Noisy Perturbations for Estimating Query Difficulty in Dense Retrievers

Negar Arabzadeh
University of Waterloo
narabzad@uwaterloo.ca

Radin Hamidi Rad
Toronto Metropolitan University
radin@torontomu.ca

Maryam Khodabakhsh
Shahrood University of Technology
m_khodabakhsh@shahroodut.ac.ir

Ebrahim Bagheri
Toronto Metropolitan University
bagheri@torontomu.ca

ABSTRACT

Query Performance Prediction (QPP), is concerned with assessing the retrieval quality of a ranking method for an input query. Most traditional unsupervised frequency-based models and many recent supervised neural methods have been designed specifically for predicting the performance of sparse retrievers such as BM25. In this paper we propose an unsupervised QPP method for *dense neural retrievers* which operates by redefining the well-known concept of *query robustness* i.e., a more robust query to perturbations is an easier query to handle. We propose to generate query perturbations for measuring query robustness by systematically injecting noise into the contextualized neural representation of each query. We then compare the retrieved list for the original query with that of the perturbed query as a way to measure query robustness. Our experiments on four different query sets including MS MARCO, TREC Deep Learning track 2019 and 2020 and TREC DL-Hard show consistently improved performance on linear and ranking correlation metrics over the state of the art.

CCS CONCEPTS

• **Information systems → Evaluation of retrieval results**.

KEYWORDS

Information Retrieval, Query Performance Prediction

ACM Reference Format:

Negar Arabzadeh, Radin Hamidi Rad, Maryam Khodabakhsh, and Ebrahim Bagheri. 2023. Noisy Perturbations for Estimating Query Difficulty in Dense Retrievers. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3583780.3615270

1 INTRODUCTION

Despite advances on tasks such as ad hoc retrieval [36, 58, 60], conversational search [24, 62], and question answering [32, 59], recent research has shown there is still much room for improvement especially on *harder queries* [2]. In order to identify such hard-to-satisfy queries, the Information Retrieval (IR) community has explored the task of Query Performance Prediction (QPP), which aims to estimate the quality of the retrieved list of documents for a given query [1, 5, 6, 15, 28, 33, 34, 43, 66].

Background Literature. Earliest post-retrieval QPP methods often relied on frequency-based statistical characteristics of each query and its associated list of retrieved documents [15, 66]. These statistical characteristics included measures such as the similarity between the query and the retrieved documents [52], the divergence between the retrieved documents and the corpus [16], and the distribution of the relevance scores obtained for the retrieved documents, to name a few [15, 52, 64]. More recently, several supervised QPP such as `NQA-QPP` [27], `BERT-QPP` [1] and `qppBERT-PL` [20] [18] have shown to outperform traditional QPP methods for sparse retrievers [20, 33]; however, they all require a large number of training instances (e.g., the MS MARCO dataset) [1, 19, 20, 22, 39].

Context of Our Work. With the growing influence of neural-based models [21, 55], dense retrievers are now the state-of-the-art baselines for many tasks in IR [26, 32, 36, 40, 42, 59]. Given most existing QPP methods are designed for sparse retrievers (except few recent ones such as [53]), they are primarily using statistics that hint at how sparse retrievers function [12]. In contrast, while dense retrievers may implicitly consider such statistics when trained on a corpus, they are less sensitive to frequency statistics and primarily rely on the semantics and context of the query and the document collection [4]. In addition, it has been shown that score-based QPP metrics would not necessarily work well when predicting the performance of neural models primarily because the distribution of retrieval scores in neural models is different from sparse retrievers [19, 23, 39, 53]. To the best of our knowledge, there are only a few studies that have explored QPP for dense retrievers such as Singh et al. [53] that employ pairwise rank preference probabilities obtained from strong re-rankers.

Overview of Approach. An ideal QPP method for dense retrievers would be one that would take advantage of the characteristics of dense retrievers in order to accurately determine query performance. The major characteristic of dense retrievers that differentiates them from sparse retrievers is the fact that they encode queries and documents within a low-dimensional embedding space. Thus, we focus on embedding representations of queries and documents to perform QPP. The intuition behind our work is based on the notion of *query robustness* [61]. The idea of query robustness has been explored in the context of QPP for sparse retrievers, often based on

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615270

the notion of pseudo-relevance feedback and reference lists [19, 46–48, 51, 56, 66]. A query is considered to be *robust* if its performance is not significantly impacted by perturbations applied to the query [12]. We propose a method, called `Dense-QPP`, to perform QPP for dense retrievers by generating query perturbations based on the embedding representations of input queries. Earlier works on QPP for sparse retrievers apply query perturbations by rewriting the initial query as sparse retrievers deal with keyword-based representation of the query [65, 67]. However, we propose to generate query perturbations by modifying the embedding representations of each query as this is the representation used by dense neural retrievers. In our approach, a query perturbation would be obtained by systematically injecting noise into the embedding representations of queries by which we are in essence slightly moving the query away from its original position in the embedding space to a new position. A less robust query would be one that would experience a noticeable change in its retrieval. We take such differences as a sign of query difficulty.

Summary of Experiments. We perform extensive experiments on four widely used query collections on the MS MARCO passage collection as well as TREC DL query sets from 2019 and 2020 [13, 14, 41] as well as DL-Hard query set which includes more challenging queries [38]. We show that `Dense-QPP` exhibits a more consistent and improved performance compared to the state-of-the-art QPP methods when predicting the performance of two SOTA first-stage dense retrievers, i.e, S-BERT [44] and ANCE [58].

2 PROPOSED APPROACH

Let $Q = \{q_i\}$ be a set of queries, and $C = \{d_j | 1 \leq j \leq N\}$ represent the corpus which consists of N documents. In the retrieval task, we let $D_{q_i} \subseteq C; D_{q_i} \neq \emptyset$ be the set of the ranked list of retrieved documents for a query q_i. We formulate a retriever function F as $D_{q_i} = F(q_i, C)$. The QPP function $\phi(q_i)$ is responsible for predicting the quality of retrieved documents D_{q_i} produced by the retriever $F(q_i, C)$ by estimating the rank-based evaluation metric $\widehat{M}$. Let M be the original quality of the retrieved document list. Then $\phi(q_i)$ aims to minimize the gap between the predicted performance $\widehat{M}$ and the actual performance of the retrieved results M. Common IR metrics can be plugged into $\widehat{M}$ and M, e.g., reciprocal rank.

A dense retriever such as F_{dense} encodes a query and its set of retrieved documents as embedding representations denoted by $E(q_i)$ and $E(d_j)$, respectively. With a dense retriever F_{dense}, we retrieve a ranked list of documents D_q^{dense} for a given query q as $D_{q_i}^{dense} = F_{dense}(E(q_i), C)$. Given F_{dense} as the dense retriever, we will generate a perturbed set of queries $\widehat{Q} = \{\widehat{q_i}\}$ that would allow us to measure the robustness of the queries in latent space. Since the representations of the queries are in the embedding space, we generate query perturbations in a similar space. Therefore, we propose a neural architecture that injects noise, in the form of Additive White Gaussian Noise (AWGN), into the representations of each query to produce query perturbations. Methods based on query perturbation measure the robustness of a query by the contrast between the set of retrieved documents for the original query and its perturbed version. We chose AWGN over other conventional noise forms due to its characteristics: (1) AWGN has a uniform power spectral density across frequency. This means by using AWGN embedding vector elements will receive noise with different frequencies in a uniform amount; and, (2) AWGN has a Gaussian distribution, which is desirable as noisy perturbations in the real world are modelled by Gaussian distribution [8, 37].

Let $\mathcal{X}_i = [X_1, X_2, \cdots, X_n]$ be the embedding representation of the input query q_i i.e., $\mathcal{X}_i = E(q_i)$. We propose the following neural architecture to produce query perturbations where $\mathcal{G}$ is the Gaussian noise layer responsible for adding AWGN to the input vector, μ is average and σ^2 is the variance of the added noise.:

$$\begin{aligned} h_{q_i} &= \mathcal{G}(\mathcal{X}_i, \mu, \sigma^2) \\ \widehat{\mathcal{X}_i} &= \mathcal{F}(h_{q_i}, \mathcal{W}, b) \end{aligned} \tag{1}$$

Here, $\widehat{\mathcal{X}_i}$ is the generated perturbation for query q_i. We denote the weight and bias matrices between Gaussian and output layers with $\mathcal{W}$ and b, respectively. Here, $\mathcal{F}$ is the activation function of the output layer. For the sake of simplicity, we use a linear activation function. The characteristics of the added noise are controlled using the μ and σ^2 parameters. Ultimately, we are looking for white Gaussian noise with an even distribution that does not lean the dense representation of the queries towards a particular direction. Therefore, we use zero-mean as suggested in [10, 11]. In order to determine the appropriate value for σ^2, we adopt the concept of Signal-to-Noise Ratio (SNR) [30] and refer to it as the Embedding-to-Noise Ratio (ENR) in the context of our work. Considering the initial embedding of a query as the input signal, we can calculate the proper amount of variance for the additive noise by fixing the value of ENR:

$$ENR = \frac{P_{embedding}}{P_{noise}} \tag{2}$$

where $P_{embedding}$ and P_{noise} are the second moment values of the vectors. Since the added noise is AWGN, we reformulate P_{noise} as:

$$P_{noise} = \mathbb{E}[{x^2}_{noise}] = \mu^2 + \sigma^2 \tag{3}$$

where x_{noise} are values of the noise vector drawn from Gaussian probability distribution and $\mathbb{E}[]$ is the expected value of a given variable. Since $\mu = 0$, we reformulate Equation 3 as $P_{noise} = \sigma^2$. Using the same technique, $P_{embedding}$ can be shown as:

$$P_{embedding} = \mathbb{E}[{x^2}_{embedding}] \tag{4}$$

where x is the elements of the embedding vector of the query. Given Equations 2-4, σ^2 of the additive noise can be calculated as:

$$\gamma = \frac{\mathbb{E}[{x^2}_{embedding}]}{\sigma^2} \tag{5}$$

where γ is the desired value that represents the ratio of ENR over the entire process. We can reformulate Equation 5 to have σ^2 which is the only parameter that controls noise on one side:

$$\sigma^2 = \frac{\mathbb{E}[{x^2}_{embedding}]}{\gamma} \tag{6}$$

We note that the same noise is used for all the queries, to ensure the distribution of the amplitude of the noise is consistent across all queries. Let us assume $D_{q_i}^{dense}$ is the list of documents for the original query, and $\widehat{D_{q_i}^{dense}}$ is the list of documents for the perturbed query, we consider the similarity between $\widehat{D_{q_i}^{dense}}$ and $D_{q_i}^{dense}$ as an indicator of query performance and refer it as `Dense-QPP` metric:

$$Dense - QPP(q_i, F_{dense}, C) = Sim(\widehat{D_{q_i}^{dense}}, D_{q_i}^{dense}) \tag{7}$$

We adopt the *Ranked Bias Overlap* [57] metric to compute *Sim*.

Table 1: Performance on MS MARCO, DL-2019, DL-2020 and DL-Hard dataset in terms of Pearson ρ ($P-\rho$), Kendall ($K-\tau$) measures when predicting S-BERT (on the left) and ANCE (on the right). *Italic* values indicate a statistically non-significant correlation with a p-value < 0.05. Bold and underline values indicate the highest and the runner up correlation in each column.

	S-BERT								ANCE							
	MS MARCO		DL-2019		DL-2020		DL-Hard		MS MARCO		DL-2019		DL-2020		DL-Hard	
	$P-\rho$	$K-\tau$	$P-\rho$	$K-\tau$	$P-\rho$	$K-\tau$	$P-\rho$	$K-\tau$	$P-\rho$	$K-\tau$	$P-\rho$	$K-\tau$	$P-\rho$	$K-\tau$	$P-\rho$	$K-\tau$
Clarity	0.065	0.053	0.217	0.111	0.196	0.137	0.232	0.110	0.161	0.196	0.353	0.237	0.281	0.215	0.221	0.230
QF	0.175	0.115	0.071	0.022	0.148	*0.029*	0.044	0.051	0.071	*0.034*	0.129	0.098	0.283	0.257	0.155	0.118
NQC	0.219	0.202	0.560	0.419	0.336	0.228	0.418	0.276	0.109	0.140	0.504	0.335	0.442	0.328	0.235	0.300
WIG	0.048	0.032	0.139	0.071	0.153	0.032	0.093	0.072	0.100	0.100	0.159	0.120	0.230	0.195	0.166	0.133
n($\sigma_{\%}$)	0.128	0.128	0.501	0.361	0.242	0.158	0.400	0.259	0.030	0.042	0.361	0.233	0.199	0.181	0.242	0.197
SMV	0.183	0.127	0.577	$\underline{0.428}$	0.360	$\underline{0.246}$	0.396	$\underline{0.314}$	0.109	0.152	0.518	0.337	0.417	0.328	0.174	0.290
UEF$_{NQC}$	0.218	0.166	$\underline{0.607}$	$\underline{0.428}$	0.336	0.228	$\underline{0.441}$	0.298	0.198	0.219	$\underline{0.520}$	$\underline{0.350}$	**0.458**	**0.348**	0.229	$\underline{0.309}$
Neural-QPP	0.060	0.055	0.209	0.057	0.152	0.015	0.232	0.080	0.073	0.060	0.047	0.004	0.220	0.087	0.142	0.063
P$_{clarity}$	0.213	0.135	0.428	0.314	0.183	0.201	0.088	0.053	0.125	0.086	0.383	0.247	0.209	0.308	0.157	0.172
NQA-QPP	0.267	0.216	0.269	0.129	0.221	0.159	0.113	0.240	0.267	$\underline{0.221}$	0.115	0.140	0.147	0.152	**0.334**	0.264
BERT-QPP	$\underline{0.292}$	0.223	0.334	0.143	$\underline{0.378}$	0.273	0.435	0.181	$\underline{0.271}$	0.218	0.144	0.165	0.362	0.268	0.213	0.143
qppBERT-PL	0.277	$\underline{0.230}$	0.299	0.131	0.344	0.224	0.405	0.171	0.251	0.208	0.229	0.189	0.313	0.205	0.303	0.254
Deep-QPP	*0.021*	*0.016*	0.139	0.103	0.262	0.197	0.096	0.048	0.130	0.132	0.182	0.195	0.195	0.126	0.154	0.131
QPP-PRP	*0.010*	*0.014*	0.275	0.203	0.181	0.142	0.181	0.098	*0.015*	0.014	0.296	0.186	0.320	0.269	0.115	0.104
Dense-QPP	**0.335**	**0.296**	**0.683**	**0.437**	**0.390**	**0.274**	**0.465**	**0.339**	**0.296**	**0.242**	**0.528**	**0.363**	$\underline{0.443}$	$\underline{0.332}$	$\underline{0.315}$	**0.310**

3 EXPERIMENTAL SETUP

Codebase. For reproducibility, our code and data is publicly available at https://github.com/Narabzad/Dense-QPP

Datasets: We evaluate the performance of `Dense-QPP` as well as the SOTA QPP baselines on queries from four widely adopted datasets including 6,980 queries in small dev set of MS MARCO passage collection [41], TREC Deep Learning tracks from 2019 and 2020, namely DL-2019 [13] and DL-2020 [14] as well as DL-Hard [38]. The main difference between the MS MARCO collection and the other collections is that MS MARCO has sparse labels, i.e., only less than 10% of queries have more than one relevant judged document per query [3]. The other three collections, i.e., DL-2019, DL-2020 and DL-Hard, are accompanied with a large number of human-labelled relevance judgements per query. This is important since it is possible to have a higher confidence in the results obtained from queries that have a higher number of relevant documents. DL-2019 includes 43 thoroughly judged queries and DL-2020 consists of 53 extensively judged queries. In addition, we consider DL-Hard which includes 50 queries. We consider the official evaluation metric for each dataset, i.e., MRR@10 for MS MARCO and nDCG@10 for DL-2019, DL-2020 and DL-Hard as the target metric to be predicted.

Evaluation Metrics: The common approach for evaluating a QPP method is to use correlation metrics between the ranked list of queries based on their predicted difficulty and their actual performance [12, 15]. We measure Kendall and Pearson correlations in which higher correlation values reflect more accurate performance prediction.

Retrievers: We adopt two widely used Sentence-BERT (S-BERT) [44] and ANCE [58] retrievers. S-BERT and ANCE have shown strong retrieval performance as well as low computational overhead compared to other neural-based retrievers[29, 45]. We use pre-trained models on MS MARCO from Hugging Face to encode the four query sets and the MS MARCO passage collection and perform the retrieval. In general, these bi-encoder-based dense retrievers encode both the query and the documents into fixed-length vectors using a transformer-based neural network. These encoded vectors are then compared using a similarity metric, such as cosine similarity, to retrieve the most relevant documents for a given query. For further information, we refer to the original papers [32, 44, 58].

Baselines: We compare our proposed `Dense-QPP` method against the state-of-the-art supervised and unsupervised post-retrieval QPP methods [1, 12, 27]. The unsupervised traditional term-statistics QPP baselines we consider in this paper include the `WIG` [66], `Clarity` [15], `QF` [66], `NQC` [52],UEF_{NQC} [50] and `SMV` [54]. We also consider $P_{clarity}$ [49] which is initially a pre-retrieval method but it could leverage `NQC` to interpolate with and be considered as a post-retrieval QPP method. $n(\sigma_{\%})$ [17]. More recent supervised QPP methods have outperformed their unsupervised counterparts on various query sets and different document collections [21, 27, 63]. The supervised QPP methods, which we have employed in this paper include `Neural-QPP` [63], `NQA-QPP` [27] , `BERT-QPP` [1], `qppBERT-PL` [20], `Deep-QPP` [18]. Lastly, we include the recently proposed `QPP-PRP` [53], similar to our proposed method, `QPP-PRP` is unsupervised and is the only baseline originally designed for QPP on dense retrievers.

Hyperparameter Setting: Based on the Central Limit Theorem (CLT) [25] and to generate white Gaussian noise, we sample multiple noises to ensure that generated noises accurately represent the probability distribution function of white Gaussian noise. As suggested in [7, 9, 35], we have sub-sampled 30 noises. We generate the Gaussian noise vectors by setting $\mu = 0$ and selecting σ w.r.t to γ values. We perform an element-wise addition of the noise vector to the embedded query vector and retrieve the perturbed query from the embedded document index using the Faiss library [31]. Additionally, as suggested in [1], we tune the hyper-parameters of all of the baselines as well as our method and the number of top-K retrieved documents $K \in \{100, 200, 300, ..., 1000\}$ for TREC DL-2019 on TREC DL-2020 and vice-versa. For DL-Hard, we tune the hyper parameters on non-overlapping queries from DL-2019 and DL-2020 and for MS MARCO dev set, we tune it on 5,000 randomly sampled queries from the remainder of the MS MARCO dev set (excluding 6,980 queries in MS MARCO small dev).

4 RESULTS

Table 1 reports the results of our proposed `Dense-QPP` method as well as the baselines based on Pearson ρ linear and Kendall τ ranking correlations. Based on the results, we make several observations: **(1)** Among the unsupervised baselines, those that are based on the distribution of retrieval scores perform better than the others. For instance, NQC, SMV and $n(\sigma^2_{\%})$ show a better performance compared to Clarity and QF, which were not even able to exhibit statistically significant correlation with the actual query performance in some cases. **(2)** Within the supervised QPP baselines, Neural-QPP suffers from extremely low correlation values. We hypothesize that this might be due to the fact that Neural-QPP is built from weak signals coming from unsupervised QPP methods, which are themselves not strong signals for QPP in the context of dense retrievers. In addition, Neural-QPP requires large amounts of training data and has also previously shown poor performance when there is limited training data available [1]. Similarly, Deep-QPP while it has shown to be effective in estimating the difficulty of queries with sparse retrievers, it failed to show consistent and promising performance for dense retrievers. On the other hand, NQA-QPP, BERT-QPP and qppBERT-PL show higher degrees of correlation with actual query performance, specifically on MS MARCO. However, both of these approaches lack consistency across different query sets. **(3)** Our proposed Dense-QPP outperforms all of the baselines on all query sets except DL-2020 for ANCE. On the DL-2020 query set, the ranking correlations of UEF_{NQC} is *slightly* higher than Dense-QPP. For DL-Hard, we also note that NQA-QPP performs slightly better in terms of Pearson ρ; however, even in this circumstance, the ranking correlation obtained by Dense-QPP on DL-Hard outperforms that of NQA-QPP. In both cases where Dense-QPP shows inferior performance w.r.t. the baseline, the performances have not shown statistically significant difference through paired t-test with a p-value of 0.05. **(4)** Our proposed method shows the most consistent performance across all the query sets, i.e., it is the only method that shows consistently high performance on all datasets and all correlation metrics when predicting the performance of both QPP methods. **(5)** We mention that Dense-QPP is generalizable across different neural rankers. To show this, as seen in Table 1, while in general, the baseline QPP methods were more successful on ANCE compared to S-BERT; however, our proposed Dense-QPP method consistently outperforms the baselines on both ANCE and S-BERT, indicating its generalizability. **(6)** Lastly, we note that among the baselines, QPP-PRP was the only one that was originally designed for dense retrievers. We show that QPP-PRP is not able to show consistent performance on all datasets e.g., on MS MARCO, its correlation is not statistically significant. In addition, our proposed Dense-QPP shows superior performance w.r.t this baseline on all the datasets and across both of the retrievers.

It is important to note that by comparing the reported correlation values of the baseline methods when predicting the performance of sparse retrievers on MS MARCO and TREC DL 2019 and 2020, as reported in [1], compared to their performance on predicting the performance of dense retrievers, we observe that all the baselines show relatively lower correlation when predicting the performance of dense retrievers. We hypothesize that this may be because **(i)** the retrieval effectiveness of dense retrievers is often much higher compared to sparse retrievers [32, 36, 39], and hence, the better performance of a ranker over a range of queries makes it hard to distinguish between these queries and consequently, makes the QPP task more difficult on dense retrievers; and, **(ii)** other than the supervised methods that can be used equally for both dense and sparse retrievers, the other QPP metrics were not specifically intended for predicting the performance of dense retrievers as they

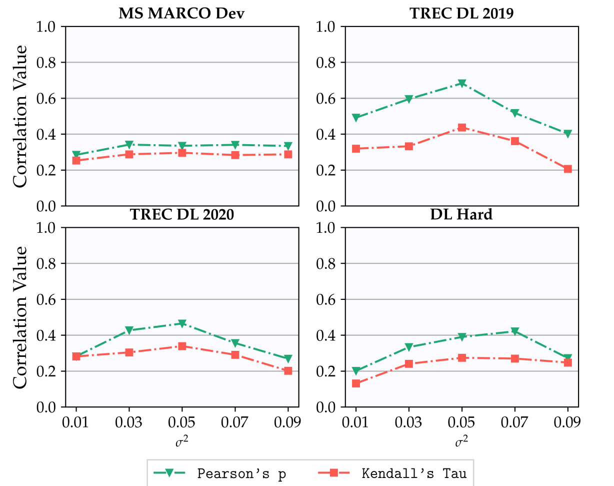

Figure 1: Impact of noise variance on the performance of Dense-QPP. The variance of injected noise (X-axis) vs performance (Kendall and Pearson) of Dense-QPP (Y-axis).

leverage signals that are based on corpus statistics, which would not be strong indicators for the performance of dense retrievers. However, our proposed Dense-QPP method is intentionally designed for predicting the performance of dense retrievers by injecting noise into the embedding representation of queries and documents.

Further, we investigate the impact of the distribution of the injected noise on the performance of Dense-QPP. We sweep the variance of the injected noise in Equation 6 and depict the results in Figure 1 for S-BERT. We do not sweep the *mean* as the mean should always be set to zero as discussed in Equation 4. As shown in Figure 1, by increasing the noise variance, the correlation between the dense retriever's actual performance and the predicted performance of Dense-QPP increases. However, after a certain degree of increase, the prediction performance would show a downward trajectory. We hypothesize that as the degree of noise increases, the alternative query starts to become too far from the original query. As such, the retrieved documents from the noisy query will lose their resemblance to those from the original query; therefore leading to decreased performance. Similarly, when the noise level is below a certain level, the retrieved results of the noisy query would not differ much from the original query and thus the predicted performance is low. However, by conducting sensitivity analysis on the four datasets, we observe that a variance of $5-7\%$ for the injected noise results in the best performance. As shown in our experiments, the appropriate value for σ can be effectively identified through hyperparameter tuning on a held-out set or cross-validation.

5 CONCLUDING REMARKS

We propose an unsupervised QPP method specifically for predicting the performance of *dense retrievers.* Our work is motivated by the concept of query robustness for measuring query difficulty. We measure query robustness by generating query perturbations for an input query. To generate perturbations, we introduce a systematic approach for injecting noise into the embedding representation of each query derived from the neural ranker. We show that our proposed approach has a consistently better performance on two different neural rankers compared to the state-of-the-art when predicting over four different query sets on MS MARCO V1 collection.

REFERENCES

[1] Negar Arabzadeh, Maryam Khodabakhsh, and Ebrahim Bagheri. 2021. BERT-QPP: Contextualized Pre-trained transformers for Query Performance Prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2857–2861.

[2] Negar Arabzadeh, Bhaskar Mitra, and Ebrahim Bagheri. 2021. MS MARCO Chameleons: Challenging the MS MARCO Leaderboard with Extremely Obstinate Queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4426–4435.

[3] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal* 25, 4 (2022), 365–385.

[4] Negar Arabzadeh, Xinyi Yan, and Charles L. A. Clarke. 2021. Predicting Efficiency/Effectiveness Trade-offs for Dense vs. Sparse Retrieval Strategy Selection. *CoRR* abs/2109.10739 (2021). arXiv:2109.10739 https://arxiv.org/abs/2109.10739

[5] Negar Arabzadeh, Fattane Zarrinkalam, Jelena Jovanovic, Feras Al-Obeidat, and Ebrahim Bagheri. 2020. Neural embedding-based specificity metrics for pre-retrieval query performance prediction. *Information Processing & Management* 57, 4 (2020), 102248.

[6] Negar Arabzadeh, Fattane Zarrinkalam, Jelena Jovanovic, and Ebrahim Bagheri. 2020. Neural embedding-based metrics for pre-retrieval query performance prediction. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*. Springer, 78–85.

[7] Moatasim A Barri. 2019. A simulation showing the role of central limit theorem in handling non-normal distributions. *American Journal of Educational Research* 7, 8 (2019), 591–598.

[8] Roger Berger and George Casella. 2001. *Statistical Inference* (2 ed.). Duxbury Press, Florence, AL.

[9] Harald Bergström. 1944. On the central limit theorem. *Scandinavian Actuarial Journal* 1944, 3-4 (1944), 139–153.

[10] A.B. Carlson and P.B. Crilly. 2009. *Communication Systems*. McGraw-Hill Education. https://books.google.ca/books?id=-hRGAQAAIAAJ

[11] A.B. Carlson, P.B. Crilly, and J.C. Rutledge. 2002. *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*. McGraw-Hill. https://books.google.ca/books?id=cSVGAQAAIAAJ

[12] David Carmel and Elad Yom-Tov. 2010. Estimating the query difficulty for information retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 2, 1 (2010), 1–89.

[13] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[14] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Ellen M Voorhees, and Ian Soboroff. 2021. TREC deep learning track: Reusable test collections in the large data regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2369–2375.

[15] Steve Cronen-Townsend, Yun Zhou, and W Bruce Croft. 2002. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 299–306.

[16] Steve Cronen-Townsend, Yun Zhou, and W Bruce Croft. 2006. Precision prediction based on ranked list coherence. *Information Retrieval* 9, 6 (2006), 723–755.

[17] Ronan Cummins, Joemon Jose, and Colm O'Riordan. 2011. Improved query performance prediction using standard deviation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 1089–1090.

[18] Suchana Datta, Debasis Ganguly, Derek Greene, and Mandar Mitra. 2022. Deep-qpp: A pairwise interaction-based deep learning model for supervised query performance prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 201–209.

[19] Suchana Datta, Debasis Ganguly, Mandar Mitra, and Derek Greene. 2022. A Relative Information Gain-based Query Performance Prediction Framework with Generated Query Variants. *ACM Transactions on Information Systems* 41, 2 (2022), 1–31.

[20] Suchana Datta, Sean MacAvaney, Debasis Ganguly, and Derek Greene. 2022. A'Pointwise-Query, Listwise-Document'based Query Performance Prediction Approach. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2148–2153.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[22] Guglielmo Faggioli, Nicola Ferro, Cristina Ioana Muntean, Raffaele Perego, and Nicola Tonellotto. 2023. A Geometric Framework for Query Performance Prediction in Conversational Search. (2023).

[23] Guglielmo Faggioli, Thibault Formal, Stefano Marchesin, Stéphane Clinchant, Nicola Ferro, and Benjamin Piwowarski. 2023. Query Performance Prediction for Neural IR: Are We There Yet?. In *European Conference on Information Retrieval*. Springer, 232–248.

[24] Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2022. Neural Approaches to Conversational Information Retrieval. *arXiv preprint arXiv:2201.05176* (2022).

[25] Peter Hall. 1984. Central limit theorem for integrated square error of multivariate nonparametric density estimators. *Journal of multivariate analysis* 14, 1 (1984), 1–16.

[26] Radin Hamidi Rad, Hossein Fani, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2023. A Variational Neural Architecture for Skill-Based Team Formation. *ACM Trans. Inf. Syst.* 42, 1, Article 7 (aug 2023), 28 pages. https://doi.org/10.1145/3589762

[27] Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2019. Performance Prediction for Non-Factoid Question Answering. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 55–58.

[28] Ben He and Iadh Ounis. 2006. Query performance prediction. *Information Systems* 31, 7 (2006), 585–594.

[29] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 113–122.

[30] Don H. Johnson. 2006. Signal-to-noise ratio. *Scholarpedia* 1, 12 (2006), 2088. https://doi.org/10.4249/scholarpedia.2088

[31] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[32] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[33] Maryam Khodabakhsh and Ebrahim Bagheri. 2021. Semantics-enabled query performance prediction for ad hoc table retrieval. *Information Processing & Management* 58, 1 (2021), 102399.

[34] Maryam Khodabakhsh and Ebrahim Bagheri. 2023. Learning to rank and predict: Multi-task learning for ad hoc retrieval and query performance prediction. *Information Sciences* 639 (2023), 119015.

[35] Sang Gyu Kwak and Jong Hae Kim. 2017. Central limit theorem: the cornerstone of modern statistics. *Korean journal of anesthesiology* 70, 2 (2017), 144–156.

[36] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies* 14, 4 (2021), 1–325.

[37] Aidan Lyon. 2014. Why are normal distributions normal? *The British Journal for the Philosophy of Science* (2014).

[38] Iain Mackie, Jeff Dalton, and Andrew Yates. 2021. How Deep is your Learning: the DL-HARD Annotated Deep Learning Dataset. *CoRR* abs/2105.07975 (2021). arXiv:2105.07975 https://arxiv.org/abs/2105.07975

[39] Chuan Meng, Negar Arabzadeh, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Query Performance Prediction: From Ad-hoc to Conversational Search. *arXiv preprint arXiv:2305.10923* (2023).

[40] Hoang Nguyen, Radin Hamidi Rad, Fattane Zarrinkalam, and Ebrahim Bagheri. 2023. DyHNet: Learning dynamic heterogeneous network representations. *Inf. Sci.* 646 (2023), 119371. https://doi.org/10.1016/j.ins.2023.119371

[41] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[42] Radin Hamidi Rad, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2021. Retrieving Skill-Based Teams from Collaboration Networks. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2015–2019. https://doi.org/10.1145/3404835.3463105

[43] Fiana Raiber and Oren Kurland. 2014. Query-performance prediction: setting the expectations straight. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 13–22.

[44] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[45] Nils Reimers and Iryna Gurevych. 2020. The Curse of Dense Low-Dimensional Information Retrieval for Large Index Sizes. *CoRR* abs/2012.14210 (2020). arXiv:2012.14210 https://arxiv.org/abs/2012.14210

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

[47] Haggai Roitman. 2020. Systems and methods for query performance prediction using reference lists. (Aug. 11 2020). US Patent 10,740,338.

[48] Haggai Roitman and Oren Kurland. 2019. Query performance prediction for pseudo-feedback-based retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1261–1264.

[49] Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth JF Jones. 2019. Estimating gaussian mixture models in the local neighbourhood of embedded word

vectors for query performance prediction. *Information processing & management* 56, 3 (2019), 1026–1045.

[50] Anna Shtok, Oren Kurland, and David Carmel. 2010. Using statistical decision theory and relevance models for query-performance prediction. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 259–266.

[51] Anna Shtok, Oren Kurland, and David Carmel. 2016. Query performance prediction using reference lists. *ACM Transactions on Information Systems (TOIS)* 34, 4 (2016), 1–34.

[52] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems (TOIS)* 30, 2 (2012), 1–35.

[53] Ashutosh Singh, Debasis Ganguly, Suchana Datta, and Craig Macdonald. 2023. Unsupervised Query Performance Prediction for Neural Models utilising Pairwise Rank Preferences. *def* 1 (2023), 2.

[54] Yongquan Tao and Shengli Wu. 2014. Query performance prediction by considering score magnitude and variance together. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 1891–1894.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[56] Vishwa Vinay, Ingemar J Cox, Natasa Milic-Frayling, and Ken Wood. 2006. On ranking the effectiveness of searches. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 398–404.

[57] William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)* 28, 4 (2010), 1–38.

[58] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *CoRR* abs/2007.00808 (2020). arXiv:2007.00808 https://arxiv.org/abs/2007.00808

[59] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718* (2019).

[60] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. *CoRR* abs/1903.10972 (2019). arXiv:1903.10972 http://arxiv.org/abs/1903.10972

[61] Elad Yom-Tov, Shai Fine, David Carmel, and Adam Darlow. 2005. Metasearch and federation using query difficulty prediction. In *Proceedings of the ACM SIGIR Workshop on Predicting Query Difficulty, Salvador, Brazil*. Citeseer.

[62] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 829–838.

[63] Hamed Zamani, W Bruce Croft, and J Shane Culpepper. 2018. Neural query performance prediction using weak supervision from multiple signals. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 105–114.

[64] Yun Zhou. 2008. *Retrieval performance prediction and document quality*. University of Massachusetts Amherst.

[65] Yun Zhou and W Bruce Croft. 2006. Ranking robustness: a novel framework to predict query performance. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. 567–574.

[66] Yun Zhou and W Bruce Croft. 2007. Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 543–550.

[67] Yun Zhou and W Bruce Croft. 2008. Measuring ranked list robustness for query performance prediction. *Knowledge and Information Systems* 16 (2008), 155–171.