# Query Performance Prediction using Dimension Importance Estimators

Guglielmo Faggioli[1] [0000−0002−5070−2049], Nicola Ferro[1] [0000−0001−9219−6239], Raffaele Perego[2] [0000−0001−7189−4724], and Nicola Tonellotto[3] [0000−0002−7427−1001]

[1] University of Padua, Padua, Italy
[2] ISTI-CNR, Pisa, Italy
[3] University of Pisa, Pisa, Italy

**Abstract.** Query Performance Prediction (QPP) tends to fall short when predicting the performance of dense Information Retrieval (IR) systems. Therefore, the research community is investigating QPP approaches designed to synergize with this class of state-of-the-art IR models. At the same time, recent advances concerning dense IR have shown that we can improve the retrieval performance by projecting embeddings in a (query-wise) optimal linear subspace of the dense representation space. The Dimension IMportance Estimation (DIME) framework was proposed to identify such optimal subspaces on a query-by-query basis. In this paper, we illustrate how to design QPP models that rely on measuring the alignment between the query and document representations and the optimal DIME dimensions, based on the hypothesis that good alignment indicates better retrieval performance. We experimentally evaluate the proposed QPPs, showing that our approach outperforms the state-of-the-art when predicting the performance of two commonly used dense encoders, Contriever and TAS-B, on two popular TREC collections, Deep Learning 2019 and 2020.

## 1 Introduction

*Query Performance Prediction (QPP)* consists in determining the performance of a *Information Retrieval (IR)* system in the absence of human-made relevance judgments [3, 19]. This task allows us to determine which queries are likely to fail; it can be used to select the best IR system to answer a specific query; or as a signal to combine multiple systems [3]. Most QPP approaches were designed to be used with IR systems relying on lexical matching and they still struggle to achieve optimal performance when used to predict the performance of dense IR models, which employ semantic signals instead [2, 9, 10, 14, 15].

On a different research line, Faggioli et al. [13] observed that, when using a dense IR models, only some (query-dependent) dimensions of the latent embedding space contribute positively to the optimal ranking, while others are useless or, even, detrimental. As a consequence, by projecting the query and documents on a subspace that includes only the useful dimensions, it is possible to improve retrieval performance. Therefore, Faggioli et al. postulated the *manifold clustering hypothesis*, which states that "*High-dimensional representations of queries and documents relevant to them often lie in a query-dependent lower-dimensional manifold of the representation space*". They propose a novel class of estimators, called *Dimension IMportance Estimators (DIMEs)*, to determine a linear subspace of the original embedding space where the retrieval is more effective by predicting the importance of each dimension.

In this work, we apply the DIME framework to the QPP task, by proposing a novel family of predictors that exploit information on the importance of the dimensions of an embedding space to formulate their predictions. In detail, we first determine which dimensions are relevant to a query. This information is then used to instantiate a set of heuristics that measure the alignment between the query and document representations with such relevant dimensions. The underlying hypothesis is that if these representations are already well aligned with the important dimensions, we can expect effective retrieval and predict high performance. Vice-versa, a poor alignment with the optimal dimensions suggests a suboptimal retrieval that can lead to low performance. Our experimentation suggests that our hypothesis holds. By measuring the correlation between the representations of the retrieved documents with the importance of the dimensions determined using a DIME, we can effectively predict the performance of a set of state-of-the-art IR systems. More in detail, we can overcome the current state-of-the-art QPPs when predicting the performance for two popular dense encoders, Contriever [22] and TAS-B [21], on two TREC collections, Deep Learning 2019 and 2020.

The remainder of this work is organized as follows: Section 2 reports the related work. Section 3 introduces the DIME framework and describes the QPP employed in this work. Section 4 reports our experimental evaluation, while in Section 5, we draw our conclusions and outline the future work.

## 2 Related Work

*Dense IR.* Three main categories of Dense IR systems have been developed: bi-encoders, cross-encoders, and multivector or late-interaction models [39]. In this work, we focus on bi-encoders, and in particular, we consider symmetric approaches that use the same encoder to project queries and documents within the same latent space. Bi-encoders (a.k.a dual-encoders) rely on two separate (but possibly identical) neural networks to embed queries and documents within a latent space [39]. Once the representation for both documents and a query has been computed, the documents are ranked according to the inner product between the representation of the query and the one of the document. In recent years, several such models have been released, e.g., STAR [38], ANCE [34], Contriever [22], TAS-B [21] and they are considered state-of-the-art solutions.

*Traditional QPP.* Depending on the features they rely upon, traditional QPPs are divided into pre- and post-retrieval predictors [3, 19, 20]. The former rely 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 [24, 40]. On the other hand, post-retrieval predictors infer their predictions by taking as input also the ranked list of documents in response to the query. Traditional QPPs, such as Clarity [6], *Weighted Information Gain (WIG)* [41], *Normalized Query Commitment (NQC)* [30], and *Score Magnitude and Variance (SMV)* [32], are designed to operate on traditional IR methods, such as BM25 [25] or language models. The characteristic of such models is that they rely on exact lexical match, i.e., the presence of the query terms in the document was considered a relevance signal. On the contrary, neural IR, including dense approaches, rely on semantic matching: therefore novel QPP are needed to synergize at best with such systems [15].

*QPP for Neural IR.* While over time many QPP systems that employ semantic signals were developed [1, 23, 26, 36], most of them employ such signals to predict the performance of lexical IR models, making them less effective in predicting the performance of dense IR models that rely on semantic signals to determine which documents are relevant. Among the QPPs specifically designed to carry out QPP for neural IR, Hashemi et al. [18] introduce the Non-Factoid Question Answering QPP (NQAQPP). This approach combines retrieval scores, query lexical features, and both query and answer lexical features within a deep neural network framework to address the prediction of the performacne in the Non-Factoid Question Answering task. Hashemi et al.'s work is also one of the earliest to evaluate the effectiveness of QPP on neural IR models, specifically testing it on aNMM [35], and Conv-KNRM [8]. They note a significant gap in predictive accuracy between BM25 and neural IR models, which they attribute to the distinct score distributions produced by neural models. In a recent study, Faggioli et al. [15] examine the ability of traditional QPP techniques to predict the performance of neural IR systems. Through a series of experiments, they find a notable decline in the performance of existing QPP models when applied to neural IR systems. This trend persists even when using BERT-QPP as a predictive model for neural IR. This pattern is also observed by Datta et al. [9, 10]. Singh et al. [31] propose an innovative QPP that uses an auxiliary pairwise ranker (DuoT5) as an unsupervised QPP model. This model measures how often the ranking produced by the IR system aligns with the pairwise comparisons made by the auxiliary model. Meanwhile, Faggioli et al. [12] leverage the geometric properties of dense representations for performance prediction in conversational search. They develop the *Hypervolume (HV)* predictor, which calculates the volume of the axis-aligned bounding box that encompasses the top-$k$ retrieved documents and the query. Arabzadeh et al. [2] introduced a strategy specifically tailored for dense IR systems. The predictor they proposed, known as *DenseQPP (DQPP)*, operates by assessing the similarity between the original ranked list and the ranked list generated after the query is perturbed with appositely crafted Gaussian noise. On a similar line, the approach proposed by Salamat et al. [28] employs the dense representation to predict the query difficulty. More recently, Faggioli et al. [11] propose the *Dense-Centroid (DC)* framework.

In this case, the underlying idea is to adapt traditional QPP to the dense IR systems. To do so, the centroid of the corpus is used as a proxy representation of the collection. Such representation is then used to replace the language model of the collection used by several classic predictors such as WIG, NQC and SMV.

## 3 Methodology

We introduce here the notation and background on DIMEs as proposed in [13] and we describe the proposed QPP developed in this work.

### 3.1 Background on Dimension IMportance Estimation

Consider a query $q$ for which the user wants to retrieve documents from a corpus $\mathcal{D}$. We define $\mathcal{R}(q, \mathcal{D}; \nu)$ the ranked list produced by the IR model $\nu$ in response to $q$. Assuming relevance judgments are available, we can compute a measure $\mathcal{M}(\mathcal{R}(q, \mathcal{D}; \nu))$ that takes as input the list of retrieved documents and outputs a performance score. This work focuses on dense IR models employing an encoder $\phi$ to project the text (i.e., the query and the documents) into a $d$ dimensional embedding space $\mathbb{R}^d$. The encoder is often a neural network trained with the objective of maximising the dot product between a query and corresponding relevant documents. Therefore, the score assigned to a document $D$ in response to a query $q$ is $s(q, D) = \langle \phi(q), \phi(D) \rangle$. With an abuse of notation, we call $\mathcal{R}(q, \mathcal{D}; \phi, \langle\rangle)$ the ranker that takes in input the query and the corpus, embeds them in the $d-$dimensional space using $\phi$, computes the dot product $\langle\rangle$ between the query and each document, and ranks the documents accordingly. We define the *masked dot*, $\langle \vec{v}, \vec{w} \rangle_{\setminus\{i\}} = \sum_{j=1;j\neq i}^{d} v_j \cdot w_j$, the dot product between two arbitrary vectors $\vec{v}$ and $\vec{w}$, where the $i$-th dimension is ignored. Faggioli et al. [13] experimentally showed that, given a query $q$, it exists a set $\delta \subset \{1, ..., d\}$ s.t.

$$\mathcal{M}(\mathcal{R}(q, \mathcal{D}; \phi, \langle\rangle)) < \mathcal{M}(\mathcal{R}(q, \mathcal{D}; \phi, \langle\rangle_{\setminus\delta})) \tag{1}$$

In other terms, given an encoder $\phi$ and a query $q$ there is a set of dimensions that are harmful to the retrieval: by simply discarding those dimensions when computing the dot product, it is possible to improve the quality of the retrieval[4]. Faggioli et al. [13] showed that the improvement depends on the collection considered and on the encoder $\phi$, reaching peaks as big as +0.30 nDCG@10 points, moving from 0.5 to 0.8. Furthermore, they observe that the optimal dimensions are query-dependent, with each query being optimized by a different set of dimensions. While discarding some dimensions allows astonishing retrieval improvements, e.g. up to +73.4% in nDCG@10 when using TAS-B for RB '04 queries with 40% dimensions, determining which dimensions are optimal is not

[4] Faggioli et al. [13] conjecture that the optimal subspace can be any subspace of the original embedding space but, to make the problem tractable, they focus only linear subspaces where some dimensions are removed.

trivial. Therefore, Faggioli et al. [13] propose a novel class of models, called "*Dimension IMportance Estimators (DIMEs)*", that rely on heuristics to determine which dimensions to preserve/remove. A DIME is a function $u : (\mathbb{R}^d; \theta) \rightarrow \mathbb{R}^d$ that takes in input a representation of a query $\phi(q) \in \mathbb{R}^d$ – and possibly some additional parameters $\theta$ – and outputs a vector $\vec{r} \in \mathbb{R}^d$ that describes how much each dimension is *important*. The relation between the importance $r_i$ and $r_j$ of respectively dimensions $i$ and $j$ is defined as follows:

$$\mathcal{M}(\mathcal{R}(q, \mathcal{D}; \phi, \langle\rangle_{\setminus\{i\}}) < \mathcal{M}(\mathcal{R}(q, \mathcal{D}; \phi, \langle\rangle_{\setminus\{j\}})) \implies r_i > r_j \tag{2}$$

In other terms, the $i$-th dimension is more important than the $j$-th ($r_i > r_j$) if the DIME $u$ considers it more likely the result will be worse by removing $i$ instead of $j$ when computing the dot product.

The most effective DIMEs, according to Faggioli et al., are the *Active Feedback* DIME ($u^{REL}$) and the *LLM Pseudo Relevant Feedback* DIME ($u^{LLM}$). The former employs a relevant document $D^R$ (e.g., obtained by inspecting a query log or the user's clicks) and the importance of a dimension is defined as follows:

$$u_i^{REL}(q; D^R) = \phi(q)_i \cdot \phi(D^R)_i, \tag{3}$$

where $\phi(q)_i$ and $\phi(D^R)_i$ are respectively the $i$-th dimensions of the query and relevant document's representations. Similarly, $u^{LLM}$ is based on generating a pseudo-relevant document $LLM(q)$ by feeding the query to an LLM. The dimension importance in this case is defined as:

$$u_i^{LLM}(q; LLM) = \phi(q)_i \cdot \phi(LLM(q))_i \tag{4}$$

Faggioli et. al. [13] employ the proposed DIMEs by selecting the $l$ most important dimensions with a fixed $l$.

### 3.2 The Proposed Query Performance Predictors

The predictors proposed here comprise an `input` and an `aggregator` component:

- The `input` describes which input is used to compute the prediction. There are three options: the query vector, the document vectors, or the interaction vectors (the Hadamard product between the query and document vectors).
- The `aggregator` component describes how to combine the input vectors with the DIME.

All the predictors are instantiated by first inputting a query and computing the dimension importance using a DIME. Such DIME values are combined with the `input` vectors using the `aggregator` function. A predictor can be described as `aggregator` (`input`; DIME)). In terms of notation, the predictors are identified by ⟨`input ID`⟩-⟨`aggregator ID`⟩-⟨`DIME ID`⟩; for example, D-C-LLM indicates the QPP that uses documents as input (D), relies on the correlation `aggregator` (C) and estimates the dimension importance using $u^{LLM}$ as DIME. We now describe each class of components in more detail.

**The `input` component.** Our predictors can be based on a single vector (e.g., they can consider only the representation of the query) or can employ multiple vectors. In our framework, in the former case, a statistic is computed for each vector to formulate the prediction. In the latter case, each vector's statistic is computed individually and then aggregated by computing the mean. More in detail, let us call $\alpha(v)$ the score that an `aggregator` component assigns to a vector $v$. For the moment, we consider $\alpha : \mathbb{R}^d \rightarrow \mathbb{R}$ an arbitrary function that takes in input a vector and outputs a real number. Based on this definition, we can define three `input` components: *"Q-"* (Query), *"D-"* (Document), and *"I-"* (Interaction). Given an arbitrary `aggregator` function $\alpha$, the `input` components are defined as follows:

- "Q-" `input`: given a query $q$, the prediction is $Q\text{-}\alpha(q) = \alpha(\phi(q))$ (i.e., the `aggregator`, directly applied on the query vector).
- "D-" `input`: given $k$ documents $D_1$, ..., $D_k$, the prediction is $D\text{-}\alpha(D_1, ..., D_k) = \sum_{i=1}^{k} \frac{\alpha(\phi(D_i))}{k}$. In this case, the `aggregator` is applied separately on each document vector and then averaged.
- "I-" `input`: given a query $q$ and a $k$ documents $D_1$, ..., $D_k$, the prediction is $I\text{-}\alpha(q, D_1, ..., D_k) = \sum_{i=1}^{k} \frac{\alpha(\phi(q) \circ \phi(D_i))}{k}$, where $\circ$ represents the Hadamard product (i.e., the element-wise multiplication between the two vectors).

Since the predictors based on the Q- `input` employ only the representation of the query and do not require access to the retrieved list of documents, they can be considered pre-retrieval predictors. On the contrary, predictors based on D- and I- `input` employ the top-$k$ documents retrieved, making them post-retrieval predictors. Additionally, notice that D- and I- predictors will have the number of documents considered $k$ as an additional hyper-parameter.

**The `aggregator` component**

*Negative Importance (NI) `aggregator`.* If a DIME considers a dimension to be detrimental, i.e., it would be better to remove it to increase the retrieval performance, this dimension should be as small as possible to obtain the best performance. Vice-versa, observing a high absolute value on such dimensions suggests non-effective retrieval. Therefore, the Negative Importance (NI) `aggregator` correlates the performance of the query with the inverse of the magnitude of the not important dimensions according to the DIME. We focus on the absolute value of the dimension: if the DIME would like to exclude it, the best case occurs when the absolute value is close to zero.

Let's call $\delta_l^- \subset \{1, ..., d\}$, with $|\delta_l^-| = l$, the set of $l$ dimensions having the smallest relevance score $r_i$ according to an arbitrary DIME. In this case, the `aggregator` function can be defined as follows:

$$\text{NI}(\vec{v}; l, \delta_l^-) = \frac{l}{\sum_{i \in \delta_l^-} \text{abs}(v_i)} \quad (5)$$

Where $\vec{v}$ is the `input` vector for which we want to compute the aggregation. As mentioned before, this value can be used to instantiate a predictor based on Q-, D-, or I- `input` (respectively Q-NI, D-NI, and I-NI). Notice that the NI `aggregator` function has $l$ as a hyperparameter.

*Positive Importance (PI)* ***aggregator****.* The second `aggregator` function, called the Positive Importance (PI) `aggregator`, associates good performance with vectors having a large absolute value on dimensions considered important by the DIME. It can be considered the opposite of the NI `aggregator`. In line with the NI `aggregator`, we define $\delta_l^+ \subset \{1, ..., d\}$, with $|\delta_l^+| = l$, the set of $l$ dimensions having the highest relevance score $r_i$ according to an arbitrary DIME. The `aggregator` function in this case is:

$$\mathrm{PI}(\vec{v}; l, \delta_l^+) = \frac{\sum_{i \in \delta_l^+} \mathrm{abs}(v_i)}{l} \tag{6}$$

As before, the predictor has $l$ as a hyperparameter. The predictors are called Q-PI, D-PI, and I-PI, depending on which vector $\vec{v}$ is fed in input.

*Ratio (R)* ***aggregator****.* This `aggregator` computes the product between NI and PI. It is based on the same rationale as the previous two: large important dimensions are a positive signal, while large detrimental dimensions are a negative one. This `aggregator` is defined as follows:

$$\mathrm{R}(\vec{v}; l_1, l_2, \delta_{l_1}^+, \delta_{l_2}^-) = \mathrm{PI}(\vec{v}; l_1, \delta_{l_1}^+) \cdot \mathrm{NI}(\vec{v}; l_2, \delta_{l_2}^-) = \frac{\sum_{i \in \delta_{l_1}^+} \mathrm{abs}(v_i)}{\sum_{i \in \delta_{l_2}^-} \mathrm{abs}(v_i)} \cdot \frac{l_2}{l_1} \tag{7}$$

Notice that, from a technical point of view, the two hyper-parameters $l_1$ and $l_2$ can be considered independent. To reduce the number of possible combinations to be tested, align it with other solutions, and make the approach more stable, we set $l_1 = l$ and $l_2 = d - l$, reducing the hyper-parameters to only $l$. In other terms, the first $l$ dimensions are considered useful, while the remaining $d - l$ dimensions are deemed detrimental. As in the other cases, the three variants of this approach are called Q-R, D-R, and I-R.

*Alignment (A)* ***aggregator****.* The Alignment `aggregator` measures the cosine similarity between the representation fed as input with a second vector constructed using the dimensions considered important by the DIME. More in detail, we call $\delta_l^+$ the $l$ most relevant dimensions according to the DIME. We then construct a masking vector $\vec{m}$ s.t. $m_i = 1$ if $i \in \delta_l^+$, 0 otherwise. Then the score is computed as:

$$\mathrm{A}(\vec{v}; l, \delta_l^+) = \frac{\langle \mathrm{abs}(\vec{v}), \vec{m} \rangle}{|\vec{m}| |\mathrm{abs}(\vec{v})|} \tag{8}$$

Similarly to the R `aggregator`, also in this case, we have a contribution both from negative and positive dimensions. Still, while the contribution of the positive dimensions is explicit through the dot product, the negative dimensions play a role in changing the normalisation value $|\vec{m}|$. As before, $l$ is a hyperparameter.

*Correlation (C)* `aggregator`*.* Our final `aggregator` measures the correlation between the input vector $\vec{v}$ and the importance DIME vector $\vec{r}$:

$$C(\vec{v}; \vec{r}) = \text{corr}(\text{abs}(\vec{v}), \vec{r}) \tag{9}$$

Multiple correlation functions can be used and in our experiments, we consider Kendall's $\tau$ and Pearson's $\rho$ correlations. More in detail, we do not select explicitly the correlation function but we treat it as a hyperparameter, choosing the optimal one according to the validation procedure described in Section 4.

## 4 Experimental Evaluation

### 4.1 Experimental Setup

In our experiments, we use our predictors to predict the performance of two IR models, Contriever [22] and TAS-B [21], on two collections, TREC Deep Learning 2019 (DL' 19) [5] and TREC Deep Learning 2020 (DL' 20) [4], and with respect to two evaluation measures P@10 and nDCG@10. We consider 5 state-of-the-art baselines, Clarity [6], $n(\sigma_{\%})$ [7],WIG [41], NQC [30], and SMV [32] as well as their *Utility Estimation Framework (UEF)* [29] enhanced counterparts. We consider a state-of-the-art QPP for dense models (DCWIG [14]) and BERTQPP [1]. To optimize the QPP hyperparameters, we adopt the well-known two-fold cross-validation procedure in which queries are disjointly divided into two folds and, in turn, a fold is used to choose the hyperparameters and the other as a test set. The final performance is averaged across 30 repetitions, as commonly done in this setting [12, 30, 36, 37]. In terms of QPP evaluation measures, we report Pearson's $\rho$ and Kendall's $\tau$ between the actual and predicted performance. Additionally, instead of sMARE, we report 1-sMARE [16, 17], so that, in line with Pearson's $\rho$ and Kendall's $\tau$, bigger values indicate more favourable results. All the results have been validated statistically using ANOVA [27] and Tukey's honestly significant difference post-hoc comparison [33] with significance at 0.05 to correct for multiple comparisons. For all predictors, we validate the number of documents considered $k \in \{5, 10, 25, 50, 100, 250, 500\}$. For the number of important dimensions $l$, we validate its value by considering $l \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

### 4.2 Determining the Optimal QPP `input` and `aggregator` Function

We start our analysis by considering the aggregated performance based on different `input` and `aggregator` components proposed in this paper.

Figure 1 reports the performance of the different DIME-based `input` aggregated across datasets, IR models, `aggregator` functions, and IR measures. The first pattern that we observe is that the average performance for the query (Q-) `input` is always worse than both documents (D-) and interactions (I-) `input` components. This is somewhat expected from the QPP perspective: the Q- approaches are pre-retrieval: i.e., they consider only the query. Pre-retrieval

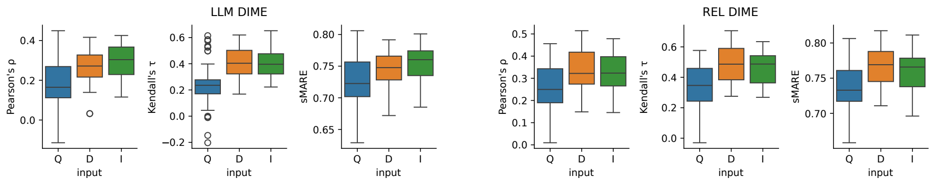


Fig. 1: Performance of the three `input`. These results are aggregated across evaluation measures, datasets, `aggregator` functions, and IR models. While Q is always the worst `input`, the relationship between D and I depends on the dime considered.

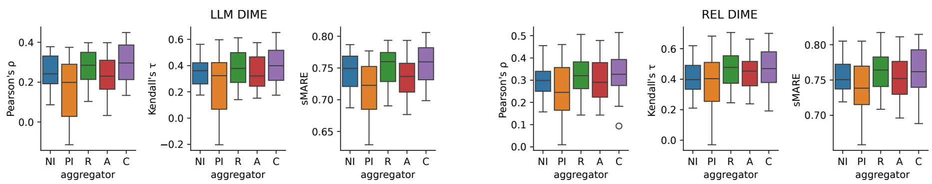


Fig. 2: Performance of the five `aggregator` components. These results are aggregated across evaluation measures, datasets, `input` components, and IR models. While NI and PI tend to be the worst performing. R and A behave similarly, while C is the optimal approach in all scenarios.

approaches are known to be typically less performing than their post-retrieval counterparts [3, 16, 19]: by using less information, they lack sufficient capacity to perform effectively. Furthermore, this sheds some light on the DIMEs themselves: since the representation of the query alone is less indicative of the performance, we can expect it to have a lesser impact on the performance itself. Conversely, documents play a much more prominent role. Concerning I- and D- behaviour, we highlight how typically D- has a narrower distribution, producing more stable predictions. Another interesting pattern is that with the LLM-based DIME (left) the D- `input` component is typically less effective than the I- `input` component. Conversely, for the Active Feedback DIME (right), D- and I- behave similarly (except for the variance of the performance). An explanation might be that, by taking a real relevant, the vocabulary/structure is similar to other relevant documents, making predictors based on the document representations more effective. Vice-versa, by generating the (pseudo-)relevant document with a language model, its structure and term distribution will likely differ from those of relevant documents, impairing the possibility of relying only on the alignment between their representations.

Figure 2 reports the boxplot across different `aggregator` approaches. As for the `input` boxplots, the boxplots report the distribution across IR systems, IR measures, collections and `input`. Figure 2 highlights how both the NI and PI `aggregator` tend to be the least effective: they do not have enough information to work properly. Indeed, predictors based on R, A and C perform better as they

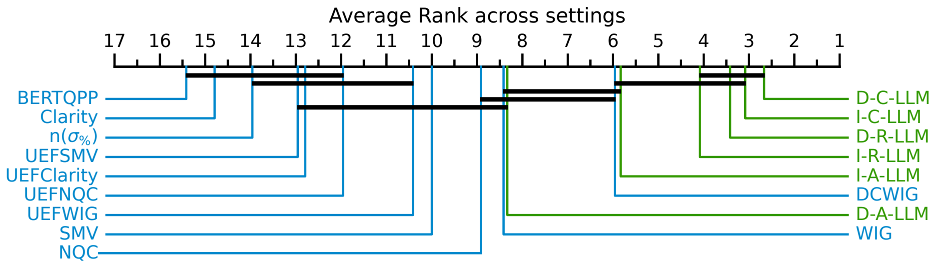


Fig. 3: Average rank across different experimental settings, excluding predictors based on Active Feedback DIME. In green and blue, predictors proposed here and baselines, respectively. Horizontal bars indicate statistically equivalent approaches according to a Wilcoxon signed-rank test.

combine both NI and PI information. the A `aggregator` behaves in line with the R `aggregator`. This pattern is explainable thanks to the fact that similarly to R `aggregator`, A `aggregator` considers both the NI and PI information. Overall, the best performing is the C `aggregator`: the additional information provided by the ordering/magnitude of the dimensions is effective in achieving better predictions.

### 4.3 Comparison With the State-of-the-Art

The previous section highlighted how the best `input` components are either D- or I-, while the best `aggregator` appear to be A, R and C. Therefore, in the remainder of this paper, we focus on the combinations of such approaches. Table 1 reports the performance of the current state-of-the-art approaches (top) compared to the proposed predictors based on either the LLM DIME (centre) or the Active Feedback DIME (bottom). Across all scenarios, predictors based on DIMEs are the most effective solutions. In general, the approaches based on the Active Feedback DIME (indicated with -REL) that employ a relevant document are more effective, regardless of the `input` and `aggregator` used. This makes sense considering that this DIME effectively employs a stronger relevance signal, compared to the LLM-based DIME, which uses only pseudo-relevance information. Nevertheless, with a few exceptions (e.g., WIG on DL' 19 or NQC on DL' 20 when predicting P@10 and evaluating with Pearson's $\rho$), the predictors employing the LLM-based DIME are capable of overcoming all the state of the art approaches. To predict Precision, the most effective solutions are those employing the Active Feedback DIME, using the set of retrieved documents as `input` and the Ratio `aggregator`. When it comes to predicting nDCG@10, the most effective solutions are either the one based on the Interaction `input`, Correlation `aggregator`, and the LLM-based DIME for DL' 19 and the approach based on the Document `input`, Correlation `aggregator`, and Active feedback DIME for DL' 20.

Table 1: Comparison of the predictors with state-of-the-art approaches. In bold and underlined, are the best and second-best approaches respectively. * indicates the set of statistically best systems using ANOVA and Tukey's post-hoc test.

| | Contriever | | | | | | TAS-B | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DL' 19 | | | DL' 20 | | | DL' 19 | | | DL' 20 | | |
| | K-$\tau$ | P-$\rho$ | sMARE | K-$\tau$ | P-$\rho$ | sMARE | K-$\tau$ | P-$\rho$ | sMARE | K-$\tau$ | P-$\rho$ | sMARE |
| | P@10 | | | | | | | | | | | |
| n($\sigma_{\%}$) | 0.182 | 0.134 | 0.718 | -0.059 | 0.009 | 0.645 | 0.244 | 0.288 | 0.741 | 0.120 | 0.197 | 0.698 |
| Clarity | 0.178 | 0.255 | 0.728 | -0.053 | -0.065 | 0.659 | 0.114 | 0.153 | 0.707 | -0.043 | -0.064 | 0.651 |
| SMV | 0.309 | 0.413 | 0.759 | 0.043 | 0.071 | 0.673 | 0.335 | 0.435 | 0.767 | 0.241 | 0.312 | 0.732 |
| NQC | 0.327 | 0.438 | 0.760 | 0.044 | 0.081 | 0.678 | 0.344 | 0.404 | 0.779 | 0.311 | 0.383 | 0.756 |
| WIG | 0.296 | 0.425 | 0.757 | 0.136 | 0.125 | 0.699 | 0.392 | 0.504 | 0.786 | 0.232 | 0.313 | 0.721 |
| UEFClarity | 0.302 | 0.413 | 0.753 | 0.023 | 0.029 | 0.676 | 0.289 | 0.398 | 0.749 | 0.103 | 0.064 | 0.696 |
| UEFNQC | 0.305 | 0.383 | 0.747 | 0.052 | -0.086 | 0.683 | 0.359 | 0.407 | 0.777 | 0.118 | 0.137 | 0.697 |
| UEFSMV | 0.280 | 0.345 | 0.744 | 0.041 | -0.195 | 0.687 | 0.346 | 0.391 | 0.770 | 0.095 | -0.091 | 0.702 |
| UEFWIG | 0.326 | 0.448 | 0.759 | 0.067 | 0.077 | 0.677 | 0.294 | 0.415 | 0.761 | 0.211 | 0.239 | 0.722 |
| BERTQPP | -0.005 | -0.021 | 0.659 | -0.097 | -0.141 | 0.634 | -0.097 | -0.206 | 0.658 | -0.124 | -0.170 | 0.633 |
| DCWIG | 0.380 | 0.538 | 0.779 | 0.245 | 0.383* | 0.720 | 0.388 | 0.493 | 0.776 | 0.336 | 0.403 | 0.768* |
| D-R-LLM | 0.398 | 0.612 | 0.790 | 0.279* | 0.342 | 0.760* | 0.346 | 0.527 | 0.774 | 0.360 | 0.487* | 0.777* |
| D-A-LLM | 0.358 | 0.573 | 0.763 | 0.165 | 0.345 | 0.700 | 0.266 | 0.408 | 0.745 | 0.259 | 0.323 | 0.746 |
| D-C-LLM | 0.416 | 0.620 | 0.792 | 0.302* | 0.416* | **0.765*** | 0.395 | 0.505 | 0.782 | 0.367 | 0.440 | 0.782* |
| I-R-LLM | 0.384 | 0.608 | 0.776 | 0.284* | 0.285 | 0.760* | 0.367 | 0.419 | 0.785 | 0.328 | 0.453 | 0.768* |
| I-A-LLM | 0.355 | 0.453 | 0.771 | 0.116 | 0.232 | 0.685 | 0.383 | 0.493 | 0.775 | 0.354 | 0.459 | 0.773* |
| I-C-LLM | 0.425 | 0.621 | 0.787 | 0.291* | 0.316 | 0.750* | 0.384 | 0.510 | 0.785 | 0.317 | 0.373 | 0.759 |
| D-R-REL | **0.484*** | **0.706*** | **0.817*** | 0.300* | 0.378 | 0.748 | 0.506* | **0.671*** | 0.814* | **0.435*** | **0.551*** | **0.783*** |
| D-A-REL | 0.442* | 0.663* | 0.787 | 0.202 | 0.342 | 0.720 | 0.370 | 0.506 | 0.778 | 0.230 | 0.290 | 0.744 |
| D-C-REL | 0.453* | 0.700* | 0.807* | **0.305*** | **0.439*** | 0.750 | **0.514*** | 0.627* | **0.815*** | 0.317 | 0.383 | 0.756 |
| I-R-REL | 0.449* | 0.633 | 0.799 | 0.239 | 0.298 | 0.730 | 0.446 | 0.594 | 0.795 | 0.284 | 0.332 | 0.745 |
| I-A-REL | 0.373 | 0.508 | 0.776 | 0.146 | 0.284 | 0.696 | 0.478* | 0.623* | 0.811* | 0.397* | 0.496* | 0.777* |
| I-C-REL | 0.434* | 0.629 | 0.801* | 0.279* | 0.364 | 0.738 | 0.429 | 0.610* | 0.794 | 0.292 | 0.341 | 0.748 |
| | nDCG@10 | | | | | | | | | | | |
| n($\sigma_{\%}$) | 0.331 | 0.429 | 0.753 | 0.074 | 0.114 | 0.704 | 0.210 | 0.291 | 0.731 | 0.197 | 0.303 | 0.726 |
| Clarity | 0.199 | 0.299 | 0.722 | -0.015 | -0.019 | 0.657 | 0.159 | 0.248 | 0.717 | -0.045 | -0.014 | 0.653 |
| SMV | 0.230 | 0.351 | 0.737 | 0.108 | 0.199 | 0.701 | 0.175 | 0.214 | 0.712 | 0.196 | 0.346 | 0.721 |
| NQC | 0.238 | 0.403 | 0.742 | 0.102 | 0.196 | 0.701 | 0.183 | 0.230 | 0.714 | 0.202 | 0.335 | 0.718 |
| WIG | 0.219 | 0.379 | 0.726 | 0.104 | 0.239 | 0.696 | 0.190 | 0.365 | 0.726 | 0.174 | 0.279 | 0.711 |
| UEFClarity | 0.254 | 0.300 | 0.745 | -0.045 | -0.098 | 0.655 | 0.197 | 0.261 | 0.722 | -0.032 | -0.105 | 0.669 |
| UEFNQC | 0.233 | 0.230 | 0.736 | -0.023 | -0.117 | 0.669 | 0.207 | 0.204 | 0.730 | 0.004 | -0.072 | 0.667 |
| UEFSMV | 0.204 | 0.120 | 0.728 | -0.001 | -0.226 | 0.670 | 0.214 | 0.209 | 0.727 | -0.009 | -0.206 | 0.667 |
| UEFWIG | 0.266 | 0.316 | 0.747 | -0.015 | -0.075 | 0.668 | 0.211 | 0.305 | 0.725 | 0.004 | 0.003 | 0.676 |
| BERTQPP | 0.165 | 0.156 | 0.724 | 0.068 | 0.137 | 0.684 | 0.040 | 0.012 | 0.685 | 0.025 | 0.087 | 0.669 |
| DCWIG | 0.299 | 0.499 | 0.750 | 0.259 | 0.415 | 0.747 | 0.171 | 0.169 | 0.713 | 0.178 | 0.253 | 0.726 |
| D-R-LLM | 0.281 | 0.501 | 0.751 | 0.286 | 0.370 | 0.748 | 0.218 | 0.340 | 0.729 | 0.326 | 0.463 | 0.768 |
| D-A-LLM | 0.285 | 0.557 | 0.752 | 0.192 | 0.320 | 0.720 | 0.033 | 0.171 | 0.677 | 0.179 | 0.230 | 0.715 |
| D-C-LLM | 0.285 | 0.536 | 0.753 | 0.237 | 0.400 | 0.743 | 0.212 | 0.301 | 0.722 | 0.331 | 0.489 | 0.775* |
| I-R-LLM | 0.377 | 0.575 | 0.794* | 0.176 | 0.266 | 0.722 | 0.286* | 0.353 | 0.760* | 0.252 | 0.386 | 0.736 |
| I-A-LLM | 0.303 | 0.490 | 0.753 | 0.182 | 0.270 | 0.715 | 0.193 | 0.288 | 0.717 | 0.262 | 0.393 | 0.746 |
| I-C-LLM | 0.425* | **0.652*** | **0.801*** | 0.208 | 0.358 | 0.728 | **0.306*** | 0.374 | **0.763*** | 0.242 | 0.399 | 0.737 |
| D-R-REL | 0.339 | 0.571 | 0.772 | 0.365* | 0.469* | 0.783* | 0.225 | **0.494*** | 0.743* | 0.359 | 0.515 | 0.777* |
| D-A-REL | **0.429*** | 0.638* | 0.793* | 0.277 | 0.443 | 0.744 | 0.149 | 0.361 | 0.715 | 0.279 | 0.401 | 0.752 |
| D-C-REL | 0.336 | 0.588* | 0.773 | **0.387*** | 0.520* | **0.794*** | 0.197 | 0.445* | 0.728 | **0.412*** | **0.596*** | **0.790*** |
| I-R-REL | 0.287 | 0.507 | 0.761 | 0.356* | 0.400 | 0.776 | 0.226 | 0.360 | 0.732 | 0.302 | 0.486 | 0.747 |
| I-A-REL | 0.398* | 0.578 | 0.775 | 0.266 | 0.405 | 0.753 | 0.229 | 0.478* | 0.730 | 0.340 | 0.521 | 0.767 |
| I-C-REL | 0.311 | 0.537 | 0.765 | 0.380* | **0.526*** | 0.792* | 0.183 | 0.342 | 0.711 | 0.363* | 0.512 | 0.766 |

Since the Active Feedback-based DIME employ one relevant document, we also report in Figure 3 the average rank across different experimental settings for the predictors, excluding the ones based on the Active Feedback DIME. We can observe how predictors based on the LLM DIME are ranked, on average, above all the baseline predictors. In particular, the D-C-LLM predictor is on average ranked the highest (average rank 2.6). Nevertheless, the other predictors based on Ratio and Correlation `aggregator` (I-C-LLM, D-R-LLM, and I-R-LLM), are statistically equivalent (according to the Wilcoxon signed rank test) to the best, followed by the predictors based on the Alignment `aggregator` and DCWIG. The first four approaches have statistically significantly higher ranks than any baseline.

Researchers and practitioners interested in using the DIME-base predictors should consider the following:

- While using only the query representation as input (Q-) is suboptimal, the document (D-) and interaction (I-) inputs exhibit comparable results: the practitioner can validate the `input` depending on their setting.
- Approaches based on the Alignment (-A-), Negative Importance (-NI-), and Positive importance (-PI-) should be avoided, as their performance is suboptimal compared to the approaches based on Ratio (-R-) and Correlation (-C-). Similarly to the `input`, the optimal `aggregator` between Ratio and Correlation should be validated.
- If the practitioner has access to at least one relevant document for the query, the approaches based on the Active Feedback should be favoured (-REL). Nevertheless, predictors relying on the LLM-based DIME (-LLM) still overcome the current state-of-the-art performance.

## 5 Conclusion

In this work, we investigated how to employ DIMEs to carry out QPP. In particular, DIMEs are a class of models meant to determine the (query-specific) importance of each dimension in a latent embedding space used for dense IR. The predictors proposed in this work rely on measuring the alignment between the vectors involved in the retrieval and the importance estimated according to a DIME. The proposed QPPs can be instantiated with different inputs (the query, the documents, or their interaction vectors) and rely on different aggregations of such inputs. The most effective predictors are those based on either the documents or interaction representations that compute the correlation between such vectors and the dimension importance. The proposed approaches remarkably outperform the current state-of-the-art to predict the performance for two well-known dense models (Contriever and TAS-B) on two collections, DL' 19 and DL' 20. Among future work, we are interested in applying other DIME to instantiate our predictors as well as to develop QPP models that can serve as DIME themselves, for example, by providing insight on how each dimension contributes to the predicted performance of the system.

**Acknowledgement.** This work was partially supported by the Spoke "FutureHPC & BigData" of the ICSC – Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing funded by the Italian Government, the PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI" funded by the European Union (EU) under the NextGeneration EU programme, and by the Horizon Europe RIA "Extreme Food Risk Analytics" (EFRA) funded by the European Commission under the NextGeneration EU programme grant agreement n. 101093026. We acknowledge also the FoReLab and CrossLab projects (Departments of Excellence), the NEREO PRIN project (Research Grant no. 2022AEFHAZ) and the CAMEO PRIN project (Research Grant no. 2022ZLL7MW) funded by the Italian Ministry of Education and Research (MUR).

## References

[1] Arabzadeh, N., Khodabakhsh, M., Bagheri, E.: BERT-QPP: contextualized pre-trained transformers for query performance prediction. In: Demartini, G., Zuccon, G., Culpepper, J.S., Huang, Z., Tong, H. (eds.) CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021, pp. 2857–2861, ACM (2021), https://doi.org/10.1145/3459637.3482063, URL `https://doi.org/10.1145/3459637.3482063`

[2] Arabzadeh, N., Rad, R.H., Khodabakhsh, M., Bagheri, E.: Noisy perturbations for estimating query difficulty in dense retrievers. In: Frommholz, I., Hopfgartner, F., Lee, M., Oakes, M., Lalmas, M., Zhang, M., Santos, R.L.T. (eds.) Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023, pp. 3722–3727, ACM (2023), https://doi.org/10.1145/3583780.3615270, URL `https://doi.org/10.1145/3583780.3615270`

[3] Carmel, D., Yom-Tov, E.: Estimating the Query Difficulty for Information Retrieval. Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgan & Claypool Publishers (2010), https://doi.org/10.2200/S00235ED1V01Y201004ICR015, URL `https://doi.org/10.2200/S00235ED1V01Y201004ICR015`

[4] Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the TREC 2020 deep learning track. CoRR **abs/2102.07662** (2021), URL `https://arxiv.org/abs/2102.07662`

[5] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the TREC 2019 deep learning track. CoRR **abs/2003.07820** (2020), URL `https://arxiv.org/abs/2003.07820`

[6] Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: Järvelin, K., Beaulieu, M., Baeza-Yates, R.A., Myaeng, S. (eds.) SIGIR 2002: Proceedings of the 25th Annual International ACM

SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland, pp. 299–306, ACM (2002), https://doi.org/10.1145/564376.564429, URL https://doi.org/10.1145/564376.564429

[7] Cummins, R., Jose, J., O'Riordan, C.: Improved query performance prediction using standard deviation. In: Ma, W., Nie, J., Baeza-Yates, R., Chua, T., Croft, W.B. (eds.) Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011, pp. 1089–1090, ACM (2011), https://doi.org/10.1145/2009916.2010063, URL https://doi.org/10.1145/2009916.2010063

[8] Dai, Z., Xiong, C., Callan, J., Liu, Z.: Convolutional neural networks for soft-matching n-grams in ad-hoc search. In: Chang, Y., Zhai, C., Liu, Y., Maarek, Y. (eds.) Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018, pp. 126–134, ACM (2018), https://doi.org/10.1145/3159652.3159659, URL https://doi.org/10.1145/3159652.3159659

[9] Datta, S., Ganguly, D., Mitra, M., Greene, D.: A relative information gain-based query performance prediction framework with generated query variants. ACM Trans. Inf. Syst. **41**(2), 38:1–38:31 (2023), https://doi.org/10.1145/3545112, URL https://doi.org/10.1145/3545112

[10] Datta, S., MacAvaney, S., Ganguly, D., Greene, D.: A 'pointwise-query, listwise-document' based query performance prediction approach. In: Proceedings of 45th international ACM SIGIR conference research development in information retrieval, pp. 2148—-2153 (2022), https://doi.org/10.1145/3477495.3531821, URL https://doi.org/10.1145/3477495.3531821

[11] Faggioli, G., Ferro, N., Mothe, J., Raiber, F.: 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, pp. 388–391, Springer (2023), https://doi.org/10.1007/978-3-031-28241-6_42, URL https://doi.org/10.1007/978-3-031-28241-6_42

[12] Faggioli, G., Ferro, N., Muntean, C., Perego, R., Tonellotto, N.: 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 (2023), https://doi.org/https://doi.org/10.1145/3539618.3591625

[13] Faggioli, G., Ferro, N., Perego, R., Tonellotto, N.: Dimension importance estimation for dense information retrieval. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC,

USA, ACM (2024), https://doi.org/10.1145/3626772.3657691, URL https://doi.org/10.1145/3626772.3657691

[14] Faggioli, G., Formal, T., Lupart, S., Marchesin, S., Clinchant, S., Ferro, N., Piwowarski, B.: Towards query performance prediction for neural information retrieval: Challenges and opportunities. In: Yoshioka, M., Kiseleva, J., Aliannejadi, M. (eds.) Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2023, Taipei, Taiwan, 23 July 2023, pp. 51–63, ACM (2023), https://doi.org/10.1145/3578337.3605142, URL https://doi.org/10.1145/3578337.3605142

[15] Faggioli, G., Formal, T., Marchesin, S., Clinchant, S., Ferro, N., Piwowarski, B.: 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, pp. 1–18 (2023), https://doi.org/10.48550/ARXIV.2302.09947, URL https://arxiv.org/abs/2302.09947

[16] Faggioli, G., Zendel, O., Culpepper, J.S., Ferro, N., Scholer, F.: An enhanced evaluation framework for query performance prediction. In: Hiemstra, D., Moens, M., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) 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, pp. 115–129, Springer (2021), https://doi.org/10.1007/978-3-030-72113-8_8, URL https://doi.org/10.1007/978-3-030-72113-8_8

[17] Faggioli, G., Zendel, O., Culpepper, J.S., Ferro, N., Scholer, F.: smare: a new paradigm to evaluate and understand query performance prediction methods. Inf. Retr. J. **25**(2), 94–122 (2022), https://doi.org/10.1007/s10791-022-09407-w, URL https://doi.org/10.1007/s10791-022-09407-w

[18] Hashemi, H., Zamani, H., Croft, W.B.: Performance prediction for non-factoid question answering. In: Fang, Y., Zhang, Y., Allan, J., Balog, K., Carterette, B., Guo, J. (eds.) Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2019, Santa Clara, CA, USA, October 2-5, 2019, pp. 55–58, ACM (2019), https://doi.org/10.1145/3341981.3344249, URL https://doi.org/10.1145/3341981.3344249

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

[20] Hauff, C., Hiemstra, D., de Jong, F.: A survey of pre-retrieval query performance predictors. In: Shanahan, J.G., Amer-Yahia, S., Manolescu, I., Zhang, Y., Evans, D.A., Kolcz, A., Choi, K., Chowdhury, A. (eds.) Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008, pp. 1419–1420, ACM (2008), https://doi.org/10.1145/1458082.1458311, URL https://doi.org/10.1145/1458082.1458311

[21] Hofstätter, S., Lin, S., Yang, J., Lin, J., Hanbury, A.: Efficiently teaching an effective dense retriever with balanced topic aware sampling. In: Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., Sakai, T. (eds.) SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, pp. 113–122, ACM (2021), https://doi.org/10.1145/3404835.3462891, URL https://doi.org/10.1145/3404835.3462891

[22] Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., Grave, E.: Towards unsupervised dense information retrieval with contrastive learning. CoRR **abs/2112.09118** (2021), URL https://arxiv.org/abs/2112.09118

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

[24] Mothe, J., Tanguy, L.: Linguistic features to predict query difficulty. In: ACM Conference on research and Development in Information Retrieval, SIGIR, Predicting query difficulty - methods and applications workshop, pp. 7–10, Salvador de Bahia, Brazil (2005)

[25] Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009), https://doi.org/10.1561/1500000019, URL https://doi.org/10.1561/1500000019

[26] Roy, D., Ganguly, D., Mitra, M., Jones, G.J.F.: Estimating gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction. Inf. Process. Manag. **56**(3), 1026–1045 (2019), https://doi.org/10.1016/j.ipm.2018.10.009, URL https://doi.org/10.1016/j.ipm.2018.10.009

[27] Rutherford, A.: ANOVA and ANCOVA: a GLM approach. John Wiley & Sons (2011)

[28] Salamat, S., Arabzadeh, N., Seyedsalehi, S., Bigdeli, A., Zihayat, M., Bagheri, E.: Neural disentanglement of query difficulty and semantics. In: Frommholz, I., Hopfgartner, F., Lee, M., Oakes, M., Lalmas, M., Zhang, M., Santos, R.L.T. (eds.) Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023, pp. 4264–4268, ACM (2023), https://doi.org/10.1145/3583780.3615189, URL https://doi.org/10.1145/3583780.3615189

[29] Shtok, A., Kurland, O., Carmel, D.: Using statistical decision theory and relevance models for query-performance prediction. In: Crestani, F., Marchand-Maillet, S., Chen, H., Efthimiadis, E.N., Savoy, J. (eds.) Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010, pp. 259–266, ACM (2010), https://doi.org/10.1145/1835449.1835494, URL https://doi.org/10.1145/1835449.1835494

[30] Shtok, A., Kurland, O., Carmel, D., Raiber, F., Markovits, G.: Predicting query performance by query-drift estimation. ACM Trans. Inf. Syst. **30**(2), 11:1–11:35 (2012), https://doi.org/10.1145/2180868.2180873, URL `https://doi.org/10.1145/2180868.2180873`

[31] Singh, A., Ganguly, D., Datta, S., MacDonald, C.: Unsupervised query performance prediction for neural models with pairwise rank preferences. In: Chen, H., Duh, W.E., Huang, H., Kato, M.P., Mothe, J., Poblete, B. (eds.) Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023, pp. 2486–2490, ACM (2023), https://doi.org/10.1145/3539618.3592082, URL `https://doi.org/10.1145/3539618.3592082`

[32] Tao, Y., Wu, S.: Query performance prediction by considering score magnitude and variance together. In: Li, J., Wang, X.S., Garofalakis, M.N., Soboroff, I., Suel, T., Wang, M. (eds.) Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014, pp. 1891–1894, ACM (2014), https://doi.org/10.1145/2661829.2661906, URL `https://doi.org/10.1145/2661829.2661906`

[33] Tukey, J.W.: Comparing individual means in the analysis of variance. Biometrics **5**(2), 99–114 (1949), ISSN 0006341X, 15410420, URL `http://www.jstor.org/stable/3001913`

[34] Xiong, L., Xiong, C., Li, Y., Tang, K., Liu, J., Bennett, P.N., Ahmed, J., Overwijk, A.: 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 (2021), URL `https://openreview.net/forum?id=zeFrfgyZln`

[35] Yang, L., Ai, Q., Guo, J., Croft, W.B.: anmm: Ranking short answer texts with attention-based neural matching model. In: Mukhopadhyay, S., Zhai, C., Bertino, E., Crestani, F., Mostafa, J., Tang, J., Si, L., Zhou, X., Chang, Y., Li, Y., Sondhi, P. (eds.) Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016, pp. 287–296, ACM (2016), https://doi.org/10.1145/2983323.2983818, URL `https://doi.org/10.1145/2983323.2983818`

[36] Zamani, H., Croft, W.B., Culpepper, J.S.: Neural query performance prediction using weak supervision from multiple signals. In: Collins-Thompson, K., Mei, Q., Davison, B.D., Liu, Y., Yilmaz, E. (eds.) The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018, pp. 105–114, ACM (2018), https://doi.org/10.1145/3209978.3210041, URL `https://doi.org/10.1145/3209978.3210041`

[37] Zendel, O., Shtok, A., Raiber, F., Kurland, O., Culpepper, J.S.: Information needs, queries, and query performance prediction. In: Piwowarski, B., Chevalier, M., Gaussier, É., Maarek, Y., Nie, J., Scholer, F. (eds.) Proceedings of the 42nd International ACM SIGIR Conference on Research and De-

velopment in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, pp. 395–404, ACM (2019), https://doi.org/10.1145/3331184.3331253, URL https://doi.org/10.1145/3331184.3331253

[38] Zhan, J., Mao, J., Liu, Y., Guo, J., Zhang, M., Ma, S.: Optimizing dense retrieval model training with hard negatives. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 1503–1512, SIGIR '21, Association for Computing Machinery, New York, NY, USA (2021), ISBN 9781450380379, https://doi.org/10.1145/3404835.3462880, URL https://doi.org/10.1145/3404835.3462880

[39] Zhao, W.X., Liu, J., Ren, R., Wen, J.: Dense text retrieval based on pretrained language models: A survey. CoRR **abs/2211.14876** (2022), https://doi.org/10.48550/ARXIV.2211.14876, URL https://doi.org/10.48550/arXiv.2211.14876

[40] Zhao, Y., Scholer, F., Tsegay, Y.: Effective pre-retrieval query performance prediction using similarity and variability evidence. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) 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, pp. 52–64, Springer (2008), https://doi.org/10.1007/978-3-540-78646-7_8, URL https://doi.org/10.1007/978-3-540-78646-7_8

[41] Zhou, Y., Croft, W.B.: Query performance prediction in web search environments. In: Kraaij, W., de Vries, A.P., Clarke, C.L.A., Fuhr, N., Kando, N. (eds.) 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, pp. 543–550, ACM (2007), https://doi.org/10.1145/1277741.1277835, URL https://doi.org/10.1145/1277741.1277835