2 On the improved QPP stability of PDQPP

Overall, the high variance in terms of the quality of the predictions is a well-known problem in the QPP domain [7, 26, 28, 35]. Several factors influence the variability, such as which queries are considered, collections, retrieval models and evaluation measures. For example Hauff [35, p. 83-84] considers different subsets of queries of three collections, TREC

Table 4. Performance of PDQPP compared to the baselines in predicting TAS-B's nDCG@10 in terms of Kendall's τ, Pearson's ρ and $\overline{\text{sMARE}}$ (1 − sMARE). For each dataset and IR model, we highlight in bold the best method and underline the runner-up. The postfix '*' indicates QPP models that are statistically at par with the best. The last column reports the effectiveness index (EI), which is the number of times a QPP model either is the winner or ends up being a 'star' competitor (i.e., statistically indistinguishable from the best method).

	DL '19			DL '20			DL Hard			Robust '04			EI
	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	
	TAS-B												
SD	0.216	0.298	0.733	0.213	0.310	0.728	0.345	0.339	0.759	**0.406***	0.505	**0.796***	1
$n(\sigma)$	0.204	0.260	0.731	0.193	0.287	0.717	0.301	0.373	0.731	0.386	0.504	0.787	0
Clarity	0.128	0.223	0.720	-0.034	-0.101	0.656	0.187	0.299	0.709	0.160	0.218	0.721	0
NQC	0.216	0.288	0.734	0.182	0.287	0.719	0.345	0.323	0.759	0.397	0.493	0.793*	1
SMV	0.210	0.260	0.728	0.178	0.177	0.723	0.249	0.215	0.736	0.371	0.430	0.782	0
RSD	0.191	0.294	0.715	0.249*	0.374	0.747*	0.292	0.420	0.760	0.382	0.541*	0.783	3
WIG	0.202	0.370*	0.728	0.170	0.262	0.706	0.190	0.311	0.710	0.323	0.477	0.767	1
UEFClarity	0.174	0.251	0.718	-0.033	-0.109	0.655	-0.078	-0.034	0.646	0.196	0.305	0.724	0
UEFNQC	0.210	0.225	0.734	0.060	0.129	0.686	-0.004	0.029	0.670	0.248	0.346	0.743	0
UEFSMV	0.191	0.204	0.727	0.065	0.066	0.683	-0.040	0.008	0.662	0.241	0.334	0.741	0
UEFWIG	0.144	0.195	0.706	0.004	-0.035	0.669	-0.151	-0.108	0.633	0.217	0.314	0.729	0
WRIG	0.254	0.392*	0.761*	0.228	0.175	0.739	0.108	0.123	0.677	0.166	0.260	0.717	2
DCNQC	0.196	0.276	0.727	0.170	0.331	0.719	0.357	0.441	0.775	0.402*	0.534*	0.792	2
DCSMV	0.192	0.256	0.725	0.193	0.342	0.723	0.262	0.401	0.748	0.400*	0.521	0.790	1
DCWIG	0.172	0.164	0.719	0.172	0.254	0.724	-0.215	-0.223	0.605	0.304	0.440	0.760	0
RV	0.146	0.250	0.715	0.283*	**0.474***	0.748*	-0.066	-0.063	0.635	0.220	0.327	0.739	3
DenseQPP	0.196	0.243	0.734	-0.019	-0.015	0.667	0.143	0.220	0.705	0.247	0.380	0.743	0
BERTQPP-bi	0.049	0.074	0.688	0.018	0.104	0.666	0.355	0.501	0.777	-0.014	-0.037	0.659	0
BERTQPP-ce	0.032	0.013	0.687	0.019	0.075	0.669	**0.451***	**0.620***	**0.803***	0.197	0.298	0.725	3
PDQPP	**0.332***	**0.408***	**0.766***	**0.291***	0.432*	**0.759***	0.309	0.446*	0.752	**0.406***	**0.548***	0.793*	10

Vol. 4 and 5, WT10g, and GOV2, observing how the best QPP heavily depends on which subset of queries is considered. On a different line, but with similar conclusions, Carmel and Yom-Tov [7, p. 23-24,35-36] apply several predictors on multiple collections, observing high volatility in terms of which QPP can be considered the most effective, depending on the collection. [54] employed 9 different corpora, observing again variability in which system performs the best. More recently, Ganguly et al. [30] explore the impact that several factors have on the QPP effectiveness, observing important consequences linked to the chosen metric as well as the IR system. Finally, Faggioli et al. [26] investigate several predictors applied on lexical and neural IR systems, observing a strong variability on what is the best predictor, depending on which IR system we are trying to predict the performance for.

The very same behaviour can be observed in our results reported in Tables 2, 3, and 4). Depending on what collection is considered and which retrieval model is the target of our predictions, we observe most of the baselines exhibit a high variance in evaluation metric values. For example, consider Table 2, where we observe that when predicting the performance of ANCE on Robust '04, WIG is the best system. If we apply WIG on Contriever and DL '20 (Table 3), WIG performance is 63% worse than PDQPP, the most effective predictor in those cases. Generally speaking, this pattern is more severe for agnostic predictors than for dense ones (with few exceptions, such as HV and DenseQPP which also exhibit instability).

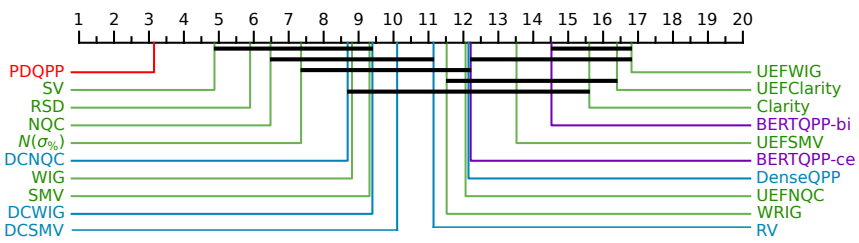


Fig. 2. Critical difference diagram across all experimental settings (IR system, collection, correlation measure). The average rank for PDQPP is 3.15, and it is statistically better than the average rank of the second best (SV, with an average rank of 4.88).

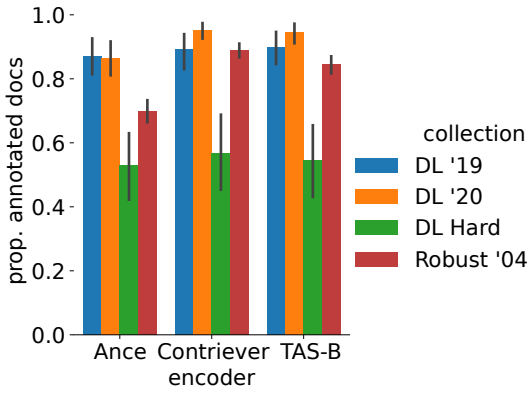


Fig. 3. Proportion of annotated documents among the top 10 retrieved documents by each IR system and for each retrieval collection.

The major advantage of PDQPP is indeed able to provide a more stable performance than the current baseline predictors (as can be seen from the consistency in the EI values from Tables 2 to 4). Even in scenarios where PDQPP fails to outperform all the baselines, it is either statistically at par with the best, or reasonably close to the best.

To better exemplify this, we report the *critical difference diagram* of the evaluated QPP in Figure 2. The critical difference diagram reports on the x-axis (on top) the rank, indicating what is the average rank for a QPP over the various experimental settings (i.e., retrieval model, collection, and correlation measure considered). Furthermore, the thick horizontal lines represent groups of statistically equivalent approaches, according to the Wilcoxon test [70] corrected according to the Holm correction procedure [38]. For example, in Figure 2, we observe that the average rank of PDQPP is 3.15. Furthermore, the second-best approach is SD with an average rank of 4.88. PDQPP is statistically the best according to the multiple-comparison adjusted Wilcoxon test, while the second-best, SD, given the high variance in its rank across different scenarios, is statistically at par with RSD, NQC, $n(\sigma_{\%})$, DCNQC, WIG, SMV and DCWIG.

While designing a QPP that performs the best on all possible situations – predicted IR system, measure, collection – is a very complex task, we argue that the QPP systems should be reasonably reliable, without major drops in performance which render them untrustworthy. Our choice of including multiple IR systems and collections in our analyses aims at showing the overall stability of the proposed PDQPP. Indeed, where PDQPP is not the best, it still provides reasonable guarantees of effectiveness, even if compared against an always different most effective predictor.

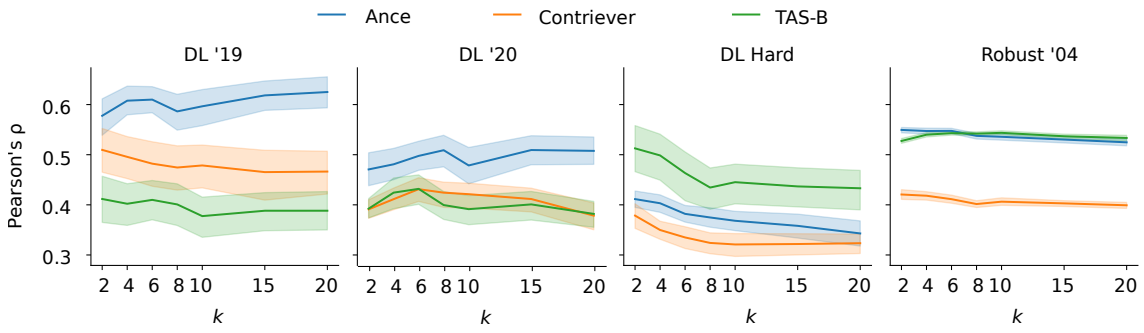


Fig. 4. The performance of the PDQPP when varying the number of pseudo-relevant documents. The general trend suggests that choosing 2-6 documents as pseudo-relevant is the most effective strategy, but large confidence intervals (or the almost flat lines for Robust '04), indicate a relatively small impact on the performance due to picking a wrong amount of pseudo-relevant documents.

[Comment: 3.2] As an additional analysis, we test if there are differences in the experimental setting that might cause variations in the performance of PDQPP. First, we notice that the performance of PDQPP are particularly unstable when it comes to DL Hard. The major difference between DL Hard and other collections is the usage of sparse annotations [44] More in detail, out of the 50 topics available in DL Hard, 25 (and their corresponding relevance judgments) were taken from DL '19 and DL '20. The other 25 were annotated by Mackie et al. [44] using a shallow pooling and assessing only the top 10 documents [44]. To highlight this phenomenon, Figure 3 reports the proportion of annotated documents among the top 10 documents retrieved for each collection by each system. As expected, DL Hard is the collection with the lowest proportion of annotated documents compared to other collections: approximately, only half of the documents in the top 10 for each topic have a corresponding relevance judgment. On the contrary, DL '20 and DL '19 are the collections for which we always have the largest proportion of annotated documents. Finally, for Robust '04, we notice that the proportion of annotated documents among the top 10 retrieved is larger when Contriever and TAS-B are used as retrieval models, while it is lower when ANCE is used. In general, the proportion of annotations for ANCE is lower, regardless of the collection considered. This behaviour perfectly aligns with the stability of the results and the effectiveness of PDQPP. Indeed, when DL Hard is used as a testbed, the best-performing predictor tends to vary, depending on the IR model considered. Standard deviation-based predictors (SD, NQC, RSD and DCNQC), tend to be the best options on such a collection.

5.3 Sensitivity to the pivot documents

To keep the number of experiments to tractable limits, we used the set of top 5 documents as pivots for computing the PDD values, i.e., we set $k = 5$, for all the results reported in Table 2 to 4. We now analyse the sensitivity of our predictor on this parameter. Figure 4 reports the effect of modifying the number of pivot documents from which we can make some interesting observations.

Firstly, we observe that since DL '19, DL '20, and DL Hard contain a much smaller number of queries than Robust '04, as a result of which, the performance of PDQPP on such collections is affected by a larger variance. Secondly, as a general trend, we observe that the performance tends to decrease with an increase in the number of pivot documents. This is in line with our hypothesis that such documents provide a way to disambiguate the meanings of the query in a latent space. The more documents we use further down a ranked list to define the reference spaces for computing the projection

Table 5. Performance of PDQPP compared to the baselines in predicting MiniLM-l12's nDCG@10 in terms of Kendall's τ, Pearson's ρ and $\overline{\text{sMARE}}$ (1 − sMARE). For each dataset and IR model, we highlight in bold the best method and underline the runner-up. The postfix '*' indicates QPP models that are statistically at par with the best. The last column reports the effectiveness index (EI), which is the number of times a QPP model either is the winner or ends up being a 'star' competitor (i.e., statistically indistinguishable from the best method).

	DL '19			DL '20			DL Hard			Robust '04			EI
	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	τ	ρ	$\overline{\text{sMARE}}$	
	MiniLM-l12												
SD	.270	.467	.751*	.091	.174	.692	.220	.337	.737	.348	.462	.777	1
$n(\sigma)$	.304*	.418	**.765***	.035	.129	.681	.299*	.427*	**.760***	.291	.399	.755	5
Clarity	.059	.063	.698	.007	-.007	.663	.165	.274	.715	.150	.224	.708	0
NQC	.256	.396	.749*	.052	.113	.676	.239	.326	.750*	.291	.363	.757	2
SMV	.225	.362	.738	.060	.086	.681	.168	.219	.730	.271	.340	.748	0
RSD	.146	.361	.710	.116	.234	.702	**.349***	**.495***	.758*	.346	.473	.778	3
WIG	.206	.417	.744	.053	.086	.681	.111	.171	.699	.254	.374	.751	0
UEFClarity	.161	.293	.711	-.074	-.149	.644	-.134	-.171	.636	.250	.399	.739	0
UEFNQC	.225	.335	.732	-.011	.009	.663	-.066	-.045	.662	.292	.399	.751	0
UEFSMV	.196	.310	.721	-.021	.010	.659	-.083	-.075	.656	.283	.389	.748	0
UEFWIG	.167	.224	.707	-.063	-.027	.645	-.146	-.202	.638	.231	.341	.737	0
WRIG	**.321***	.449	.759*	.110	.176	.700	-.082	-.074	.648	.011	.058	.666	2
DCNQC	.057	.028	.667	-.183	-.151	.621	-.003	-.207	.672	.174	.047	.722	0
DCSMV	.054	.030	.666	-.172	-.148	.622	.005	-.202	.674	.171	.046	.721	0
DCWIG	.262	.480	.745	.303*	**.447***	.760*	.069	.026	.690	.263	.390	.749	3
RV	.245	.418	.741	**.346***	.373	**.764***	-.038	-.044	.640	.247	.367	.744	2
DenseQPP	.112	.395	.686	.094	.191	.702	-.056	-.054	.650	.260	.377	.744	0
PDQPP	.272*	**.578***	.752*	.340*	.321	.763*	.194	.239	.721	**.363***	**.487***	**.784***	8

displacements, the more the chances are that such documents are not relevant to the query, thus incorporating noise in the prediction.

We also observe that the QPP effectiveness mostly decreases (often monotonically with a small number of exceptions) with an increase in k, e.g., see the results for the DL Hard collection. For some collections, we observe that the QPP effectiveness peaks at a value close to the range of about 2 to 4 documents beyond which it decreases almost steadily, e.g., see the plots for DL '19 and DL '20 collections. The ANCE model on DL '19 and DL '20 collections shows a reverse trend of improved QPP effectiveness with a larger number of pivots.

5.4 Dense models with Fewer Dimensions

[Comment: 1.6] As additional evidence of the robustness of PDQPP, we test its capabilities using a model that encodes the text (queries and documents) in a space with fewer dimensions. In particular, we test it to predict the performance of MiniLM-l12[6] [69]. Table 5 reports the performance of PDQPP when used to predict the performance of MiniLM-l12. Overall, the patterns align with what was observed in Sections 5.1 and 5.2. First of all, we can notice that in terms of EI, the PDQPP predictor is the most effective approach, further highlighting the general stability of the model in multiple scenarios. If we consider the results by individual collection, we notice that the predictor is particularly effective on the Robust '04 collection, where it is the best approach by a statistically significant margin regardless of the measure

[6]https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

Table 6. Comparison between PDQPP, its numerator (PDD), and the normalization factor that corresponds to the standard deviation of the scores, i.e., the SD predictor.

	DL '19			DL '20			DL Hard			Robust '04		
	τ	ρ	sMARE	τ	ρ	sMARE	τ	ρ	sMARE	τ	ρ	sMARE
	ANCE											
SD (den)	.374*	.539	**.788***	.261	.346	.745	**.329***	.396*	**.774***	**.401***	.503*	**.794***
PDD (num)	.248	.387	.732	.326	.457	.764	-.108	-.146	.631	.216	.301	.725
PDQPP	**.378***	**.603***	**.788***	**.396***	**.519***	**.787***	.299*	**.397***	.763	.389	**.510***	.789
	Contriever											
SD (den)	.278*	.457*	**.756***	.108	.214	.695	**.246***	.206	**.753***	**.307***	.380	**.769***
PDD (num)	.143	.261	.717	.238	.337	.735	-.084	-.122	.644	.124	.182	.706
PDQPP	**.280***	**.458***	.748*	**.288***	**.411***	**.756***	.229*	**.349***	.751*	.294*	**.404***	.762
	TAS-B											
SD (den)	.216	.298	.733	.213	.310	.728	**.345***	.339	**.759***	**.406***	.505	**.796***
PDD (num)	.220	.261	.737	.287*	.409*	.756*	.136	.213	.687	.141	.223	.714
PDQPP	**.332***	**.408***	**.766***	**.291***	**.432***	**.759***	.309*	**.446***	.752*	**.406***	**.548***	.793*

considered, If we consider DL '19 and DL '20, we notice that in most of the cases (except for DL '20 and Pearson's ρ), the predictor belongs to the top group and is statistically equivalent to the best predictor. For what concerns DL '19, the best predictors are either $n(\sigma_{\%})$ or WRIG, depending on the measure. For DL '20 the best predictors are either DCWIG or RV. This is again a sign of the robustness of the PDQPP: even if it is not necessarily the top-performing predictor in all scenarios, it is statistically on a par with the best-performing solution, which is different depending on the collection. When it comes to the DL Hard collection, the best predictors are either $n(\sigma_{\%})$ or RSD: in line with what was observed before, given the intrinsic instability of the collection, using a simple standard deviation-based solution is the best approach.

5.5 The role of the Normalization Factor

[Comment: 1.5] An interesting analysis concerns the role played by PDQPP components in determining its performance. To this end, Table 6 reports the performance of PDQPP compared to its denominator (which corresponds to the SD predictor, i.e., the standard deviation of the retrieval scores of the top-k retrieved documents) and its numerator, i.e., the sum of the PDD for the top-k documents. Table 6 shows that, for both DL '19 and DL '20, the numerator (i.e., PDD) contributes positively to the performance. For both collections, PDQPP tends to be more effective than its parts. The only exceptions are when we try to predict ANCE applied on DL '19 and using sMARE as evaluation measure, where PDQPP performs as SD, and when we predict the performance of Contriever applied on DL '19 using sMARE, where SD is more effective than PDQPP. In most cases, the improvement induced by combining the denominator and numerator is statistically significant over at least one of the two strategies alone.

5.6 Using arbitrary subspaces for projections

As mentioned in Section 3, PDQPP relies on pseudo-relevant documents to identify the axes on which to project the query and retrieved documents. While we argue that this approach allows leveraging the information within the ranked

Table 7. A comparison between the original PDQPP and its three variants where directions to project the embedded document and query vectors are sampled from different distributions. Similar to the results of Tables 2, 3, and 4, the target IR metric to compute QPP effectiveness is nDCG@10.

	DL '19			DL '20			DL Hard			Robust '04		
	τ	ρ	sMARE	τ	ρ	sMARE	τ	ρ	sMARE	τ	ρ	sMARE
	ANCE											
R-PDQPP	.149	.239	.709	.155	.139	.707	.162	.241	.716	.242	.340	.734
Q-PDQPP	.256	.432	.748	.246	.332	.735	.214	.317	.734	.250	.368	.742
D-PDQPP	.315	**.610***	.765	.315	.406	.761	.260*	.361*	.743	.347	.478	.770
PDQPP	**.378***	.603*	**.788***	**.396***	**.519***	**.787***	**.299***	**.397***	**.763***	**.389***	**.510***	**.789***
	Contriever											
R-PDQPP	.124	.161	.704	.203	.271	.720	.065	.080	.693	.193	.278	.728
Q-PDQPP	.124	.182	.700	.197	.305	.722	.081	.128	.700	.198	.278	.728
D-PDQPP	.278*	.368	**.753***	.220	.359*	.730	.220*	.319*	.747*	.261	.360	.749
PDQPP	**.280***	**.458***	.748*	**.288***	**.411***	**.756***	**.229***	**.349***	**.751***	**.294***	**.404***	**.762***
	TAS-B											
R-PDQPP	.207	.251	.718	.108	.142	.693	.225	.208	.725	.219	.336	.730
Q-PDQPP	.147	.261	.702	.189	.179	.729	.197	.252	.724	.225	.324	.731
D-PDQPP	.231	.297	.729	.207	.280	.735	.249	.286	.738*	.364	.506	.780
PDQPP	**.332***	**.408***	**.766***	**.291***	**.432***	**.759***	**.309***	**.446***	**.752***	**.406***	**.548***	**.793***

list itself, there might be alternative approaches to sample directions that might also be effective. Therefore, we conduct additional experiments with three different variations of PDQPP, each with its way of obtaining the directions to compute the projection displacements, as detailed below.

- **Random PDQPP (R-PDQPP)** samples the directions from a Normal distribution centered at zero with standard deviation tuned in $[0.1, 0.9]$ with steps of 0.1.
- **Query PDQPP (Q-PDQPP)** samples directions by perturbing the query with random noise drawn from a Normal distribution centred at zero, with standard deviation tuned in $[0.1, 0.9]$ with steps of 0.1. DQPP uses the same method for generating perturbed queries. However, DQPP does not involve computing projection displacements as is the case for the variant Q-PDQPP.
- **Documents PDQPP (D-PDQPP)** samples directions from an isotropic multivariate Normal distribution with parameters estimated from the top-5 document vector samples.

Table 7 reports a comparison of these variants with the originally proposed predictor (Equation 6). As a general trend, we observe that R-PDQPP is the worst-performing solution, the likely reason for which can be attributed to the fact that the projection axes being randomly sampled do not contain enough semantic information to differentiate between the different aspects of the information need inherent in a query. [Comment: 3.3] Depending on the scenario, Q-PDQPP is more effective than R-PDQPP. Nevertheless, we generally observe low performance, suggesting that the query's geometric perturbation does not allow the perturbation induced on the document list to be considered a performance indicator. In fact, we can assume that different distributions of documents' vectors around the query vector (i.e., different queries produce different distributions of scores). Neglecting this aspect reduces the effectiveness of the predictor. To

confirm this, we observe that the improved performance exhibited by D-PDQPP is based on sampling the pivot vector from a distribution that is constructed considering the top retrieved documents: this allows the pivot vector to better model the space around the query. In a small fraction of cases, we observe that D-PDQPP is statistically as effective as PDQPP, with even larger average scores in a couple of scenarios (DL '19, ANCE, Pearson's ρm and DL '19, Contriever, sMARE)). Overall, the most effective solution remains PDQPP. This suggests that using pseudo-relevant documents as pivots is the best approach, and it should be favored by the practitioners.

5.7 PDQPP Limitations

While compared to the current state of the art PDQPP appears more robust and capable of achieving good performance across all scenarios, three major limitations that affect PDQPP need to be discussed. *Limitation 1*: PDQPP is not a model agnostic QPP. Indeed, PDQPP can be applied to predict the performance only of dense IR models. Two mitigating conditions should be taken into consideration. First, dense models are more and more popular in the IR community. It is usually common for IR pipelines to include a dense component for the purposes of first stage retrieval (as in this work), for reranking, or for both. Even though PDQPP, in principle, can also be applied for sparse vectors, the method is particularly suitable for dense vectors. This is because the projection of a sparse vector over another sparse one can lead to an abrupt effect of removing term weights from the former thus making it more sparse. Whereas for dense vectors the projections over subspaces retain more information. The popularity of dense IR models motivates its importance. Secondly, a model agnostic QPP cannot take into consideration specific additional information available to the IR model that might lead to an improvement in performance. In this case, the predictor exploits the geometric properties of the embedding space to better identify queries whose documents are affected by high variability in their semantics, suggesting possibly weak retrieval.

Limitation 2: PDQPP may not be suited for scenarios where the diversity in results is particularly important. PDQPP operates under the assumption that a stable and coherent retrieval list is likely more effective than a highly diversified one. These assumptions underly many QPPs such as Clarity [13], the UEF framework [61] or the reference lists framework [56]. This might not be the case of a fairness-oriented IR system which aims at maximizing the diversity of the results. Nevertheless, as future research direction, PDQPP should be tested for fairness-oriented IR tasks.

Limitation 3: PDQPP is not always the best performing QPP. This limitation has been extensively discussed in 5.2. To summarize such discussion, PDQPP is the most stable predictor compared to all other baselines, making it reliable even when it is not the most effective predictor. Conversely, most of the other approaches exhibit both gains and losses of high magnitudes in effectiveness depending on the experimental setup considered.

6 CONCLUSIONS AND FUTURE WORK

In this work, we proposed PDQPP, a novel QPP model capable of exploiting geometric properties in a dense embedding space to predict IR performance. The proposed predictor is based on the concept of *projection displacement*: we project the query and the retrieved documents on a reference subspace induced by the pseudo-relevant documents. The change of retrieval scores observed in the novel space represents a measure of the incoherence of the IR system. If, in the novel subspace, the query and the documents remain closely related, then we can assume the dense IR system to be successful. On the other hand, if we observe major changes in the novel subs-pace, then it is possible that the retrieval was unsuccessful and the performance will be low. In terms of effectiveness, the proposed QPP model can overcome several state-of-the-art baselines under a wide range of settings. Additionally, we also show that using pseudo-documents as subspaces yield better solutions than to use randomly selected ones.

In future directions, we plan to extend our predictor to other types of representation-learning based IR systems, including distillation models of late-interaction systems and sparse IR systems. We also plan to investigate other strategies to devise projection spaces, such as the space defined by previous utterances for a conversational search system or clustering of documents.

REFERENCES

[1] Negar Arabzadeh, Maryam Khodabakhsh, and Ebrahim Bagheri. 2021. BERT-QPP: Contextualized Pre-trained transformers for Query Performance Prediction. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 2857–2861. https://doi.org/10.1145/3459637.3482063

[2] Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi, and Ebrahim Bagheri. 2024. Query Performance Prediction: From Fundamentals to Advanced Techniques. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 14612)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 381–388. https://doi.org/10.1007/978-3-031-56069-9_51

[3] Negar Arabzadeh, Radin Hamidi Rad, Maryam Khodabakhsh, and Ebrahim Bagheri. 2023. Noisy Perturbations for Estimating Query Difficulty in Dense Retrievers. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, 3722–3727. https://doi.org/10.1145/3583780.3615270

[4] Negar Arabzadeh, Mahsa Seifikar, and Charles L. A. Clarke. 2022. Unsupervised Question Clarity Prediction through Retrieved Item Coherency. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, 3811–3816. https://doi.org/10.1145/3511808.3557719

[5] Negar Arabzadeh, Fattane Zarrinkalam, Jelena Jovanovic, Feras N. Al-Obeidat, and Ebrahim Bagheri. 2020. Neural embedding-based specificity metrics for pre-retrieval query performance prediction. *Inf. Process. Manag.* 57, 4 (2020), 102248. https://doi.org/10.1016/j.ipm.2020.102248

[6] Negar Arabzadeh, Fattane Zarrinkalam, Jelena Jovanovic, and Ebrahim Bagheri. 2020. 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 (Lecture Notes in Computer Science, Vol. 12036)*, Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins (Eds.). Springer, 78–85. https://doi.org/10.1007/978-3-030-45442-5_10

[7] David Carmel and Elad Yom-Tov. 2010. *Estimating the Query Difficulty for Information Retrieval.* Morgan & Claypool Publishers. https://doi.org/10.2200/S00235ED1V01Y201004ICR015

[8] Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. 2020. Retrievability based Document Selection for Relevance Feedback with Automatically Generated Query Variants. In *CIKM*. ACM, 125–134.

[9] Xiaoyang Chen, Ben He, and Le Sun. 2022. Groupwise Query Performance Prediction with BERT. In *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 13186)*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer, 64–74. https://doi.org/10.1007/978-3-030-99739-7_8

[10] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. 1998. Efficient Construction of Large Test Collections. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel (Eds.). ACM, 282–289. https://doi.org/10.1145/290941.291009

[11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *CoRR* abs/2102.07662 (2021). arXiv:2102.07662 https://arxiv.org/abs/2102.07662

[12] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR* abs/2003.07820 (2020). arXiv:2003.07820 https://arxiv.org/abs/2003.07820

[13] Stephen Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland*, Kalervo Järvelin, Micheline Beaulieu, Ricardo A. Baeza-Yates, and Sung-Hyon Myaeng (Eds.). ACM, 299–306. https://doi.org/10.1145/564376.564429

[14] Ronan Cummins, Joemon M. Jose, and Colm O'Riordan. 2011. Improved query performance prediction using standard deviation. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, Wei-Ying Ma, Jian-Yun Nie, Ricardo Baeza-Yates, Tat-Seng Chua, and W. Bruce Croft (Eds.). ACM, 1089–1090. https://doi.org/10.1145/2009916.2010063

[15] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (Eds.). ACM, 126–134. https://doi.org/10.1145/3159652.3159659

[16] 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 *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (Eds.). ACM, 201–209.

https://doi.org/10.1145/3488560.3498491

[17] Suchana Datta, Debasis Ganguly, Sean MacAvaney, and Derek Greene. 2024. A Deep Learning Approach for Selective Relevance Feedback. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 14609)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 189–204. https://doi.org/10.1007/978-3-031-56060-6_13

[18] Suchana Datta, Debasis Ganguly, Mandar Mitra, and Derek Greene. 2023. A Relative Information Gain-based Query Performance Prediction Framework with Generated Query Variants. *ACM Trans. Inf. Syst.* 41, 2 (2023), 38:1–38:31. https://doi.org/10.1145/3545112

[19] Suchana Datta, Sean MacAvaney, Debasis Ganguly, and Derek Greene. 2022. A 'Pointwise-Query, Listwise-Document' Based Query Performance Prediction Approach. In *Proceedings of 45th international ACM SIGIR conference research development in information retrieval*. 2148—-2153. https://doi.org/10.1145/3477495.3531821

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. ACL, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[21] Sajad Ebrahimi, Maryam Khodabakhsh, Negar Arabzadeh, and Ebrahim Bagheri. 2024. Estimating Query Performance Through Rich Contextualized Query Representations. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part IV (Lecture Notes in Computer Science, Vol. 14611)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 49–58. https://doi.org/10.1007/978-3-031-56066-8_6

[22] Guglielmo Faggioli, Nicola Ferro, Josiane Mothe, and Fiana Raiber. 2023. QPP++ 2023: Query-Performance Prediction and Its Evaluation in New Tasks. In *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 13982)*. Springer, 388–391. https://doi.org/10.1007/978-3-031-28241-6_42

[23] Guglielmo Faggioli, Nicola Ferro, Josiane Mothe, and Fiana Raiber. 2023. QPP++ 2023: Query-Performance Prediction and Its Evaluation in New Tasks. In *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 13982)*, Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo (Eds.). Springer, 388–391. https://doi.org/10.1007/978-3-031-28241-6_42

[24] Guglielmo Faggioli, Nicola Ferro, Cristina Muntean, Raffaele Perego, and Nicola Tonellotto. 2023. A Geometric Framework for Query Performance Prediction in Conversational Search. In *Proceedings of 46th international ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2023 July 23–27, 2023, Taipei, Taiwan*. ACM. https://doi.org/10.1145/3539618.3591625

[25] Guglielmo Faggioli, Thibault Formal, Simon Lupart, Stefano Marchesin, Stéphane Clinchant, Nicola Ferro, and Benjamin Piwowarski. 2023. Towards Query Performance Prediction for Neural Information Retrieval: Challenges and Opportunities. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2023, Taipei, Taiwan, 23 July 2023*, Masaharu Yoshioka, Julia Kiseleva, and Mohammad Aliannejadi (Eds.). ACM, 51–63. https://doi.org/10.1145/3578337.3605142

[26] Guglielmo Faggioli, Thibault Formal, Stefano Marchesin, Stéphane Clinchant, Nicola Ferro, and Benjamin Piwowarski. 2023. Query Performance Prediction for Neural IR: Are We There Yet?. In *Advances in Information Retrieval - 45th European Conference on IR Research, ECIR 2023, Dublin, Ireland, April 2-6, 2023*. 1–18. https://doi.org/10.48550/ARXIV.2302.09947

[27] Guglielmo Faggioli, Oleg Zendel, J. Shane Culpepper, Nicola Ferro, and Falk Scholer. 2021. An Enhanced Evaluation Framework for Query Performance Prediction. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12656)*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer, 115–129. https://doi.org/10.1007/978-3-030-72113-8_8

[28] Guglielmo Faggioli, Oleg Zendel, J. Shane Culpepper, Nicola Ferro, and Falk Scholer. 2022. sMARE: a new paradigm to evaluate and understand query performance prediction methods. *Inf. Retr. J.* 25, 2 (2022), 94–122. https://doi.org/10.1007/s10791-022-09407-w

[29] Debasis Ganguly, Suchana Datta, Mandar Mitra, and Derek Greene. 2022. An Analysis of Variations in the Effectiveness of Query Performance Prediction. In *ECIR (1) (Lecture Notes in Computer Science, Vol. 13185)*. Springer, 215–229.

[30] Debasis Ganguly, Suchana Datta, Mandar Mitra, and Derek Greene. 2022. An Analysis of Variations in the Effectiveness of Query Performance Prediction. In *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 13185)*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer, 215–229. https://doi.org/10.1007/978-3-030-99736-6_15

[31] Debasis Ganguly and Emine Yilmaz. 2023. Query-specific Variable Depth Pooling via Query Performance Prediction towards Reducing Relevance Assessment Effort. *CoRR* abs/2304.11752 (2023). https://doi.org/10.48550/arXiv.2304.11752 arXiv:2304.11752

[32] Debasis Ganguly and Emine Yilmaz. 2023. Query-specific Variable Depth Pooling via Query Performance Prediction towards Reducing Relevance Assessment Effort. *CoRR* abs/2304.11752 (2023).

[33] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi (Eds.). ACM, 55–64. https://doi.org/10.1145/2983323.2983769

[34] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2019. Performance Prediction for Non-Factoid Question Answering. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2019, Santa Clara, CA, USA, October 2-5, 2019*, Yi Fang, Yi Zhang, James Allan, Krisztian Balog, Ben Carterette, and Jiafeng Guo (Eds.). ACM, 55–58. https://doi.org/10.1145/3341981.3344249

[35] Claudia Hauff. 2010. Predicting the effectiveness of queries and retrieval systems. *SIGIR Forum* 44, 1 (2010), 88. https://doi.org/10.1145/1842890.1842906

[36] 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 2008, Napa Valley, California, USA, October 26-30, 2008*, James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury (Eds.). ACM, 1419–1420. https://doi.org/10.1145/1458082.1458311

[37] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 113–122. https://doi.org/10.1145/3404835.3462891

[38] Sture Holm. 1979. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics* 6, 2 (1979), 65–70. http://www.jstor.org/stable/4615733

[39] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards Unsupervised Dense Information Retrieval with Contrastive Learning. *CoRR* abs/2112.09118 (2021). arXiv:2112.09118 https://arxiv.org/abs/2112.09118

[40] Omar Khattab and Matei Zaharia. 2020. 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 2020, Virtual Event, China, July 25-30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 39–48. https://doi.org/10.1145/3397271.3401075

[41] Maryam Khodabakhsh and Ebrahim Bagheri. 2021. Semantics-enabled query performance prediction for ad hoc table retrieval. *Inf. Process. Manag.* 58, 1 (2021), 102399. https://doi.org/10.1016/j.ipm.2020.102399

[42] Maryam Khodabakhsh, Fattane Zarrinkalam, and Negar Arabzadeh. 2024. BertPE: A BERT-Based Pre-retrieval Estimator for Query Performance Prediction. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 14610)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 354–363. https://doi.org/10.1007/978-3-031-56063-7_27

[43] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Trans. Assoc. Comput. Linguistics* 9 (2021), 329–345. https://doi.org/10.1162/tacl_a_00369

[44] Iain Mackie, Jeffrey Dalton, and Andrew Yates. 2021. How Deep is your Learning: the DL-HARD Annotated Deep Learning Dataset. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2335–2341. https://doi.org/10.1145/3404835.3463262

[45] Stefano Marchesin, Alberto Purpura, and Gianmaria Silvello. 2020. Focal elements of neural information retrieval models. An outlook through a reproducibility study. *Inf. Process. Manag.* 57, 6 (2020), 102109. https://doi.org/10.1016/J.IPM.2019.102109

[46] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. 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*. http://arxiv.org/abs/1301.3781

[47] Josiane Mothe and Ludovic Tanguy. 2005. Linguistic features to predict query difficulty. In *ACM Conference on research and Development in Information Retrieval, SIGIR, Predicting query difficulty - methods and applications workshop*. Salvador de Bahia, Brazil, 7–10.

[48] 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 *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016 (CEUR Workshop Proceedings, Vol. 1773)*, Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

[49] Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. *CoRR* abs/1910.14424 (2019). arXiv:1910.14424 http://arxiv.org/abs/1910.14424

[50] Rodrigo Frassetto Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *CoRR* abs/1904.08375 (2019). arXiv:1904.08375 http://arxiv.org/abs/1904.08375

[51] Joaquín Pérez-Iglesias and Lourdes Araujo. 2010. Standard Deviation as a Query Hardness Estimator. In *String Processing and Information Retrieval - 17th International Symposium, SPIRE 2010, Los Cabos, Mexico, October 11-13, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6393)*, Edgar Chávez and Stefano Lonardi (Eds.). Springer, 207–212. https://doi.org/10.1007/978-3-642-16321-0_21

[52] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. *CoRR* abs/1904.07531 (2019). arXiv:1904.07531 http://arxiv.org/abs/1904.07531

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8748–8763. http://proceedings.mlr.press/v139/radford21a.html

[54] Fiana Raiber and Oren Kurland. 2014. Query-performance prediction: setting the expectations straight. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 13–22. https://doi.org/10.1145/2600428.2609581

[55] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[56] Haggai Roitman. 2017. An Enhanced Approach to Query Performance Prediction Using Reference Lists. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 869–872. https://doi.org/10.1145/3077136.3080665

[57] Haggai Roitman. 2019. Normalized Query Commitment Revisited. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 1085–1088. https://doi.org/10.1145/3331184.3331334

[58] Haggai Roitman, Shai Erera, and Bar Weiner. 2017. Robust Standard Deviation Estimation for Query Performance Prediction. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2017, Amsterdam, The Netherlands, October 1-4, 2017*, Jaap Kamps, Evangelos Kanoulas, Maarten de Rijke, Hui Fang, and Emine Yilmaz (Eds.). ACM, 245–248. https://doi.org/10.1145/3121050.3121087

[59] Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth J. F. Jones. 2019. Estimating Gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction. *Inf. Process. Manag.* 56, 3 (2019), 1026–1045. https://doi.org/10.1016/j.ipm.2018.10.009

[60] Abbas Saleminezhad, Negar Arabzadeh, Soosan Beheshti, and Ebrahim Bagheri. 2024. Context-Aware Query Term Difficulty Estimation for Performance Prediction. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part IV (Lecture Notes in Computer Science, Vol. 14611)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 30–39. https://doi.org/10.1007/978-3-031-56066-8_4

[61] Anna Shtok, Oren Kurland, and David Carmel. 2010. Using statistical decision theory and relevance models for query-performance prediction. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy (Eds.). ACM, 259–266. https://doi.org/10.1145/1835449.1835494

[62] Anna Shtok, Oren Kurland, and David Carmel. 2016. Query Performance Prediction Using Reference Lists. *ACM Trans. Inf. Syst.* 34, 4 (2016), 19:1–19:34. https://doi.org/10.1145/2926790

[63] 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 (2012), 11:1–11:35. https://doi.org/10.1145/2180868.2180873

[64] Ashutosh Singh, Debasis Ganguly, Suchana Datta, and Craig MacDonald. 2023. Unsupervised Query Performance Prediction for Neural Models with Pairwise Rank Preferences. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (Eds.). ACM, 2486–2490. https://doi.org/10.1145/3539618.3592082

[65] Yongquan Tao and Shengli Wu. 2014. Query Performance Prediction By Considering Score Magnitude and Variance Together. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, Jianzhong Li, Xiaoyang Sean Wang, Minos N. Garofalakis, Ian Soboroff, Torsten Suel, and Min Wang (Eds.). ACM, 1891–1894. https://doi.org/10.1145/2661829.2661906

[66] John W. Tukey. 1949. Comparing Individual Means in the Analysis of Variance. *Biometrics* 5, 2 (1949), 99–114. http://www.jstor.org/stable/3001913

[67] C. J. van Rijsbergen. 1979. *Information Retrieval*. Butterworth.

[68] Ellen Voorhees. 2005. Overview of the TREC 2004 Robust Retrieval Track. https://doi.org/10.6028/NIST.SP.500-261

[69] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[70] Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* 1, 6 (1945), 80–83. http://www.jstor.org/stable/3001968

[71] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=zeFrfgyZln

[72] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=zeFrfgyZln

[73] Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. 2016. aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi (Eds.). ACM, 287–296. https://doi.org/10.1145/2983323.2983818

[74] Hamed Zamani, W. Bruce Croft, and J. Shane Culpepper. 2018. Neural Query Performance Prediction using Weak Supervision from Multiple Signals. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018,*

Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 105–114. https://doi.org/10.1145/3209978.3210041

[75] Oleg Zendel, Anna Shtok, Fiana Raiber, Oren Kurland, and J. Shane Culpepper. 2019. Information Needs, Queries, and Query Performance Prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 395–404. https://doi.org/10.1145/3331184.3331253

[76] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (<conf-loc>, <city>Virtual Event</city>, <country>Canada</country>, </conf-loc>) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1503–1512. https://doi.org/10.1145/3404835.3462880

[77] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense Text Retrieval based on Pretrained Language Models: A Survey. *CoRR* abs/2211.14876 (2022). https://doi.org/10.48550/ARXIV.2211.14876 arXiv:2211.14876

[78] Ying Zhao, Falk Scholer, and Yohannes Tsegay. 2008. 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 (Lecture Notes in Computer Science, Vol. 4956)*, Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White (Eds.). Springer, 52–64. https://doi.org/10.1007/978-3-540-78646-7_8

[79] Yun Zhou and W. Bruce Croft. 2007. Query performance prediction in web search environments. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando (Eds.). ACM, 543–550. https://doi.org/10.1145/1277741.1277835