A Contrastive Neural Disentanglement Approach for Query Performance Prediction

Sara Salamat[1*], Negar Arabzadeh[2], Shirin Seyedsalehi[1], Amin Bigdeli[2], Morteza Zihayat[1], Ebrahim Bagheri[1]

[1*]Toronto Metropolitan University, Toronto, Ontario, Canada.
[2]University of Waterloo, , Waterloo, Ontario, Canada.

*Corresponding author(s). E-mail(s): sara.salamat@torontomu.ca;
Contributing authors: narabzad@uwaterloo.ca;
shirin.seyedsalehi@torontomu.ca; abigdeli@uwaterloo.ca;
mzihayat@torontomu.ca; bagheri@torontomu.ca;

Abstract

We propose a novel approach, referred to as Contrastive Disentangled Representation for Query Performance Prediction (`CoDiR-QPP`), to estimate search query performance by disentangling query content semantics from query difficulty. Our proposed approach leverages neural disentanglement to isolate the information need expressed in search queries from the complexities that affect retrieval performance. Motivated by empirical observations that varying query formulations for the same information need can significantly impact retrieval outcomes, we hypothesize that separating content semantics from query difficulty can enhance query performance prediction. Utilizing contrastive learning, `CoDiR-QPP` distinguishes between well-performing and poorly performing query variants, facilitating the estimation of a given query's performance. Our extensive experiments on four standard benchmark datasets demonstrate that `CoDiR-QPP` outperforms state-of-the-art baselines in predicting query performance, offering improved semantic similarity computation and higher correlation metrics such as Kendall τ, Spearman ρ, and scaled Mean Absolute Ranking Error (sMARE).

Keywords: Query Performance Prediction, Information Retrieval, Neural Disentanglement

1 Introduction

Recent advancements in Information Retrieval (IR) have been significantly driven by the adoption of neural retrieval methods [1]. These methods excel primarily due to their capacity to learn latent data distributions through dense representations. For example, in ad hoc retrieval, dense neural rankers leverage these representations to effectively bridge the query space with the document space, thus enhancing the retrieval of relevant documents for a given query [2] . However, despite these improvements, the benefits are not uniformly distributed across all ranges of queries [2]. More specifically, information retrieval methods seem to be quite effective on a subset of the query space and not so effective on others, leading to disparities in retrieval effectiveness. This uneven performance across different query spaces underscores the importance of *Query Performance Prediction (QPP)* which focuses on estimating how well an information retrieval method is able to satisfy an input query.

Effective query performance prediction plays a crucial role in enhancing the effectiveness and efficiency of information retrieval systems in several real-world applications [3–8]. One prominent application is in adaptive retrieval strategies, where QPP can assess the difficulty of a query and adjust the retrieval process accordingly per query [9, 10]. By predicting query difficulty, it would be possible to deploy cost-effective rankers for easier queries and more robust ones for more difficult queries in order to optimize resource use and minimize latency. Furthermore, QPP can enhance user engagement by identifying challenging queries, prompting users to clarify their intent [11, 12], thus improving overall user experience and quality of interactions [13]. Another example of the application of QPP methods is ranked list truncation for multi-stage ranking pipelines. QPP can balance effectiveness against efficiency by dynamically adjusting the document pool size based on query complexity [4, 14]. In addition, QPP is widely used in federated search and metasearch engines, where it can guide the integration of results from multiple data sources by weighting them based on their estimated quality. Additionally, QPP is valuable for content enhancement through missing content analysis, enabling system administrators to identify and address gaps in the document collection to meet emerging user needs more effectively. These applications illustrate how QPP contributes to making retrieval systems more responsive, efficient, and user-focused.

Given the significance of the QPP task in information retrieval, existing research have focused on analyzing information such as corpus statistics [15], association between the query and document spaces [16], distribution of neural embeddings [17], among others to estimate query performance at runtime. In this paper, we propose a novel approach, namely `CoDiR-QPP` (Contrastive Disentangled Representation for Query Performance Prediction), to estimate query performance by proposing to perform *neural disentanglement* on query and document representations such that the information need expressed through the query is isolated from the actual expression of the query that would impact how difficult or easy the query would be for the retrieval method. Our work is primarily motivated by the following empirical observations in the literature:

Table 1 Examples of query variants representing the same information need with relatively lower (left column) and higher (right column) effectiveness.

Query	AP	Query	AP
kwh solar system cost	0.0031	how much for 20kw of solar panels	1.0
average salary for nurses in dallas texas	0.0345	average salary for registered nurse in dallas	0.3333
adrenaclick price	0.3333	adrenaclick pens cost	1.0
how much does an accounts assistant earn	0.0	what is the average wage for accounting assistants	0.1667
cost of gutter stuff	0.0039	average cost to install gutter guards	1.0

1. **Variability in expression of information needs:** Previous work has shown that identical information need can be articulated through varying query formulation which significantly affect retrieval performance [18–21]. For instance, a clearly phrased query can lead to a high-quality search results, whereas an ambiguous query might result in poor outcomes. This suggests that query content—representing the core information need— can and should be distinguished from the query's complexity. As such, we hypothesize that disentangling content semantics from the difficulty of the query might offer a means for estimating query performance.
2. **Effectiveness of contrastive learning:** Contrastive learning has been shown to be highly effective in various downstream IR and NLP tasks, including but not limited to ranking and question answering systems [22–25]. By comparing different variants of the same query with different degrees of effectiveness, one can potentially expose the model to both poorly performing and well-performing queries that are representing the same information need. This approach could allow for identifying aspects of a query that make the query to be difficult (or easy) regardless of the information need that the query is seeking to fulfil.

The aforementioned points motivate our proposed approach, which focuses on disentangling content semantics and difficulty of the query through a contrastive learning approach. Table 1 shows examples where the same information need is expressed through two different queries: The left column displays poorly performing queries while queries ~~on~~ in the right column enjoy higher performance. We note that queries on the left and the right are expressed to fulfill the same information need. For example, consider a user seeking up-to-date pricing information for Adrenaclick, a brand of epinephrine auto-injector pens used for emergency treatment of severe allergic reactions. The user can represent this information need via different queries. For instance, while the query 'adrenaclick price' has an Average Precision of 0.33, the alternative query 'adrenaclick pens cost', which expresses the same information need obtains a perfect average precision of 1.0. Such observations allow us to hypothesize that it may be possible to separate query content semantics from the difficulty of the query.

Disentanglement approaches are designed to separate distinct factors of variation within data, enabling improved understanding, interpretation, and manipulation of complex datasets. As illustrated in Figure 1, the disentanglement process is applied to a latent space vector. Typically, the outputs of neural models are a mixture of intertwined attributes that are difficult to separate. Disentanglement methods guide

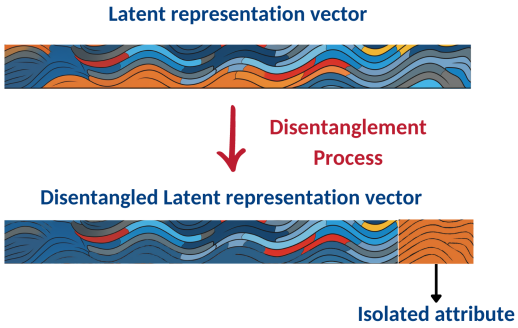

Fig. 1 Visualization of the disentanglement process: The initial latent representation vector (top) undergoes a disentanglement process, resulting in a disentangled latent representation vector where an individual attribute can be independently identified and extracted (bottom).

these models to produce latent vectors where one or more desired attributes can be independently identified and extracted.

On this basis, we propose to disentangle a query's representation into two independent components: one representing query content semantics and the other representing query difficulty. With such decoupling, we anticipate that similar queries will have similar semantics representations, while the difficulty component would capture how challenging the query would be for the retrieval method to satisfy. The contributions of this paper are summarized as follows:

- We introduce `CoDiR-QPP`, a novel framework that utilizes neural disentanglement to separate query semantics from query difficulty to enhance the accuracy of the query performance prediction task;
- We propose to jointly adopt contrastive representation learning and point-wise prediction in order to differentiate between query difficulty and query semantics such that query performance is accurately estimated both when comparing queries with each other and also in isolation;
- Through extensive experiments on multiple benchmark datasets, such as MS MARCO Development Set, TREC DL 2019 and 2020, and TREC DL-Hard [26], we find that our proposed disentanglement approach offers a superior means for query performance prediction, outperforming strong state of the art baseline methods on metrics such as Kendall τ and Spearman ρ Correlations, and the scaled Mean Absolute Ranking Error (sMARE) [27, 28]. In addition, we empirically show that our disentangled content semantics representations allow for a more refined computation of semantic similarity across different queries.

2 Related Work

Query Performance Prediction. QPP methods are broadly categorized into (1) Pre-retrieval and (2) Post-retrieval classes. The former predictors estimate query performance based solely on the query and the collection statistics before any documents are retrieved [15, 16, 29]. These methods are particularly valuable because they are

generally fast and do not depend on the results of the query, making them ideal for real-time applications. Unlike pre-retrieval methods, post-retrieval predictors utilize information from the retrieved documents to predict query performance. These methods tend to be more accurate as they incorporate feedback from the retrieval process itself [30–33]. While post-retrieval QPP has shown to be more effective, they have more limited applications since the system has already lost on query latency time given it needs to perform one round of retrieval on the given query before being able to estimate the query performance. In this work, we focus on the more challenging task of *pre-retrieval QPP*, which is crucial for applications requiring low-latency responses.

Historically, QPP research has focused on the statistical relationships between query terms and the document corpus [29, 31, 34–38]. A common approach involves using signals derived from comparing the language model of the query with that of the collection, treating this comparison as an indicator of query difficulty. The more similar the query is to the corpus, the higher the likelihood that relevant documents exist in the collection, making the query easier to satisfy [15, 39]. [39, 40]. Another common approach for traditional QPP methods postulates that queries with highly coherent terms tend to retrieve more relevant documents, leading to better performance [37].

With the advent of neural-based models, query performance prediction has greatly improved. Recent advances have focused on leveraging neural embeddings to enhance pre-retrieval predictions. For example, Zhou and Croft [41] used word embeddings to analyze semantic coherence among query terms, demonstrating that semantic relationships are strong indicators of query performance. Arabzadeh et al. [42] proposed using neural embedding representations of queries to assess query specificity, which serves as an indicator of performance. Similarly, Roy et al. [43] used contextual embeddings to evaluate query ambiguity by estimating the number of senses associated with each query term. These neural-based methods have generally outperformed traditional term frequency-based pre-retrieval methods across various benchmarks [44]. More recent studies have utilized deep learning-based models to tackle the QPP task and they demonstrate that supervised methods for QPP are more effective than unsupervised approaches. However, these supervised methods necessitate a substantial amount of data and training instances to perform QPP effectively [45–48]. While supervised neural-based approaches have been extensively explored in post-retrieval QPP, to the best of our knowledge, they have not yet been explored in pre-retrieval QPP due to the limited information available from the query alone, which is insufficient for training.

Neural Disentanglement. In this paper, we not only leverage contextualized neural representations of queries to preserve their semantic integrity of queries but also propose a neural disentanglement-based method to distinguish between query content semantics and difficulty. While previous studies have explored the concept of query difficulty, none have explicitly disentangled a query's semantics from its difficulty. Neural disentanglement, primarily used in NLP tasks, involves separating different attributes embedded in neural representations, such as content, style, and tone [49–51]. This technique has proven to be effective in tasks such as controlled text generation, style transfer, and sentiment classification [52, 53]. Our work proposes a disentanglement approach inspired by these recent developments, specifically for the disentanglement of query semantics from query difficulty. This is in line with the work by Xie et al.

[54] and Li et al. [55] who proposed to disentangle visual content and style in an unsupervised and interpretable way for image retrieval.

The first attempts to apply concepts of disentanglement in text were inspired heavily by the advances made in the image domain. For instance, the work by Hu et al. [52] extended Variational Autoencoders (VAEs) [56] to handle discrete data such as text, introducing controlled text generation where a user can control the attributes (such as sentiment) of the generated text. Further, disentanglement was used for text style transfer tasks, allowing changes in properties like sentiment, tense, and author style while keeping the main content of the text unchanged [49–51]. Newer approaches of disentanglement have explored the separation of content and attribute by text generation from disentangled representations and re-disentanglement [57]. The use of statistical regularizers to assist disentanglement of textual data [58], and mixup approaches for class-specific features and class-agnostic features disentanglement [59] are among recent approaches for disentangled representation learning. Inspired by the success of neural disentanglement methods in addressing various downstream tasks, in this work, we propose `CoDiR-QPP` to advance previous disentanglement approaches and enhance the state-of-the-art performance in QPP tasks.

3 Proposed Approach

3.1 Problem Formulation

In an information-seeking system, a query q is submitted to a retrieval system π. The system processes a collection of documents C to produce a ranked list of documents D_q that aims to satisfy the information need I_q behind query q. I_q represents the underlying intent or the specific information that a user seeks when formulating a query to begin a search process. I_q is inherently abstract and can be influenced by various factors such as the user's background knowledge, the context of the search, and the specific requirements or constraints of the information sought. The process is denoted as $D_q \leftarrow \pi(q, C)$. The retrieval effectiveness of system π is usually quantified by the quality of retrieved documents using an evaluation metric $\mu(D_q, q \mid R_q)$ where R_q is the set of relevant judged documents for query q from the corpus C. In general, μ assesses D_q based on the proportion of retrieved documents that are considered to be relevant for query q.

QPP estimates the effectiveness of retriever π, denoted as $\hat{M}(q \mid \pi, C)$, to assess the quality of D_q without having access to the actual effectiveness of the system μ or the relevant documents R_q. The accuracy of QPP is judged by how closely $\hat{M}(q \mid \pi, C)$ approximates $\mu(D_q, q \mid R_q)$. In pre-retrieval QPP, predictions $\hat{M}_{pre}(q \mid \pi, C)$ are made without having access to D_q, leveraging only the query conditioned on the type of retriever π and the document collection C. Pre-retrieval QPP acts prior to retrieval $\pi(q, C)$ and it offers benefits such as reduced query latency. This early prediction, denoted as $\hat{M}_{\text{pre}}(q \mid \pi, C)$, facilitates more real-time applications by allowing the retriever to respond quickly, thereby expanding the potential actions it can perform.

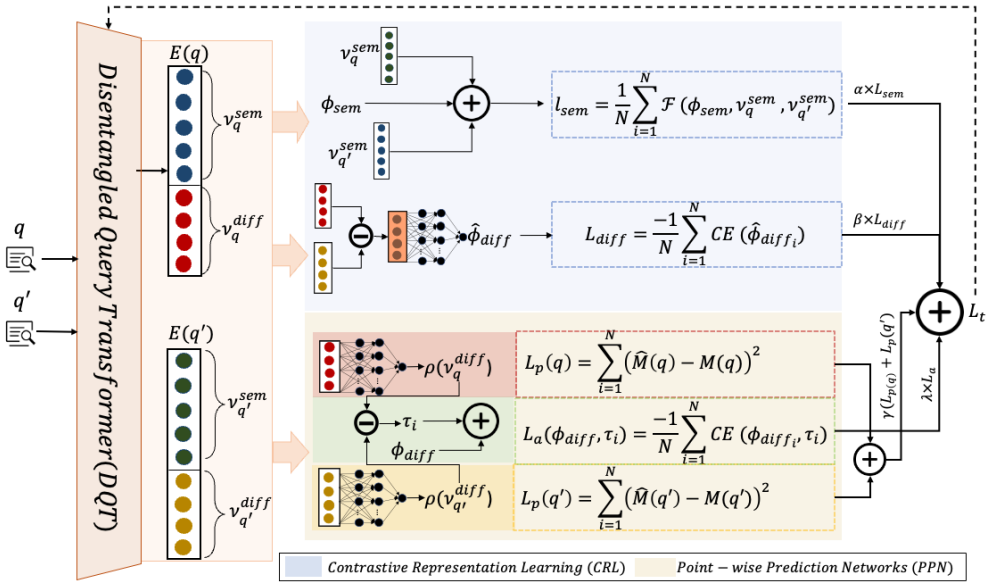

Fig. 2 Overview of the Proposed CoDiR-QPP Framework.

3.2 Model Framework

In order to disentangle content semantics from query difficulty, we undertake three tasks in tandem: **(i) Semantic-Difficulty disentanglement in query representation:** In this first task, given two queries that have similar content semantics but have varying degrees of difficulty, we focus on disentangling the representation of such queries in a way that their content semantics representation is comparable yet their difficulty representation is different in a way that solely based on the difficulty representations of this pair of query, it would be possible to train a classifier that could accurately predict the more difficult query. This will ensure that the information encoded into the difficulty representation of queries focuses solely on aspects of query representation that impact retrieval effectiveness for cases when the information need from the queries is identical. **(ii) Semantic validation across queries:** In the second task, we focus on learning query content semantics regardless of their difficulty. To this end, given two pairs of queries and regardless of their difficulty, we train the model to be able to predict whether two queries are addressing the same information need or not, solely using their content semantics representation. This will ensure that the network is separating out information related to query information need and capturing them in the content semantics representation component of the query representation. **(iii) Performance prediction of queries:** Finally, ~~and~~ in the third task, we focus on predicting the actual performance of a single query regardless of its comparison with other queries. In this task, our objective is to learn a regressor over the query difficulty representation such that the predicted value is an accurate depiction of the query performance at runtime. This task will ensure that we not only

are able to rank queries based on their difficulty when comparing two queries, but also are able to assess each query's performance in isolation.

We propose CoDiR-QPP, an end-to-end framework for disentangling content and difficulty of queries in their representation through constrastive learning. CoDiR-QPP includes three components that accomplish the three tasks, as shown in Figure 2:

1. **Disentangled Query Transformer (DQT):** This component converts query representations into a vector space where specific segments of the representation are designated for query difficulty and query semantics (content). This process breaks down the query vector representation into two distinct vectors: one capturing the difficulty level and another encapsulating the content semantics of the query.
2. **Contrastive Representation Learning (CRL):** This component employs contrastive learning to have a more accurate representation of content and difficulty of the query representation. Given a pair of queries associated with the same information need, this component brings the content representations of query pairs closer to each other. Simultaneously, it allows the difficulty aspect of the representations to diverge based on the actual performance of the queries. In other words, this component is responsible for enhancing the semantic representation of the queries by bringing queries with similar content semantics closer together in the embedding space, while distinguishing between these queries based on their performance.
3. **Point-wise Prediction Networks (PPN):** The Contrastive Representation Learning component is focused on distinguishing between queries that cater the same information need by have varying degrees of query difficulty. This requires the model to always compare two queries and make a determination as to which is more difficult. However, it is also important to provide a point-wise component that determines query difficulty on its own merit and not through pairwise comparison, which is acheived through the textttPPN component in our proposed approach.

3.2.1 Disentangled Query Transformer (DQT)

For each query q, we obtain the vector representation of the query $E(q)$ in a t-dimensional space. This representation is crucial as it captures the semantic and syntactic aspects of the query, providing a robust basis for subsequent processing steps.

Given a pair of queries, q and q', the Disentangled Query Transformer's (DQT) responsibility is to obtain the vector representations $E(q)$ and $E(q')$ such that both representations belong to the same embedding space. Additionally, DQT should be able to deterministically disentangle between v_q^{sem} and v_q^{diff}, where the former represents the semantic aspect of the query and the latter indicates the difficulty of the query. The relationship between these vectors is fundamental to the model's ability to differentiate between query semantics and query difficulty. Given a pair of queries q_i and q_i', DQT will first embed the representation of each query through a transformer model Θ and then disentangle them individually as follows:

$$\texttt{DQT}(q_i, q_i'|\Theta) = \Big((v_q^{sem} \oplus v_q^{diff}), (v_{q'}^{sem} \oplus v_{q'}^{diff})\Big) \tag{1}$$

where $\mid v_q \mid = t$, the semantic and difficulty vectors of each query will have a size of t_s and t_d such that $t = t_{diff} + t_{sem}$.

3.2.2 Contrastive Representation Learning (CRL):

The goal of this component is to produce representations for both segments of v_q^{sem} and v_q^{diff} through contrastive learning. This is pivotal in refining the model's ability to distinctly recognize the semantic relationship between query pairs. The essence of this component lies in adjusting the proximity of semantically similar and dissimilar query pairs in the embedding space. For pairs that are identified as semantically similar, the loss function strives to minimize the cosine distance between their vector representations, thereby bringing them closer within the space. Conversely, for those pairs identified as semantically dissimilar, the function tends to maximize their spatial separation, hence ensuring clear differentiation.

Let us assume a query pair Q which comprises queries $[q_i, q_i']$ where one of them shows higher effectiveness based on their retrieved documents D_{q_i} or $D_{q_i'}$ compared to the other one in terms of evaluation metric μ based on the relevance judged set R_q:

$$Q = \{[q_i, q_i'] \mid \mu(q_i, D_{q_i} \mid R_q) > \mu(q_i', D_{q_i'} \mid R_q)\} \tag{2}$$

We define function ϕ_{diff} to determine the relative difficulty of a query pair $[q_i, q_i']$ as follows:

$$\phi_{diff}\left[q_i, q_i'\right] = \begin{cases} 1 & \mu(q_i, D_{q_i} \mid R_q) > \mu(q_i', D_{q_i'} \mid R_q) \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

Queries can carry the same semantics but be expressed in different forms which affects their difficulty. As such, we define ϕ_{sem} for determining whether the pair of queries are referring to same information need ($I_q == I_{q'}$) or not.

$$\phi_{sem}\left[q_i, q_i'\right] = \begin{cases} 1 & I_{q_i} == I_{q_i'} \\ -1 & \text{Otherwise} \end{cases} \tag{4}$$

For simplicity, we refer to $\phi_{diff}\left[q_i, q_i'\right]$ and $\phi_{sem}\left[q_i, q_i'\right]$ as ϕ_{diff_i} and ϕ_{sem_i}, respectively. The semantic component of the query representation is processed using a contrastive similarity-based loss function, which refines the model's ability to distinguish between semantic similarities and differences among query pairs. Without loss of generality, we measure the similarity between the semantic component of the representation of two queries $v_{q_i}^{sem}$ and $v_{q_i'}^{sem}$ with cosine similarity between their disentangled semantic vector representation of those queries. The following loss function minimizes the cosine distance between vectors representing semantically similar query pairs, drawing them closer in the vector space.

$$L_{\text{sem}} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1+\phi_{\text{sem}_i}}{2}\left(1-\text{sim}(v_{q_i}^{sem}, v_{q_i'}^{sem})\right) + \frac{1-\phi_{\text{sem}_i}}{2}\max\left(0, \text{sim}(v_{q_i}^{sem}, v_{q_i'}^{sem})\right)\right) \tag{5}$$

where N is the total number of query pairs. This loss function effectively enables the model to optimize vector representations such that they accurately reflect the semantic intent of each query pair.

We propose to leverage contrastive learning not only for semantic representation, but also for difficulty representation of the query. To do so, we incorporate a fully connected neural network $\mathcal{F}$ designed to further refine the disentanglement process by focusing on the difficulty of the queries. Since the idea behind this learning is based on the difference between the queries that have the same information need, the input to network $\mathcal{F}$ is the subtraction of the difficulty of the two queries, i.e., $v_{q_i}^{diff} - v_{q_i'}^{diff}$, representing the disparity in difficulty aspects between the query vectors q_i and q_i'. The purpose of this network is to predict the function ϕ_{diff_i} by $\hat{\phi}_{diff_i}$, which serves as an indicator of relative difficulty between the two queries as follows:

$$\hat{\phi}_{diff_i} = \mathcal{F}(v_{q_i}^{diff} - v_{q_i'}^{diff}) \tag{6}$$

Here, network $\mathcal{F}$ will be optimized through minimizing the cross entropy loss for difficulty prediction, denoted as L_{Diff}, formulated as follows:

$$L_{diff} = \frac{-1}{N}\sum_{i=1}^{N}\left(\phi_{diff_i}\log(\hat{\phi}_{diff_i}) + (1-\phi_{diff_i})\log(1-\hat{\phi}_{diff_i})\right) \tag{7}$$

These combined mechanisms within `CRL` not only enhance the model's capacity to separate query semantics from query difficulty effectively but also optimize the alignment of these components for more effective query performance prediction.

3.2.3 Point-wise Prediction Networks (`PPN`):

The Disentangled Query Transformer followed by Contrastive Representation Learning enables us to obtain a semantic representation of a pair of queries, $v_{q_i}^{sem}$ and $v_{q'i}^{sem}$, as well as their difficulty, $v_{q_i}^{diff}$ and $v_{q'i}^{diff}$. Based on these difficulty representations, we are able to obtain $\hat{\phi}_{diff_i}$, which, as explained in Equation 3, predicts a binary value indicating whether the performance of q_i is better than that of q_i'. However, for more accurate predictions about the performance of the queries, we need to move beyond the binary predictions derived from pairwise contrastive comparisons to a more regression-based comparison that can predict the actual performance of each query. To enhance the quality of the disentanglement process, we have incorporated auxiliary functions by defining two separate networks, both mirroring the architecture of $\mathcal{F}$. These networks

are specifically trained to predict the individual performance $\mu(q, D_q \mid R_q)$ for each query based solely on their difficulty vectors, i.e., v_q^{diff}.

The introduction of $\mathcal{P}$ to predict a point-wise scalar value serves a dual purpose. First, they provide direct feedback to the model about the accuracy of the difficulty representations it generates. Second, by focusing on predicting the actual fine-grained scalar value of the evaluation metric of interest (e.g., average precision), we reinforce the training process to go beyond learning difficulty as a comparative task.

Therefore, we predict the scalar value of difficulty of the given queries $\hat{M(q)}$ based on their difficulty representation v_q^{diff} through network $\mathcal{P}$ as $\hat{M}(q) = \mathcal{P}(v_q^{diff})$ and the following loss function for point-wise prediction of the performance L_p:

$$L_p = \sum_{i=1}^{N} \Big(\hat{M}(q) - M(q)\Big)^2 \tag{8}$$

Now, we need to establish a connection between the `PPN` and the `CRL` components to deepen the model's understanding of the relationships among the networks $\mathcal{F}$ and $\mathcal{P}$. We want the model to not only predict the difference label accurately but also to align the difference between the individual predicted performances with the difference label ϕ_{diff}. To this end, we introduce L_a, which aligns the two networks and acts as a bridge, connecting the outputs of the two networks that predict individual performance from difficulty vectors. L_a is designed to ensure that τ_i, which serves as a predictive counterpart to the predetermined difference label $\phi[q_i, q_i']$, is properly aligned.

$$\tau_i = \mathcal{P}(v_{q_i}^{diff}) - \mathcal{P}(v_{q_i'}^{diff}) = \hat{M}(q_i) - \hat{M}(q_i') \tag{9}$$

$$L_a = \frac{-1}{N} \sum_{i=1}^{N} [\phi_{diff_i} \log(\tau_i) + (1 - \phi_{diff_i}) \log(1 - \tau_i)] \tag{10}$$

This integrated approach not only enhances the model's predictive capabilities but also fosters a more cohesive learning environment. It allows the model to develop a more holistic understanding of the interconnected aspects of query difficulty, thereby improving the overall effectiveness of the disentanglement process. The total loss function of the model is the linear interpolation of the defined loss functions as follows:

$$L_t = \alpha L_{sem} + \beta L_{Diff} + \gamma(L_{p_q} + L_{p_{q'}}) + \lambda L_a \tag{11}$$

Once the forward pass computes the outputs based on current weights, L_t is calculated. The gradients of L_t with respect to the output neurons are first calculated, and these gradients are successively propagated backwards through the network's layers. Each layer's weights are updated by moving in the direction that reduces L_t, using an optimization algorithm. This iterative adjustment continues across multiple epochs, progressively refining the model's weights to enhance predictive accuracy and align with the training data's underlying patterns.

3.3 Inference

During the inference stage, the objective is to predict the performance of each query based on their disentangled representations. This process involves the following steps: **Query Embedding and Disentanglement:** Each input query q is first processed by the Disentangled Query Transformer (`DQT`), which encodes the query into a vector representation that is then split into semantic and difficulty components:

$$\mathbb{E}(q) = \texttt{DQT}(q|\Theta) = (v_q^{sem} \oplus v_q^{diff})$$

Here, v_q^{sem} denotes semantic content of the query, and v_q^{diff} represents its difficulty. **Difficulty-Based Performance Prediction:** The difficulty vector v_q^{diff} is utilized by the Point-wise Prediction Networks (`PPN`) to predict the query's performance. This is done by feeding v_q^{diff} into a neural network $\mathcal{P}$, which outputs a scalar value indicating the predicted effectiveness of the query:

$$\hat{M}(q) = \mathcal{P}(v_q^{diff})$$

This predicted value $\hat{M}(q)$ represents the effectiveness of the query, serving as a direct measure of the query's potential performance in the retrieval system.

4 Experiments

4.1 Experimental Setup

Training Data and Setup. As detailed in Section 3.2, `CoDir-QPP` requires the use of query pairs $[q, q']$. To address this need, we construct a dataset that includes these query pairs along with their corresponding actual performance (μ) in terms of Mean Average Precision at a cut-off of the top-1000 retrieved documents. These query pairs were sources from the Matched Made in Heaven (MMH) dataset [19].

The MMH dataset features queries that, despite having similar information needs $\phi_{sem}[q_i, q_i'] = 1$, are represented with different linguistic formulations, resulting in queries of similar content but varying difficulty. This dataset includes over 400,000 such pairs, each demonstrating varied performance levels. The MMH dataset has been built on top of the MS MARCO V1 collection. Microsoft Machine Reading Comprehension (MS MARCO) is a widely used, large-scale, well-known dataset in information retrieval and NLP downstream tasks. MS MARCO contains over 8.8 million passages and more than 500K queries, each linked to at least one relevant passage. MS MARCO provides a rich source of queries and associated passages, making it an ideal foundation for creating query pairs with varying performance metrics.

To select contrastive samples for the training, we also randomly assigned queries from the MS MARCO V1 train queries to ensure we incorporated queries with non-similar content and different complexity levels along with their performance metrics. Consequently, our final training dataset expanded to include over 400,000 query pairs. These pairs are categorized into two groups: those with similar content are labeled as 1 ($\phi_{sem}[q_i, q_i'] = 1$), and those with differing content are labeled as -1 ($\phi_{sem}[q_i, q_i'] = -1$),

as defined in Equation 4. This approach allows our model to train on a balanced dataset representing a broad spectrum of query difficulties and content similarities, which is crucial for enhancing the robustness and accuracy of the `CoDir-QPP` framework. Our dataset is available to download on our GitHub repository[1].

For the training setup, 80% of the dataset was allocated for training purposes, while the remaining 20% served as the validation set. We optimized our model using the Stochastic Gradient Descent (SGD) algorithm, with a learning rate set at 10^{-4}. This low learning rate was chosen to allow for gradual adjustments to the model weights, thereby enhancing the training process's stability. Training spanned 10 epochs, during which we continuously evaluated the model's performance against the validation set at the end of each epoch to ensure accuracy and robustness.

Evaluation Data and Strategies. We conducted experiments on the MS MARCO V1 passage collection and four associated query sets, including the MS MARCO development (dev) set, which contains 6,980 queries. Additionally, we examined TREC DL 2019 (43 queries), TREC DL 2020 (53 queries), and DL-Hard (50 queries), all of which feature comprehensive judgments on a non-binary graded scale. The key difference between these query sets lies in their evaluation and labeling. The MS MARCO dev set is evaluated mainly with one annotated relevant passage per query, making its labeling quite sparse. In contrast, the other three query sets have relatively more comprehensive labeling, with an average of about 200 documents per query annotated. Moreover, DL-Hard contains the most challenging queries of the group, with even more judgments and queries that are difficult to satisfy.

To evaluate the effectiveness of `CoDir-QPP`, we utilized two common evaluation strategies. First, we calculated rank-based and linear-based correlations to assess the correlation between the predicted performance of our method and the actual performance of queries retrieved by BM25, as implemented by Anserini. For actual performance metrics, we used MRR@10 for the MS MARCO dev set and nDCG@10 for the other three query sets. Higher correlation values are indicative of more accurate predictions of query performance.

Furthermore, as the second strategy, we utilized the Symmetric Mean Absolute Error (sMARE) metric [28]. The sMARE metric quantifies the accuracy of a ranking model by measuring the discrepancy between the predicted and the ground truth scores or rankings of a set of items or documents. A lower sMARE value denotes superior performance, indicating a smaller mean discrepancy between the predicted and actual rankings.

Baselines. We compare against state-of-the-art pre-retrieval QPP baseline groups:

- **Term Importance-Based.** Two commonly used statistics in this category are inverse document frequency (`IDF`)[60] and inverse collection term frequency (`ICTF`) [60], which are recognized as measures of the relative importance of query terms. While being simple, these methods show relatively high performance, Also, they are inexpensive to run, therefore enjoying numerous applications in real-world scenarios.
- **Specificity-Based.** This group of pre-retrieval QPP metrics operates based on the idea of divergence between the query language model and the collection language model. The idea is that more specific queries are relatively easier to address than

[1] https://anonymous.4open.science/r/DisentangledQPP-4487

Table 2 Performance comparison on TREC DL-2019 and DL-2020. *Italic* values indicate not statistically significant correlation with p-value of 0.05. Highest values are shown in **bold**.

	TREC DL 2019			TREC DL 2020		
	Kendall ↑	Spearman ↑	sMARE ↓	Kendall ↑	Spearman ↑	sMARE ↓
SCS	0.194	0.287	0.316	0.272	**0.397**	0.333
CC	0.099	0.055	0.319	*0.106*	*0.026*	0.290
DC	0.095	0.053	0.293	0.091	*0.035*	0.327
IEF	0.187	0.166	0.387	0.064	0.081	0.334
SCQ	0.116	0.162	0.387	0.076	0.132	0.365
VAR	0.107	0.152	0.290	0.059	0.077	0.318
PMI	*0.009*	*0.017*	0.341	0.040	0.056	0.344
IDF	0.158	0.245	0.321	0.245	0.353	0.374
ICTF	0.153	0.240	0.360	**0.345**	0.330	0.330
DQT+CRL	0.2	0.3	0.273	0.274	0.385	0.248
DQT+PPN	*0.023*	*0.041*	0.318	*0.167*	*0.239*	0.372
DQT+CRL+(PPN-L_a)	*0.081*	*0.138*	0.308	*0.192*	*0.276*	0.392
CoDiR-QPP	**0.227**	**0.311**	**0.269**	0.265	0.384	**0.248**

more general ones, which are sometimes also found to be ambiguous. Simplified Clarity Score (SCS) is the leader of this group [15].

– **Similarity-Based.** The underlying idea behind this approach is that queries exhibiting high similarity to the collection are likely to be easier to answer. Similarity of Collection and Query (SCQ) belongs to this group and has shown high performance on different IR benchmarks [16].
– **Term-Relatedness Based.** These methods examine the co-occurrence statistics of terms, i.e., when query terms co-occur frequently with each other this can be a sign that they are related to the same topic and hence indication of less difficult queries. Pointwise Mutual Information (PMI) belongs to this group [61].
– **Coherence-Based.** These approaches work on the principle that query coherence refers to the inter-similarity of documents containing the query terms. The idea is that the variance (VAR) of the term weights across the documents containing that term in the collection is an indicator of query difficulty [16].
– **Neural-Based.** This group of methods operates based on the geometric relationships of query terms and their surrounding terms in the embedding space. The idea is that when a query term is surrounded by numerous similar terms in the embedding space, it is more specific. More specific queries can be considered to be less difficult. These baselines leverage various centrality measures such as Closeness Centrality (CC), Degree Centrality (DC), and Inverse Edge Frequency (IEF) to assess query term specificity and hence difficulty.

4.2 QPP Results

Tables 2 and 3 demonstrate the results of our approach and the baselines on four datasets in terms of Kendall-τ and Spearman-ρ as well as the sMARE metric. It is worth noting that for rank-based correlation metrics, higher values indicate better

Table 3 Performance comparison on MS MARCO Dev set and TREC DL-Hard. *Italic* values indicate not statistically significant correlation with p-value of 0.05. Highest values are shown in **bold**.

	MS MARCO Dev Set			TREC DL hard		
	Kendall ↑	Spearman ↑	sMARE ↓	Kendall ↑	Spearman ↑	sMARE ↓
SCS	*0.037*	0.049	0.333	0.106	0.140	0.326
CC	0.065	0.085	0.333	0.103	0.141	0.310
DC	0.107	0.144	0.333	0.123	0.165	0.335
IEF	0.094	0.104	0.330	0.140	0.191	0.377
SCQ	*0.011*	*0.014*	0.334	0.127	0.179	0.369
VAR	0.062	0.083	0.333	*0.016*	*0.035*	0.349
PMI	*0.017*	*0.023*	0.323	*0.022*	*0.031*	0.349
IDF	0.116	0.154	0.330	0.111	0.125	0.255
ICTF	0.114	0.152	0.330	0.107	0.115	0.314
`DQT+CRL`	0.24	0.359	0.259	0.171	0.257	**0.271**
`DQT+PPN`	*0.003*	*0.060*	0.360	*0.171*	*0.266*	0.283
`DQT+CRL+(PPN-`L_a`)`	0.021	0.029	0.326	*0.099*	*0.135*	0.36
`CoDiR-QPP`	**0.260**	**0.385**	**0.252**	**0.221**	**0.335**	0.275

performance. Conversely, for the sMARE metric, a lower value is preferable as it reflects a smaller discrepancy between the predicted and actual rankings of queries.

4.2.1 Ablation Study on `CoDiR-QPP`

Our proposed approach encompasses multiple components, each contributing to the overall prediction performance. To assess the individual impact of these components, we conduct a detailed ablation study to investigate the roles of the `CRL` and `PPN` components by training the model, removing one of these components at a time. As such, we report the result for `DQT+CRL` and `DQT+PPN` separately in Tables 2 and 3. In addition, since the `PPN` component is further divided into three parts, each designed to enhance the model's understanding of query difficulty, we also explore the impact of L_a within the `PPN` network. Accordingly, the last four rows report `DQT+CRL+ (PPN)-`L_a alongside our complete approach `CoDiR-QPP`, which integrates all components.

The results show that while combinations of some components, such as `DQT+PPN` and `DQT+CRL+(PPN-`L_a`)`, do not show statistically significant correlation on some query sets (e.g., TREC DL-2019), combining all three components in `CoDir-QPP` affects the overall performance, with each component contributing to the final result. The `DQT+CRL` model performs comparably to the complete `CoDir-QPP` model, further emphasizing the importance of the `CRL` component in our approach. The ablation study results (`DQT+CRL`, `DQT+PPN`, and `DQT+CRL+(PPN-`L_a`)`) indicate that the complete `CoDiR-QPP` model achieves the best overall performance. Removing every individual component tends to decrease the overall performance, particularly when parts of the `PPN` network are excluded. Specifically, the combination `DQT+CRL` shows improved performance over `DQT+PPN`, suggesting that `CRL` plays a more significant role than the individual predictors within `PPN` in this context. The ablation involving `DQT+CRL+(PPN-`L_a`)` demonstrates a notable drop, particularly in Kendall τ and Spearman ρ coefficients on the TREC DL Hard dataset, suggesting that L_a contributes significantly to handling difficult queries. In summary, the consistent performance drop

across different datasets when components are removed validates the contribution of each component of the CoDiR-QPP model to its overall performance.

4.2.2 Comparison with QPP Baselines

Based on the comparative analysis of our proposed CoDiR-QPP approach against state-of-the-art baselines presented in Tables 2 and 3, we observe the following: **(1)** Term-statistical baselines, including SCQ, VAR, PMI, and IDF exhibit inconsistent performance across various datasets. These baselines demonstrated differing effectiveness on the TREC DL 2019 and 2020 datasets, where metrics such as Spearman and Kendall correlations fluctuated notably between different tests; **(2)** Among the term-statistical baselines, ICTF (Inverse Collection Term Frequency) and SCS (Statistical Corpus Similarity) exhibit more robust performance compared to others. Notably, ICTF shows a significant increase in performance in 2020, particularly in the Kendall-τ and Spearman-ρ metrics as shown in Table 2. Despite this uptick, ICTF does not maintain stable performance across other datasets. For example, on the MS MARCO Dev set and TREC DL-Hard queries, our proposed approach, CoDiR-QPP, significantly outperforms ICTF, exhibiting more than double the performance in terms of both Kendall-τ and Spearman-ρ. Meanwhile, SCS registers the highest performance in terms of Spearman-ρ on TREC DL-2020, yet the performance margin between SCS and CoDiR-QPP is not statistically significant, as evidenced by a paired t-test with a p-value of 0.05. On the MS MARCO development set, CoDiR-QPP continues to outshine all baselines, achieving the highest Kendall-τ and Spearman-ρ correlation coefficients. These results illustrate the strengths of certain statistical baselines while underscoring the superior consistency and effectiveness of CoDiR-QPP; **(3)** The results on the TREC DL-Hard queries shows that CoDiR-QPP offers consistently high performance. Given that TREC DL-Hard is comprised of some of the most challenging queries, our proposed method shows a distinct advantage over the baselines in this demanding dataset. Specifically, CoDiR-QPP achieves a Spearman-ρ of 0.335, markedly higher than the next best baseline, IEF, which attains a correlation of only 0.191. This significant improvement highlights the robustness of CoDiR-QPP in accurately predicting query performance across varying levels of query difficulty; **(5)** The CC, DC, and IEF methods, which are neural-based baselines do not demonstrate significant performance in our experiments. We hypothesize that this is because these methods do not use contextualized representations of query terms; rather, they mostly rely on individual query terms. While the independence of query terms might work for short and keyword-based queries, this assumption is less effective for cases where the queries are in natural question formats (similar to our test cases). As such, we observe that our proposed approach outperforms all the neural-based models reported in the both tables.

Our analysis across multiple datasets and in comparison to baseline methods highlights the robustness and efficacy of our proposed CoDiR-QPP approach for the query performance prediction task. In particular, the ablation study illustrated the significant contributions of the CRL component and the L_a loss function, which are essential in enhancing the model's overall performance. Our results demonstrate the potential of our approach in adapting to varying query characteristics in different query sets.

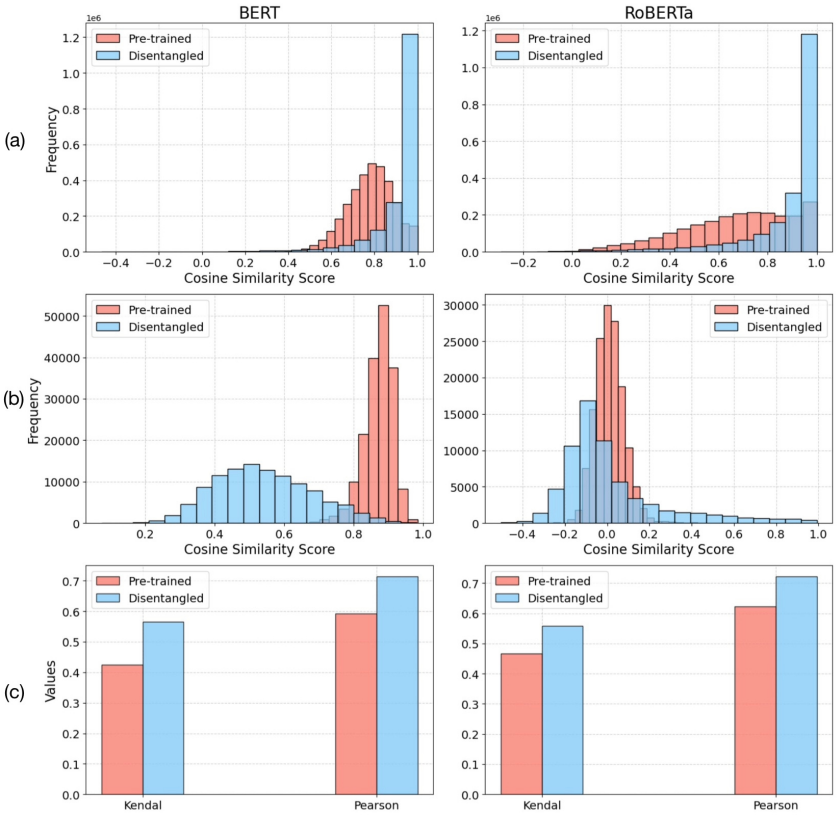

Fig. 3 The quality of the content semantics representations offered by our disentanglement approach.

4.3 Quality of Content Semantics Representations

In the previous section, we demonstrated that the difficulty component of disentangled representations of queries can effectively predict the queries' performance. In this section, we shift our focus to the content semantics component of the disentangled representations. We aim to investigate whether this component retains meaningful and informative content compared to the original query representation about the information need of the query. We investigate if the process of disentanglement successfully isolates query difficulty without compromising the content semantics of the query.

To evaluate the effectiveness of our disentanglement process in preserving query semantics, we analyze query representations before and after disentanglement using two different large language models, including RoBERTa [62], and BERT [63]. Additionally, we utilize the dataset proposed by Nogueira et al. [64], which features queries generated from the MS MARCO passage collection. These queries are designed to reflect possible questions a single passage might address. As suggested by Nogueira et al., we hypothesize that the queries generated from the same passage should exhibit high semantic similarity, a concept that has been used in different tasks such as query

generation, query refinement, and query reformulation [65]. This dataset includes 80 generated queries for each of the 8.8 million passages in the MS MARCO collection.

An effective representation should group semantically similar queries closer together in the embedding space while distancing queries with differing content. To evaluate this, we randomly selected 5,000 query pairs generated from the same documents. This selection is premised on the assumption that such pairs, being derived from identical sources, should exhibit high semantic similarity.

To visualize how well each representation performs, we plotted the distribution of similarities for 5,000 randomly selected query pairs. The resulting histogram, shown in Figure 3 (a), illustrates the distribution of similarity scores. A histogram skewed towards higher similarity values indicates a representation that has effectively captured semantic information since such a model has been able to exhibit closer proximity between semantically similar query pairs. As illustrated in the figure, the disentanglement process has led to increased skewness towards higher similarity scores across both large language models. This result confirms the effectiveness of our disentanglement approach in enhancing the models' capacity to represent query semantics accurately. To further evaluate the effectiveness of the semantic vectors, we selected random pairs of dissimilar queries from the MS MARCO development set. These queries were processed to generate content vectors for each query using both our disentangled models and their respective base pre-trained models. We then calculated the cosine similarity between each pair of queries. Given that the selected queries are semantically dissimilar, the cosine similarity scores should ideally be low. A favorable outcome is indicated by a distribution with a lower mean and a skew towards the left, suggesting that our model more effectively captures the distinctions between dissimilar queries. The distribution of cosine similarity scores are depicted in Figure 3 (b). The histograms show that the distribution of similarity scores for dissimilar queries using the disentangled model is skewed towrds lower scores which again confirms that our disentangled approach has effectively learned the content representations.

As a second experiment to test the quality of the disentangled content semantics component, we evaluated the extent to which these representations capture the semantic representation of queries in general. We leveraged the English versions of the widely used machine-translated multilingual STS benchmark dataset[2]. This dataset includes pairs of sentences accompanied by a similarity score between 0 and 5, from low to high similarity. We ran experiments on 1,500 pairs of sentences from its development set and defined the task as comparing the similarity between the representations of sentences based once on the original large language model and once based on the disentangled content semantics representation obtained from our proposed disentanglement approach. Without loss of generality, we used cosine similarity and employed the two large language models employed in the previous experiment. We measured the similarity of the embedded representations of each pair and then compared that with the original score labeled in the STS benchmark. The idea is that the more the two lists between 1,500 pairs of sentences are correlated, the better the representations are. As seen in Figure 3 (c), the representations from our proposed disentanglment approach is able to show higher values on both Kendall and Pearson correlations on both large

[2] https://huggingface.co/datasets/PhilipMay/stsb_multi_mt

language model. For instance, the Pearson and Kendall correlations improved by 20% and 33%, respectively on the BERT language model. A similar observation can be made on the RoBERTa language model as well. This reinforces our finding in the previous experiment that our disentanglement approach is able to not only capture query difficulty effectively but also capture query content semantics successfully.

5 Concluding Remarks

In this paper, we proposed a neural disentanglement approach to isolate query content semantics from other aspects of the query that may impact its retrieval effectiveness. Our approach employs contrastive representation learning and point-wise prediction to perform neural disentanglement and to accurately estimate query performance. Extensive experiments on four benchmark query sets demonstrate that our approach outperforms state-of-the-art QPP baselines. In addition, we empirically show that the disentangled content semantics representations provide a more accurate account of the information expressed by the query when compared to representations offered by widely used language models, such as, BERT and RoBERTa. While our proposed approach makes significant contributions, it has certain limitations that we plan to address in future works, which we outline as follows:

- A key area for improvement is enhancing the interpretability of our model during training and testing. Our efforts to identify metrics that assess the impact of components on disentanglement have proven to be challenging, as most existing metrics are highly case-specific and cannot be used in every disentangled representation learning model [66, 67]. We plan to further investigate this area and develop metrics that provide clearer insights into how different components affect disentanglement levels and, in turn, how these levels influence QPP accuracy.
- Another research direction we aim to explore is extending beyond separating content and difficulty to use the difficulty vectors for controlled text generation. With our successful disentangled representation for queries, we will investigate methods to modify the difficulty representation, enabling the generation of easier queries that are optimized for an IR system.

References

[1] Hambarde, K.A., Proença, H.: Information retrieval: Recent advances and beyond. IEEE Access **11**, 76581–76604 (2023)

[2] Arabzadeh, N., Mitra, B., Bagheri, E.: Ms marco chameleons: Challenging the ms marco leaderboard with extremely obstinate queries. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 4426–4435 (2021)

[3] Sarnikar, S., Zhang, Z., Zhao, J.L.: Query-performance prediction for effective query routing in domain-specific repositories. JASIST **65**(8), 1597–1614 (2014)

[4] Ganguly, D., Yilmaz, E.: Query-specific variable depth pooling via query performance prediction. In: SIGIR, pp. 2303–2307 (2023)

[5] Pal, D., Ganguly, D.: Effective query formulation in conversation contextualization: A query specificity-based approach. In: ICTIR, pp. 177–183 (2021)
[6] Tonellotto, N., Macdonald, C., Ounis, I.: Efficient and effective retrieval using selective pruning. In: WSDM, pp. 63–72 (2013)
[7] Deveaud, R., Mothe, J., Ullah, M.Z., Nie, J.-Y.: Learning to adaptively rank document retrieval system configurations. ACM TOIS **37**(1), 1–41 (2018)
[8] Deveaud, R., Mothe, J., Nie, J.-Y.: Learning to rank system configurations. In: CIKM, pp. 2001–2004 (2016)
[9] Khramtsova, E., Zhuang, S., Baktashmotlagh, M., Zuccon, G.: Leveraging llms for unsupervised dense retriever ranking. ArXiv (2024)
[10] Raiber, F., Kurland, O.: Query-performance prediction: setting the expectations straight. In: SIGIR (2014)
[11] Arabzadeh, N., Seifikar, M., Clarke, C.L.: Unsupervised question clarity prediction through retrieved item coherency. In: CIKM, pp. 3811–3816 (2022)
[12] Roitman, H., Erera, S., Feigenblat, G.: A study of query performance prediction for answer quality determination. In: ICTIR (2019)
[13] Zamani, H., Dumais, S., Craswell, N., Bennett, P., Lueck, G.: Generating clarifying questions for information retrieval. In: WWW (2020)
[14] Meng, C., Arabzadeh, N., Askari, A., Aliannejadi, M., Rijke, M.: Ranked list truncation for large language model-based re-ranking. ArXiv (2024)
[15] He, B., Ounis, I.: Inferring query performance using pre-retrieval predictors. In: SPIRE (2004). Springer
[16] Zhao, Y., Scholer, F., Tsegay, Y.: Effective pre-retrieval query performance prediction using similarity and variability evidence. In: ECIR (2008)
[17] Arabzadeh, N., Zarrinkalam, F., Jovanovic, J., Bagheri, E.: Geometric estimation of specificity within embedding spaces. In: CIKM, pp. 2109–2112 (2019)
[18] Datta, S., Ganguly, D.e.a.: A relative information gain-based query performance prediction framework with generated query variants. ACM TOIS **41**(2) (2022)
[19] Arabzadeh, N., Bigdeli, A.e.a.: Matches made in heaven: Toolkit and large-scale datasets for supervised query reformulation. In: CIKM, pp. 4417–4425 (2021)
[20] Benham, R., Culpepper, J.S., Gallagher, L., Lu, X., Mackenzie, J.M.: Towards efficient and effective query variant generation. In: DESIRES (2018)
[21] Zuccon, G., Palotti, J., Hanbury, A.: Query variations and their effect on comparing information retrieval systems. In: CIKM (2016)
[22] Izacard, G., Caron, M.e.a.: Unsupervised dense information retrieval with contrastive learning. arXiv:2112.09118 (2021)
[23] Bui, N.D., Yu, Y., Jiang, L.: Self-supervised contrastive learning for code retrieval and summarization via semantic-preserving transformations. In: SIGIR (2021)
[24] Li, Y., Liu, Z., Xiong, C., Liu, Z.: More robust dense retrieval with contrastive dual learning. In: ICTIR, pp. 287–296 (2021)
[25] Yang, N., Wei, F., Jiao, B., Jiang, D., Yang, L.: xmoco: Cross momentum contrastive learning for open-domain question answering. In: ACL — IJCNLP, pp. 6120–6129 (2021)

[26] Mackie, I., Dalton, J., Yates, A.: How deep is your learning: the dl-hard annotated deep learning dataset. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (2021)
[27] Hofstätter, S., Althammer, S., al., M.S.: Improving efficient neural ranking models with cross-architecture knowledge distillation. arXiv:2010.02666 (2020)
[28] Faggioli, G., Zendel, O., Culpepper, J.S., Ferro, N., Scholer, F.: smare: A new paradigm to evaluate and understand query performance prediction methods. Information Retrieval Journal **25**(2), 94–122 (2022)
[29] Hauff, C., Hiemstra, D., Jong, F.: A survey of pre-retrieval query performance predictors. In: CIKM (2008)
[30] Carmel, D., Yom-Tov, E.: Estimating the query difficulty for information retrieval. In: SIGIR (2010)
[31] Zhou, Y., Croft, W.B.: Query performance prediction in web search environments. In: SIGIR, pp. 543–550 (2007)
[32] Tao, Y., Wu, S.: Query performance prediction by considering score magnitude and variance together. In: CIKM, pp. 1891–1894 (2014)
[33] Pérez-Iglesias, J., Araujo, L.: Standard deviation as a query hardness estimator. In: SPIRE, pp. 207–212 (2010). Springer
[34] Shtok, A., Kurland, O., Carmel, D.: Using statistical decision theory and relevance models for query-performance prediction. In: SIGIR, pp. 259–266 (2010)
[35] Shtok, A., Kurland, O., Carmel, D., Raiber, F., Markovits, G.: Predicting query performance by query-drift estimation. TOIS **30**(2), 1–35 (2012)
[36] He, B., Ounis, I.: Query performance prediction. Information Systems **31**(7), 585–594 (2006)
[37] He, J., Larson, M., Rijke, M.: Using coherence-based measures to predict query difficulty. In: ECIR, pp. 689–694. Springer, ??? (2008)
[38] Hauff, C., Azzopardi, L., Hiemstra, D., Jong, F.: Query performance prediction: Evaluation contrasted with effectiveness. In: ECIR, pp. 204–216 (2010). Springer
[39] Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: SIGIR. SIGIR '02. ACM, ??? (2002)
[40] He, B., Ounis, I.: Query performance prediction. Information Systems **31**(7) (2006)
[41] Zhou, J., Troyanskaya, O.G.: Predicting effects of noncoding variants with deep learning–based sequence model. Nature Methods **12**(10), 931–934 (2015)
[42] Arabzadeh, N.e.a.: Neural embedding-based specificity metrics for pre-retrieval query performance prediction. IP&M Journal **57**(4) (2020)
[43] Roy, D., Ganguly, D., Mitra, M., Jones, G.J.: Estimating gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction. IPM **56**(3), 1026–1045 (2019)
[44] Arabzadeh, N., Zarrinkalam, F., Jovanovic, J., Bagheri, E.: Neural embedding-based metrics for pre-retrieval query performance prediction. In: ECIR (2020)
[45] Hashemi, H., Zamani, H., Croft, W.B.: Performance prediction for non-factoid question answering. In: ICTIR, pp. 55–58 (2019)

[46] Arabzadeh, N., Khodabakhsh, M., Bagheri, E.: Bert-qpp: Contextualized pre-trained transformers for query performance prediction. In: CIKM (2021)
[47] Zamani, H., Croft, W.B., Culpepper, J.S.: Neural query performance prediction using weak supervision from multiple signals. In: SIGIR, pp. 105–114 (2018)
[48] Datta, S., MacAvaney, S., Ganguly, D., Greene, D.: A 'pointwise-query, listwise-document' based query performance prediction approach. In: SIGIR (2022)
[49] Yang, Z., Hu, Z., Dyer, C., Xing, E.P., Berg-Kirkpatrick, T.: Unsupervised text style transfer using language models as discriminators. In: NeurIPS (2018)
[50] Shen, T., Lei, T., Barzilay, R., Jaakkola, T.: Style transfer from non-parallel text by cross-alignment. In: NeurIPS. NIPS'17 (2017)
[51] John, V., Mou, L., Bahuleyan, H., Vechtomova, O.: Disentangled representation learning for non-parallel text style transfer. In: ACL (2018)
[52] Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E.P.: Toward controlled generation of text. In: Precup, D., Teh, Y.W. (eds.) ICML. PMLR, vol. 70 (2017)
[53] Fu, Z., Tan, X., Peng, N., Zhao, D., Yan, R.: Style transfer in text: Exploration and evaluation. AAAI **32**(1) (2018)
[54] Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: ICML. PMLR (2016)
[55] Ngweta, L., Maity, S., Gittens, A., Sun, Y., Yurochkin, M.: Simple disentanglement of style and content in visual representations. In: ICML (2023)
[56] Kingma, D., Welling, M.: Auto-encoding variational bayes. arXiv:1312.6114 (2013)
[57] Sha, L., Lukasiewicz, T.: Text attribute control via closed-loop disentanglement. Transactions of the Association for Computational Linguistics **12**, 190–209 (2024) https://doi.org/10.1162/tacl_a_00640
[58] Colombo, P., Staerman, G., Noiry, N., Piantanida, P.: Learning disentangled textual representations via statistical measures of similarity. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2614–2630. Association for Computational Linguistics, Dublin, Ireland (2022). https://doi.org/10.18653/v1/2022.acl-long.187 . https://aclanthology.org/2022.acl-long.187
[59] Mahapatra, D., Jimeno Yepes, A.J., Kuanar, S., Roy, S., Bozorgtabar, B., Reyes, M., Ge, Z.: Class Specific Feature Disentanglement and Text Embeddings for Multi-label Generalized Zero Shot CXR Classification, pp. 276–286. Springer, ??? (2023). https://doi.org/10.1007/978-3-031-43895-0_26
[60] Kwok, K.L.: A new method of weighting query terms for ad-hoc retrieval. In: SIGIR, pp. 187–195 (1996)
[61] Hauff, C.: Predicting the effectiveness of queries and retrieval systems. In: SIGIR Forum, vol. 44, p. 88 (2010)
[62] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
[63] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv (2018)

[64] Nogueira, R., Lin, J., Epistemic, A.: From doc2query to doctttttquery. Online preprint **6**(2) (2019)
[65] Chan, C.-M., Xu, C., Yuan, R., Luo, H., Xue, W., Guo, Y., Fu, J.: Rq-rag: Learning to refine queries for retrieval augmented generation. arXiv preprint arXiv:2404.00610 (2024)
[66] Carbonneau, M.-A., Zaïdi, J., Boilard, J., Gagnon, G.: Measuring disentanglement: A review of metrics. IEEE Transactions on Neural Networks and Learning Systems **35**(7), 8747–8761 (2024) https://doi.org/10.1109/TNNLS.2022.3218982
[67] Do, K., Tran, T.: Theory and evaluation metrics for learning disentangled representations. arXiv preprint arXiv:1908.09961 (2019)