Query Performance Prediction: From Ad-hoc to Conversational Search

Chuan Meng
University of Amsterdam
Amsterdam, The Netherlands
c.meng@uva.nl

Negar Arabzadeh
University of Waterloo
Waterloo, Canada
narabzad@uwaterloo.ca

Mohammad Aliannejadi
University of Amsterdam
Amsterdam, The Netherlands
m.aliannejadi@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
m.derijke@uva.nl

ABSTRACT

Query performance prediction (QPP) is a core task in information retrieval. The QPP task is to predict the retrieval quality of a search system for a query without relevance judgments. Research has shown the effectiveness and usefulness of QPP for ad-hoc search. Recent years have witnessed considerable progress in conversational search (CS). Effective QPP could help a CS system to decide an appropriate action to be taken at the next turn. Despite its potential, QPP for CS has been little studied. We address this research gap by reproducing and studying the effectiveness of existing QPP methods in the context of CS. While the task of passage retrieval remains the same in the two settings, a user query in CS depends on the conversational history, introducing novel QPP challenges. In particular, we seek to explore to what extent findings from QPP methods for ad-hoc search generalize to three CS settings: (i) estimating the retrieval quality of different query rewriting-based retrieval methods, (ii) estimating the retrieval quality of a conversational dense retrieval method, and (iii) estimating the retrieval quality for top ranks vs. deeper-ranked lists. Our findings can be summarized as follows: (i) supervised QPP methods distinctly outperform unsupervised counterparts only when training data is ample; (ii) point-wise supervised QPP methods outperform their list-wise counterparts in most cases; and (iii) retrieval score-based unsupervised QPP methods show high effectiveness in assessing the conversational dense retrieval method, ConvDR.

CCS CONCEPTS

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

KEYWORDS

Query performance prediction; Ad-hoc search; Conversational search

ACM Reference Format:
Chuan Meng, Negar Arabzadeh, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Query Performance Prediction: From Ad-hoc to Conversational Search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3539618.3591919

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3591919

1 INTRODUCTION

Query performance prediction (QPP) is an essential task in information retrieval (IR). It is about estimating the retrieval quality of a search system for a given query without relevance judgments [14, 16, 22, 26, 59, 62]. QPP has been long studied in the IR community [9]. Numerous benefits of QPP have been identified, including selecting the most effective ranking algorithm for a query [26, 27, 59] based on the difficulty of the input query.

In conversational search (CS) there has been significant progress on multiple subtasks [61], including passage retrieval [13, 58], query rewriting [53, 57], mixed-initiative interactions [3, 60], response generation [38–40], and evaluation [18, 19]. Specifically, passage retrieval has been the main focus of TREC CAsT 2019–2022 [13], where modeling long conversational context for retrieval is shown to be challenging [2]. Moreover, research has shown that mixed-initiative interactions can lead to improved user and system performance [3, 63]. As with ad-hoc retrieval, QPP benefits CS in multiple ways. For instance, effective QPP can help a CS system take appropriate action at the next turn, e.g., take the initiative in asking a clarifying question or saying "I cannot answer your question" to the user, instead of giving a low-quality or risky answer when the estimated retrieval quality for the current user query is low [5, 45].

Despite its importance and significance, little research has been done on QPP for CS [37]. We take the first steps in this direction by conducting a comprehensive reproducibility study, where we examine a variety of QPP methods that were originally designed for ad-hoc retrieval in the setting of CS. We aim to characterize the novel challenges of QPP for CS and highlight the unique characteristics of this field, while simultaneously assessing the effectiveness of existing QPP methods in a conversational setting.

In particular, we highlight three main challenges of QPP applied to CS that distinguish it from the ad-hoc search setting:

(1) a user query in a conversation depends on the conversational context, i.e., it may contain omissions, coreferences, or ambiguities, leading to unforeseen QPP challenges;
(2) QPP for CS has to predict the performance of novel retrieval approaches, approaches that are specifically designed for CS; two main groups of CS methods have been proposed to solve

the query understanding challenge in CS, i.e., query-rewriting-based retrieval [33, 36, 51, 53, 54, 57] and conversational dense retrieval methods [29, 32, 34, 34, 35, 43, 58].

(3) QPP for CS should focus on estimating the retrieval quality for the top-ranked results rather than for a full-ranked list because CS systems need to return brief responses to adapt to limited-bandwidth interfaces, such as a mobile screen [61].

In this reproducibility paper, we design our experiments inspired by these CS characteristics and examine whether established findings on QPP for ad-hoc search still hold under these conditions. Specifically, we study the following findings from the literature on QPP for ad-hoc search: (i) supervised QPP methods outperform unsupervised QPP methods [4, 7, 14, 16, 23, 59]; (ii) list-wise supervised QPP methods outperform their point-wise counterparts [7, 16]; and (iii) retrieval score-based unsupervised QPP methods perform poorly in estimating the retrieval quality of neural-based retrievers [15, 23]. By examining each of these QPP-for-ad-hoc-search findings listed above in the setting of CS, we aim to characterize the problem of QPP applied to CS, with novel findings and directions for future research as additional outcomes.

In this paper, we conduct experiments on three CS datasets: (i) CAsT-19 [13], (ii) CAsT-20 [12], and (iii) OR-QuAC [43]. Our experiments show that, in the setting of CS, (i) supervised QPP methods distinctly outperform unsupervised counterparts only when a large amount of training data is available; unsupervised QPP methods show strong performance in a few-shot setting and when predicting the retrieval quality for deeper ranked lists; (ii) point-wise supervised QPP methods outperform their list-wise counterparts in most cases; however, list-wise QPP methods show a slight advantage in a few-shot setting and when predicting the retrieval quality for deeper ranked lists; and (iii) retrieval score-based unsupervised QPP methods show high effectiveness in estimating the retrieval quality of a conversational dense retrieval method, ConvDR, either for top ranks or deeper ranked lists.

2 PRELIMINARIES AND TASK DEFINITION

We recap the definition of the QPP task in the context of ad-hoc search. Generally, given a query q, a collection of documents D, an ad-hoc retrieval method M and the ranked list with top-k ranked documents $D^k_{q;M} = [d_1, d_2, \ldots, d_k]$ returned by the retriever M over the collection D with respect to the query q, a QPP method f estimates the retrieval quality of the ranked list $D^k_{q;M}$ with respect to the query q, formally:

$$\phi = f(q, D^k_{q;M}, D) \in \mathbb{R}\,, \tag{1}$$

where ϕ indicates the retrieval quality of the ad-hoc retriever M in response to the query q; the retrieval quality ϕ can depend on collection-based statistics.

Next, we define the task of QPP for CS. The CS task is to find relevant items for each query in a multi-turn conversation $Q = \{q_t\}^n_{t=1}$ [13], where n is the number of turns in a conversation. Unlike traditional ad-hoc search, the query q_t at turn t may contain omissions, coreferences, or ambiguities, making it hard for ad-hoc search methods to capture the underlying information need of the query q_t. Two main groups of CS methods have been proposed to solve the query understanding challenge in CS, i.e., query rewriting-based retrieval [33, 36, 51, 53, 57] and conversational dense retrieval methods [32, 34, 58]. Query rewriting-based retrieval methods first rewrite the query q_t into a self-contained query q'_t with the conversational history $Q_{1:t-1} = q_1, q_2, \ldots, q_{t-1}$, and then reuse ad-hoc search methods using the rewritten query q'_t as input. When estimating the retrieval quality of this group of CS methods, we define QPP for CS as:

$$\phi_t = f(q'_t, D^k_{q'_t;M}, D) \in \mathbb{R}\,, \tag{2}$$

where, given the same query rewrite q'_t, the ranked list of documents $D^k_{q'_t;M}$ retrieved by a query rewriting-based retrieval method for the query rewrite q'_t, predicts ϕ_t that is indicative of the retrieval quality of the method in response to the rewritten query q'_t.

Conversational dense retrieval methods train a query encoder to encode the current query q_t and the conversation history $Q_{1:t-1}$ into a contextualized query embedding that is used to represent the information need of the current query in a latent space [34, 58]. However, existing QPP methods do not have such a special module to understand the noisy raw utterances $Q_{1:t}$; directly feeding the raw utterances $Q_{1:t}$ into QPP methods may fuse them. Thus, when estimating the retrieval quality of a conversational dense retrieval method, we still feed a query rewrite q'_t instead of the raw utterances $Q_{1:t}$ into QPP methods, formally:

$$\phi_t = f(q'_t, D^k_{Q_{1:t};M}, D) \in \mathbb{R}\,, \tag{3}$$

where $D^k_{Q_{1:t};M}$ is the ranked list retrieved by a conversational dense retrieval method in response to the raw utterances $Q_{1:t}$.

3 REPRODUCIBILITY METHODOLOGY

We describe our research questions and the experiments designed to address them. We also describe our experimental setup.

3.1 Research questions

We address the following research questions:

(RQ1) Does the performance of QPP methods for ad-hoc search generalize to CS when estimating the retrieval quality of different query rewriting-based retrieval methods?

(RQ2) Does the performance of QPP methods for ad-hoc search generalize to CS when estimating the retrieval quality of a conversational dense retrieval method? Is the QPP effectiveness influenced by the choice of query rewrites?

(RQ3) What is the performance difference between QPP methods when predicting the retrieval quality for top-ranked items vs. for longer-ranked lists?

3.2 Experimental design

Next, we describe the experiments aimed at answering our research questions. Our main goal is to study the reproducibility of ad-hoc QPP methods in the CS setting. We compare the performance of unsupervised and supervised QPP methods on three CS datasets. Specifically, we conduct the following experiments:

E1 To address (**RQ1**), we estimate the retrieval quality of BM25 with three query rewriting methods, namely, T5, QuReTeC, and perfect rewriting (human-rewritten) [13]. Note that QPP methods and BM25 always share the same query rewrites.

E2 To address (**RQ2**), we study the performance of QPP methods for a conversational dense retrieval method, ConvDR [58], on all three datasets. As ConvDR directly models the raw conversation context, no query rewriting step is required. However,

no existing QPP methods can model raw conversations. Hence, we study the effect of feeding different query rewrites into QPP methods when predicting the performance of ConvDR.

E3 To address **(RQ3)**, we apply the QPP methods on evaluation metrics at different depths. We utilize nDCG@3 and nDCG@100 and analyze how QPP performance is affected by the ranking depth. We also consider Recall@100 to study the effectiveness of QPP for first-stage CS rankers, where high recall is desired.

3.3 Experimental setup

QPP methods. We analyze a variety of unsupervised and supervised QPP methods. For unsupervised methods, we consider clarity-based and score-based QPP methods because they have been widely used in the literature. We consider more score-based methods since they have shown great effectiveness [6]. We consider one clarity-based method:

- Clarity [9] quantifies the degree of ambiguity of a query w.r.t. a collection of documents. Specifically, it measures the KL divergence between a relevance model [31] induced from top-ranked documents and a language model induced from the collection:
$$Clarity(q, D^k_{q;M}, D) = \sum_{w \in V} P(w|D^k_{q;M}) \log \frac{P(w|D^k_{q;M})}{P(w|D)}, \quad (4)$$
where w and V denote a term and the entire vocabulary of the collection, respectively. The conjecture is that the larger the KL divergence is, the better the retrieval quality is.

We consider five score-based QPP methods:

- Weighted information gain (WIG) [62] measures the divergence of retrieval scores of top-ranked documents from those of the entire corpus: the higher the divergence is, the better the retrieval quality is [48, 49, 59]. WIG is formulated as:
$$WIG(q, D^k_{q;M}, D) = \frac{1}{k} \sum_{d \in D^k_{q;M}} \frac{1}{\sqrt{|q|}} (Score(q;d) - Score(q;D)), \quad (5)$$
where $Score(q;d)$ and $Score(q;D)$ are the retrieval scores of document d and the entire collection D, respectively; $|q|$ is q's length.
- Normalized query commitment (NQC) [48] measures the standard deviation of retrieval scores of top-ranked documents; the standard deviation is normalized by the retrieval score of the entire collection D. The higher the standard deviation is, the better the retrieval quality is assumed to be. NQC is modeled as:
$$NQC(q, D^k_{q;M}, D) = \frac{1}{Score(q;D)} \sqrt{\frac{1}{k} \sum_{d \in D^k_{q;M}} (Score(q;d) - \mu)^2}, \quad (6)$$
where μ is the mean retrieval score of the top-ranked documents.
- σ_{max} [42] is based on the standard deviation of retrieval scores of ranked documents but finds the most suitable ranked list size k for each query. The intuition is that most of the retrieved documents in a ranked list obtain a low retrieval score; considering such non-relevant documents would hurt QPP effectiveness. σ_{max} computes the standard deviation at each point in the ranked list and selects the maximum standard deviation so as to reduce the impact of the documents with a low retrieval score.
- n($\sigma_{x\%}$) [11] is also based on the standard deviation. Similar to σ_{max}, n($\sigma_{x\%}$) uses a dynamic number of documents to calculate the standard deviation for each query, but only considers the documents whose retrieval scores are at least x% of the top retrieval score. The calculated standard deviation is normalized by query length.
- Score magnitude and variance (SMV) [49] argues that WIG and NQC mainly consider the magnitude and the variance of retrieval scores, respectively. SMV takes both aspects into consideration:
$$SMV(q, D^k_{q;M}, D) = \frac{\frac{1}{k}\sum_{d \in D^k_{q;M}} (Score(q;d)|\ln \frac{Score(q;d)}{\mu}|)}{Score(q;D)}, \quad (7)$$
where $Score(q;d)$ denotes score magnitude while $|\ln \frac{Score(q;d)}{\mu}|$) represents score variance.

Recent studies show that BERT-based supervised QPP methods [4, 7, 16, 23] outperform other neural-based supervised QPP methods, such as NeuralQPP [59] and Deep-QPP [14]. Thus, we consider three BERT-based supervised QPP methods:

- NQA-QPP [23] is the first QPP approach that leverages contextualized embeddings of the query and retrieved documents. NQA-QPP consists of three key components: the retrieval score component, the query component, and the query-document component. All three components are aggregated and fed into a feed-forward neural network for predicting query performance.
- BERT-QPP [4] also leverages contextualized embeddings and achieves a significant performance improvement over earlier work. BERT-QPP fine-tunes a contextualized representation of the queries and the retrieved list of documents, followed by a linear layer for predicting query performance. We use the cross-encoder version of BERT-QPP as it outperforms other variants.
- qppBERT-PL [16] is also an end-to-end neural cross-encoder-based approach, trained list-wise over the top-ranked documents (split into chunks). Specifically, it predicts the number of relevant documents in each chunk of a ranked list.

We do not include BERT-groupwise-QPP [7] in our experiments. It is another list-wise supervised QPP method, which uses cross-query information but it cannot be directly applied in a CS setting, as it reveals future conversation turns, which is unrealistic.

Query rewriting methods. We adopt the following query rewriting techniques/data in the passage retrieval and QPP process: (i) T5 rewriter[1] is fine-tuned on CANARD [17] query rewriting dataset; (ii) QuReTeC [53] is a BERT-based term expansion query rewriting method. We use the checkpoint released by the author;[2] and (iii) Human is the human-generated oracle query rewriting model obtained from the ground-truth data annotations.

CS methods to be evaluated for retrieval quality. We estimate the retrieval quality of two groups of CS methods: query rewriting-based retrieval and conversational dense retrieval methods. For the former, we consider: (i) T5+BM25 rewrites queries using the T5 rewriter and ranks the passages using BM25[3]; (ii) QuReTeC +BM25 [53] performs query resolution using QuReTeC, followed by BM25 passage ranking; and (iii) Human+BM25 uses the ground-truth query rewrites to rank passages using BM25. For the latter, we consider ConvDR [58] and use the code released by the author.[4]

Datasets. We consider three CS datasets: (i) CAsT-19 [13] contains 20 conversations of 9.5 average utterances; (ii) CAsT-20 [12] has 25 conversations of 8.6 average utterances; and (iii) OR-QuAC [43] is

[1] https://huggingface.co/castorini/t5-base-canard
[2] https://github.com/nickvosk/sigir2020-query-resolution [3] We use Pyserini BM25 with the default parameters k1=0.9, b=0.4. [4] https://github.com/thunlp/ConvDR

Table 1: Actual retrieval quality of the CS methods used in this paper in terms of nDCG@3.

	CAsT-19	CAsT-20	OR-QuAC
T5-based query rewriter + BM25	0.330	0.170	0.218
QuReTeC-based query rewriter + BM25	0.338	0.172	0.249
Human query rewriter + BM25	0.360	0.257	0.309
ConvDR	0.471	0.343	0.614

a large-scale synthetic conversational retrieval dataset built on a conversational QA dataset, QuAC [8]; it contains ~5K conversations with ~40K questions. Table 2 lists details of the datasets.

Evaluation. A common method for evaluating QPP performance is to assess the correlation between the actual and predicted performance of a query set. Typically, Pearson's ρ linear coefficient, Kendall's τ, and Spearman ρ ranking correlation are the most commonly used correlation metrics. We report the correlation based on the major metrics adopted by TREC CAsT [13], namely, nDCG@3 for high ranks and nDCG@100 for deeper ranked lists. As mentioned above, we also adopt Recall@100 to investigate the performance of QPP when evaluating first-stage CS retrievers.

Implementation details. We implement all QPP methods using Pytorch.[5] For unsupervised QPP methods, we use hyperparameters that have been shown to be effective by previous studies. Following [62], k is set to 5 for WIG. As suggested by [48, 49], k is set to 100 for NQC and SMV; following [49], we use the average retrieval score of the top-1000 documents as the corpus score $Score(q; D)$. Following [11], we set x to 50 for n($\sigma_{x\%}$). σ_{max} does not use any hyperparameters. Following [48], we use the Clarity variant that uses the sum-normalized retrieval scores (from BM25 or ConvDR in our setting) for weighing documents when constructing a relevance model [31]; our preliminary experiments showed that this variant performed better than the original Clarity that uses query-likelihood scores to weight documents; we induce the relevance model using the top 100 documents and clip the relevance model at the top-100 terms cutoff [47].

For all supervised QPP methods, we use bert-base-uncased,[6] a fixed learning rate (0.00002), and the Adam optimizer [30]. All methods are trained and inferred on an NVIDIA RTX A6000 GPU. Following [34, 58], all training on CAsT-19 or CAsT-20 uses five-fold cross-validation; we use the data split from [58] and train all supervised QPP methods for 5 epochs. For training on OR-QuAC, we train all QPP methods for 1 epoch on the training set of OR-QuAC; we feed QPP methods with human-rewritten queries and train them to estimate the retrieval quality of BM25 with human-rewritten queries. To address the data scarcity on CAsT-19 and CAsT-20, we consider a *warm-up* setting where we first pre-train supervised QPP methods on the training set of OR-QuAC for one epoch, followed by the five-fold cross-validation training for 5 epochs on CAsT. For future reproducibility, our code and data resources are available at https://github.com/ChuanMeng/QPP4CS.

4 RESULTS AND DISCUSSIONS

Our experiments revolve around three main findings from the literature on QPP for ad-hoc search: (i) supervised QPP methods outperform unsupervised QPP methods [4, 7, 14, 16, 23, 59]; (ii) list-wise supervised QPP methods outperform their point-wise counterparts [7, 16]; and (iii) retrieval score-based unsupervised QPP methods perform poorly in estimating the retrieval quality of neural-based retrievers [15, 23]. We study whether the findings listed above continue to hold for QPP methods in CS.

Table 2: Data statistics of CAsT-19, CAsT-20 and OR-QuAC.

	CAsT-19	CAsT-20	OR-QuAC		
	test	test	train	valid	test
#conversations	50	25	4,383	490	771
#conversations (judged)	20	25	–	–	–
#questions	479	216	31,526	3,430	5,571
#questions (judged)	173	208	–	–	–
#documents	38M		11M		

4.1 Assessing query rewriting-based retrieval

4.1.1 Overall performance. To answer (**RQ1**), we examine the results of Experiment **E1**, where we run QPP methods estimating the retrieval quality of BM25 with three query rewriting methods (T5+BM25, QuReTeC+BM25, and Human+BM25). For all supervised QPP methods on CAsT, we further consider their variants that are first pre-trained on the training set of OR-QuAC for one epoch before five-fold cross-validation training on CAsT. See Table 3. Note that QPP methods and BM25 always share the same query rewrites. We have two main observations.

First, when applied to CS, supervised QPP methods only have a distinct advantage over their unsupervised counterparts when training data is sufficient and query rewriting/understanding is relatively easy. Specifically, on OR-QuAC, where training data is ample, all supervised QPP methods perform better than unsupervised methods when assessing BM25 with all three query rewriters. NQA-QPP achieves state-of-art performance on OR-QuAC. On CAsT-19, the performance of unsupervised QPP methods are comparable to supervised QPP methods using five-fold cross-validation. However, on CAsT-20, where query rewriting/understanding is much harder, unsupervised QPP methods perform better than their supervised counterparts using five-fold cross-validation. Warming up on the training set of OR-QuAC brings about improvement in supervised QPP methods in most cases. On CAsT-19, NQA-QPP with warm-up performs better than all unsupervised methods given T5-based and QuReTeC-based query rewrites. Nevertheless, on CAsT-20, even with warm-up, supervised methods do not have a clear advantage. We think it is because all supervised QPP methods need to be fed with queries and the difficulty of query understanding on CAsT-20 limits their performance. Conversely, the prediction of score-based unsupervised QPP methods does not depend on the input queries. The performance of qppBERT-PL drops after warming up on OR-QuAC in most cases. We speculate that this is because qppBERT-PL predicts the number of relevant documents in each chunk of a ranked list, and the number of relevant documents for each query in CAsT is significantly larger than in OR-QuAC, respectively. Therefore, qppBERT-PL's prediction of the relevant document count is biased towards the number of relevant documents in OR-QuAC.

Second, in most cases, point-wise supervised QPP methods such as NQA-QPP and BERTQPP outperform the list-wise supervised

[5] https://pytorch.org/ [6] https://github.com/huggingface/transformers

Table 3: Outcomes of Experiment E1. Performance of QPP methods on three CS datasets: Pearson's r, Kendall's τ, and Spearman's ρ correlation coefficients with nDCG@3, for estimating the retrieval quality of three query rewriting-based retrieval methods (BM25 fed with T5-based, QuReTeC-based, and human-written query rewrites). *Warm-up* means the QPP method is first pre-trained on the training set of OR-QuAC for one epoch. All coefficients are statistically significant (t-test, $p < 0.05$) except the ones in *italics*. The best value in each column is marked in bold, and the second best is <u>underlined</u>.

		T5+BM25			QuReTeC+BM25			Human+BM25		
Datasets	QPP methods	P-ρ	K-τ	S-ρ	P-ρ	K-τ	S-ρ	P-ρ	K-τ	S-ρ
CAsT-19	Clarity	0.321	0.234	0.330	0.327	0.211	0.304	0.359	0.231	0.335
	WIG	0.436	0.232	0.452	0.354	0.250	0.356	0.409	0.293	0.414
	NQC	0.348	0.246	0.354	0.286	0.190	0.275	0.334	0.234	0.335
	σ_{max}	0.442	<u>0.354</u>	0.501	0.351	0.251	0.357	<u>0.410</u>	**0.312**	**0.441**
	n($\sigma_{x\%}$)	0.430	0.332	0.466	0.348	0.259	0.364	0.407	<u>0.307</u>	<u>0.430</u>
	SMV	0.344	0.250	0.360	0.289	0.188	0.273	0.326	0.230	0.333
	NQA-QPP	0.188	*0.047*	*0.072*	*-0.016*	*0.010*	*0.014*	0.152	*0.069*	*0.099*
	BERTQPP	0.440	0.307	0.424	0.352	0.272	0.395	0.270	0.188	0.271
	qppBERT-PL	0.414	0.296	0.421	<u>0.392</u>	<u>0.298</u>	<u>0.406</u>	0.292	0.196	0.280
	NQA-QPP (warm-up)	**0.538**	**0.357**	**0.510**	**0.420**	**0.301**	**0.428**	0.331	0.230	0.336
	BERTQPP (warm-up)	<u>0.526</u>	**0.357**	<u>0.503</u>	0.369	0.264	0.384	**0.418**	0.282	0.411
	qppBERT-PL (warm-up)	0.317	0.218	0.313	0.330	0.232	0.326	0.297	0.190	0.277
CAsT-20	Clarity	<u>0.258</u>	0.191	0.259	*0.099*	*0.061*	*0.085*	*0.127*	*0.089*	*0.121*
	WIG	0.248	**0.251**	**0.339**	0.245	0.163	0.222	<u>0.307</u>	0.222	0.317
	NQC	0.150	<u>0.235</u>	<u>0.316</u>	0.198	0.189	0.259	0.286	**0.266**	**0.370**
	σ_{max}	0.179	0.221	0.304	0.207	0.168	0.230	0.241	0.199	0.283
	n($\sigma_{x\%}$)	0.178	0.225	0.304	0.182	0.133	0.181	0.213	0.167	0.237
	SMV	0.139	0.219	0.298	0.189	0.163	0.227	0.264	<u>0.260</u>	<u>0.363</u>
	NQA-QPP	*0.001*	*0.067*	*0.093*	*-0.064*	*-0.082*	*-0.111*	*0.086*	*-0.011*	*-0.012*
	BERTQPP	*0.042*	*-0.009*	*-0.007*	0.172	0.145	0.196	0.194	0.110	0.159
	qppBERT-PL	*0.131*	0.125	0.159	0.175	0.150	0.185	*0.043*	*0.015*	*0.021*
	NQA-QPP (warm-up)	**0.274**	0.170	0.227	0.190	0.149	0.201	0.231	0.155	0.222
	BERTQPP (warm-up)	0.207	0.171	0.236	**0.403**	**0.301**	**0.409**	**0.336**	0.227	0.318
	qppBERT-PL (warm-up)	0.228	0.213	0.275	<u>0.317</u>	<u>0.268</u>	<u>0.335</u>	*0.094*	*0.095*	*0.130*
OR-QuAC	Clarity	0.090	0.085	0.110	0.110	0.103	0.133	0.076	0.069	0.091
	WIG	0.247	0.235	0.304	0.290	0.270	0.350	0.257	0.241	0.316
	NQC	0.251	0.274	0.355	0.290	0.311	0.404	0.276	0.291	0.381
	σ_{max}	0.317	0.279	0.359	0.367	0.316	0.406	0.412	0.367	0.474
	n($\sigma_{x\%}$)	0.181	0.172	0.223	0.229	0.209	0.270	0.245	0.193	0.252
	SMV	0.204	0.239	0.310	0.239	0.273	0.355	0.194	0.232	0.304
	NQA-QPP	**0.781**	**0.566**	**0.695**	**0.792**	**0.591**	**0.725**	**0.809**	**0.621**	**0.767**
	BERTQPP	<u>0.678</u>	0.434	0.546	<u>0.692</u>	0.476	<u>0.598</u>	<u>0.725</u>	<u>0.527</u>	<u>0.666</u>
	qppBERT-PL	0.594	<u>0.507</u>	<u>0.576</u>	0.617	<u>0.526</u>	0.597	0.618	0.525	0.600

method qppBERT-PL. With five-fold cross-validation, qppBERT-PL has a slight advantage over its point-wise counterparts. E.g., qppBERT-PL achieves a better performance in predicting the performance of QuReTeC+BM25, Human+BM25 on CAsT-19, and T5+BM25, QuReTeC+BM25 on CAsT-20. qppBERT-PL's list-wise training scheme learns from interactions between a query and all documents in a ranked list, providing the model with more training signals and better use of limited training data.

4.1.2 Turn-wise QPP effectiveness. We study the QPP effectiveness on each turn of conversation on CAsT-19; we report the turn-wise effectiveness of 2 unsupervised (WIG, NQC) and 2 supervised QPP methods (NQA-QPP with warm-up, BERT-QPP with warm-up, qppBERT-PL) when they assess BM25 with T5-based and human-written query rewrites. The results are presented in the two leftmost subfigures in Figure 1. Note that we also introduce the turn-wise actual retrieval quality in terms of nDCG@3 in Figure 1. As indicated in both subfigures, there is a correlation between actual retrieval quality and QPP effectiveness: BERT-QPP effectiveness drops as the actual retrieval quality drops; in contrast, the score-based method WIG is not that sensitive to the actual retrieval quality.

4.2 Assessing conversational dense retrieval

4.2.1 Overall performance. To answer (**RQ2**), we examine the results of **E2**. We apply QPP methods (fed with different types of query rewrites) to estimate the retrieval quality of the conversational dense retrieval method ConvDR. See Table 4. Note that the results of NQC, σ_{max} and SMV are invariant to different types of

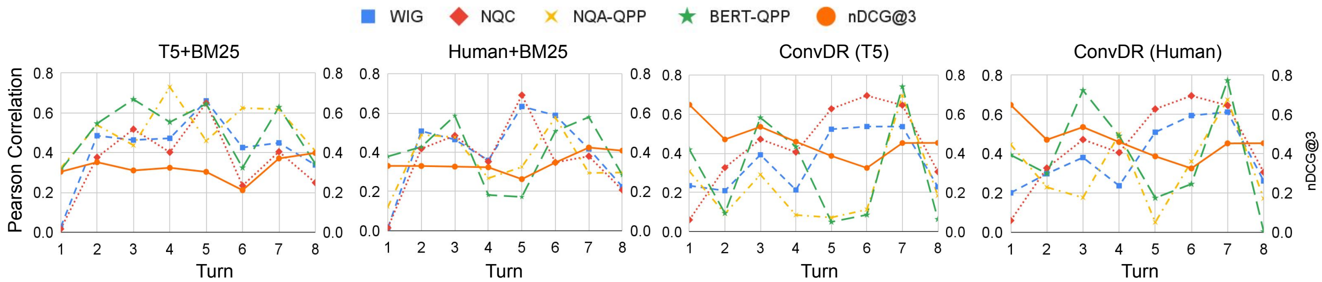

Figure 1: QPP effectiveness on each turn of conversations in CAsT-19. Pearson's r correlation between the actual nDCG@3 scores of the queries with the same turn number and their estimated retrieval quality is calculated per turn.

query rewrites because they only depend on retrieval scores; Clarity is also invariant to query rewrites because we use the Clarity variant [48] that uses retrieval scores (from ConvDR in our setting) to weight documents when constructing a relevance model [31]; see Section 3.3 for more details. We have three main observations.

First, retrieval score-based QPP methods NQC and WIG show high effectiveness in estimating the retrieval quality of the conversational dense retrieval method ConvDR, achieving the best performance in most cases on CAsT-19 and CAsT-20. Compared to Table 3, the performance of NQC and WIG is even better than their effectiveness in assessing BM25. It contradicts the previous findings [15, 23]: Datta et al. [15] found that the retrieval scores from neural-based retrievers, such as ColBERT [28], are restricted within a shorter range compared to lexical-based retrievers, which may limit the performance of score-based unsupervised QPP methods. We speculate that there are two reasons. First, the effectiveness of score-based QPP methods depends on the retrieval score distribution of a specific retriever. Figure 2 illustrates the retrieval score distributions of ConvDR and BM25 with three rewrites on the three datasets given all queries in the three datasets. The retrieval score distribution of ConvDR displays a higher variance. A higher standard deviation indicates that the score ranges vary more, and so the top-ranked documents are more distinguishable from the rest. Thus, ConvDR has a higher potential to be predicted more accurately using score-based QPP methods. Second, as discussed in Section 4.1.1, score-based QPP methods do not depend on the input queries and are not impacted by the query understanding challenge in CS. Thus, score-based QPP methods show more effectiveness when assessing ConvDR compared to other supervised methods.

Second, similar to our results for **(RQ1)**, supervised QPP methods distinctly outperform all unsupervised QPP methods on the OR-QuAC dataset where a large amount of training data is available. NQA-QPP remains the state-of-the-art method on OR-QuAC.

Third, as with the results for **(RQ1)**, point-wise supervised QPP methods outperform the list-wise supervised method qppBERT-PL in most cases (on CAsT-20 and OR-QuAC). However, on CAsT-19, qppBERT-PL trained using five-fold cross-validation outperforms its point-wise counterparts warming up from OR-QuAC, showing its potential in a few-shot setting.

4.2.2 Turn-wise QPP effectiveness. Similar to Section 4.1.2, here we report the turn-wise effectiveness of the same 4 QPP methods when they are fed with T5-based and human-written query rewrites to assess ConvDR. See the two rightmost subfigures in Figure 1. As we can see, the actual retrieval quality drops at turn 6 obviously; the effectiveness of the two supervised QPP methods drops at turn 6 sharply. Conversely, the effectiveness of WIG and NQC increases at turn 6. It shows that score-based QPP methods are less sensitive to the actual retrieval quality when assessing the conversational dense retrieval method, ConvDR.

4.3 Top ranks vs. deeper ranked lists

To answer **(RQ3)**, we report the results of **E3** in Table 5, i.e., QPP results in terms of nDCG@3, nDCG@100, and Recall@100. We have three main observations.

First, all QPP methods generally perform better when predicting the retrieval quality for deeper-ranked lists. The estimated performance by various QPP methods achieves a higher correlation with the actual nDCG@100/Recall@100 values in comparison with the nDCG@3 values, which is in line with [59], that found predicting NDCG@20 to be harder than AP@1000.

Second, unsupervised QPP methods get a higher correlation with nDCG@100 and Recall@100 on CAsT-19 and CAsT-20, showing high effectiveness in estimating the retrieval quality of deeper ranked lists. As seen in previous experiments, supervised QPP methods still keep the lead on OR-QuAC.

Third, in some cases, list-wise supervised QPP methods outperform than their point-wise counterparts when estimating the retrieval quality in terms of deeper ranked lists. E.g., qppBERT-PL without warm-up outperforms other point-wise methods (NQA-QPP and BERTQPP with warm-up) on CAsT-19 when predicting the performance of ConvDR in terms of nDCG@100 and Recall@100. Also, qppBERT-PL achieves the best performance when predicting the performance of ConvDR in terms of Recall@100 on OR-QuAC. The gains indicate that modeling a list of retrieved items has the potential of benefiting the retrieval quality estimation for deeper-ranked lists.

5 RELATED WORK

Query performance prediction. The QPP task is to estimate the retrieval quality of a search system in response to a user query without relevance judgments [6, 26]. QPP methods have shown a high correlation with the retrieval quality in the context of ad-hoc retrieval. They can help to obtain better-performing retrieval pipelines in different ways, including query routing [46]. For example, identifying poor-performing queries in practice has shown to be helpful with intelligent assistants [5]. QPP methods can be used to identify user interactions with an intelligent assistant for which the system may not have a reasonable answer [45]. In such cases, the

Table 4: Outcomes of Experiment E2. Performance of QPP methods on three CS datasets: Pearson's r, Kendall's τ, and Spearman's ρ correlation coefficients with nDCG@3, for estimating the retrieval quality of ConvDR (fed with T5-based, QuReTeC-based, and human-written query rewrites). All coefficients are statistically significant (t-test, $p < 0.05$) except the ones in *italics*. The best value in each column is marked in bold, and the second best is <u>underlined</u>.

		T5			QuReTeC			Human		
Datasets	QPP methods	P-ρ	K-τ	S-ρ	P-ρ	K-τ	S-ρ	P-ρ	K-τ	S-ρ
CAsT-19	Clarity	0.257	0.176	0.257	0.257	0.176	0.257	0.257	0.176	0.257
	WIG	<u>0.387</u>	0.274	0.395	<u>0.388</u>	0.266	0.381	<u>0.412</u>	<u>0.285</u>	<u>0.408</u>
	NQC	**0.431**	**0.307**	**0.438**	**0.431**	**0.307**	**0.438**	**0.431**	**0.307**	**0.438**
	σ_{max}	0.378	0.267	0.381	0.378	0.267	0.381	0.378	0.267	0.381
	n($\sigma_{x\%}$)	0.187	0.175	0.262	0.181	0.170	0.256	0.216	0.196	0.288
	SMV	0.386	<u>0.285</u>	<u>0.405</u>	0.386	<u>0.285</u>	<u>0.405</u>	0.386	<u>0.285</u>	0.405
	NQA-QPP	*0.121*	*0.075*	*0.115*	*0.118*	*0.073*	*0.109*	0.150	0.109	0.153
	BERTQPP	0.167	0.107	0.169	0.220	0.145	0.217	0.298	0.193	0.296
	qppBERT-PL	0.344	0.225	0.324	0.316	0.197	0.284	0.276	0.178	0.255
	NQA-QPP (warm-up)	0.187	0.128	0.186	0.161	0.107	0.157	0.287	0.191	0.282
	BERTQPP (warm-up)	0.282	0.187	0.277	0.234	0.157	0.233	0.371	0.251	0.361
	qppBERT-PL (warm-up)	0.212	0.151	0.213	0.167	0.117	0.170	0.172	0.115	0.154
CAsT-20	Clarity	*0.126*	*0.088*	*0.127*	*0.126*	*0.088*	*0.127*	*0.126*	*0.088*	*0.127*
	WIG	**0.377**	**0.277**	**0.386**	**0.377**	**0.263**	**0.373**	<u>0.384</u>	0.264	0.368
	NQC	<u>0.339</u>	<u>0.261</u>	<u>0.360</u>	<u>0.339</u>	<u>0.261</u>	<u>0.360</u>	0.339	0.261	0.360
	σ_{max}	0.282	0.219	0.310	0.282	0.219	0.310	0.282	0.219	0.310
	n($\sigma_{x\%}$)	0.199	0.168	0.236	0.197	0.156	0.224	0.201	0.156	0.220
	SMV	0.275	0.216	0.299	0.275	0.216	0.299	0.275	0.216	0.299
	NQA-QPP	*-0.037*	*-0.037*	*-0.058*	*-0.081*	*-0.063*	*-0.092*	*0.059*	*0.023*	*0.032*
	BERTQPP	0.223	0.157	0.226	0.216	0.146	0.212	**0.404**	**0.281**	**0.395**
	qppBERT-PL	0.185	0.144	0.191	*0.029*	*0.023*	*0.031*	0.251	0.171	0.232
	NQA-QPP (warm-up)	0.315	0.218	0.313	0.240	0.178	0.245	0.374	0.267	0.375
	BERTQPP (warm-up)	0.253	0.183	0.257	0.320	0.236	0.338	0.349	0.244	0.346
	qppBERT-PL (warm-up)	0.218	0.164	0.227	0.140	0.115	0.157	0.348	<u>0.268</u>	<u>0.376</u>
OR-QuAC	Clarity	-0.050	-0.029	-0.038	-0.050	-0.029	-0.038	-0.050	-0.029	-0.038
	WIG	0.137	0.107	0.145	0.116	0.088	0.120	0.140	0.111	0.149
	NQC	0.227	0.163	0.221	0.227	0.163	0.221	0.227	0.163	0.221
	σ_{max}	0.442	0.339	0.443	0.442	0.339	0.443	0.442	0.339	0.443
	n($\sigma_{x\%}$)	-0.032	*-0.003*	*-0.004*	-0.073	-0.035	-0.045	*-0.022*	*0.008*	*0.011*
	SMV	0.098	0.076	0.106	0.098	0.076	0.106	0.098	0.076	0.106
	NQA-QPP	**0.615**	**0.479**	**0.615**	**0.639**	**0.499**	**0.638**	**0.600**	**0.470**	**0.601**
	BERTQPP	<u>0.481</u>	<u>0.417</u>	<u>0.541</u>	<u>0.505</u>	<u>0.435</u>	<u>0.563</u>	<u>0.481</u>	<u>0.408</u>	<u>0.529</u>
	qppBERT-PL	0.391	0.250	0.287	0.424	0.294	0.335	0.437	0.306	0.349

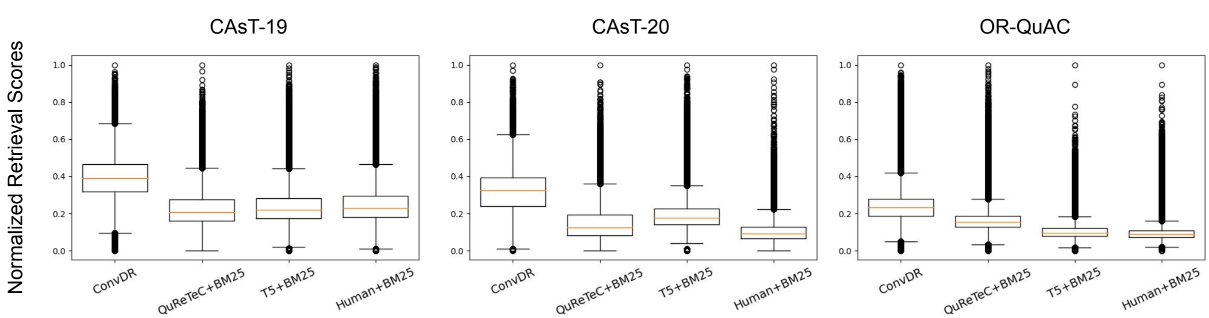

Figure 2: Distributions of retrieval scores for ConvDR and BM25 with three different rewriters on the three datasets. For the sake of comparison, we normalize the retrieval scores of a retriever for all queries in a dataset by min-max normalization.

assistant may respond with "I do not know the answer to your question" or ask the user to reformulate their question in order to avoid being unhelpful to the user [5]. Moreover, query difficulty signals have been used to provide direct feedback to users, allowing them

Table 5: Outcomes of Experiment E3. Performance of QPP methods on three CS datasets: Pearson's r, Kendall's τ, and Spearman's ρ correlation coefficients with nDCG@3, nDCG@100 and Recall@100, for estimating the retrieval quality of BM25 fed with T5-based query rewrites and ConvDR. All coefficients are statistically significant (t-test, $p < 0.05$) except the ones in *italics*. The best value in each column is marked in bold, and the second best is <u>underlined</u>.

		T5 + BM25						ConvDR (QPP fed with T5 query rewrites)					
		nDCG@3		nDCG@100		Recall@100		nDCG@3		nDCG@100		Recall@100	
	QPP methods	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
CAsT-19	Clarity	0.321	0.234	0.326	0.257	0.214	0.187	0.257	0.176	0.342	0.227	0.335	0.216
	WIG	0.436	0.232	**0.608**	<u>0.429</u>	**0.579**	0.426	<u>0.387</u>	0.274	0.542	0.398	0.379	0.301
	NQC	0.348	0.246	0.548	0.397	<u>0.545</u>	<u>0.444</u>	**0.431**	**0.307**	**0.647**	**0.481**	<u>0.557</u>	0.421
	σ_{max}	0.442	<u>0.354</u>	<u>0.574</u>	**0.433**	0.494	0.399	0.378	0.267	<u>0.637</u>	0.456	**0.591**	**0.441**
	n($\sigma_{x\%}$)	0.430	0.332	0.569	0.406	0.505	0.365	0.187	0.175	0.358	0.292	0.362	0.288
	SMV	0.344	0.250	0.548	0.417	0.541	**0.466**	0.386	<u>0.285</u>	0.619	<u>0.471</u>	0.556	<u>0.423</u>
	NQA-QPP (warm-up)	**0.538**	**0.357**	0.542	0.392	0.537	0.377	0.187	0.128	0.401	0.275	0.364	0.263
	BERTQPP (warm-up)	<u>0.526</u>	**0.357**	0.532	0.391	0.463	0.325	0.282	0.187	0.378	0.249	0.261	0.194
	qppBERT-PL (warm-up)	0.317	0.218	0.412	0.279	0.363	0.263	0.212	0.151	0.354	0.233	0.345	0.249
	qppBERT-PL	0.414	0.296	0.509	0.358	0.452	0.312	0.344	0.225	0.461	0.310	0.455	0.327
CAsT-20	Clarity	<u>0.258</u>	0.191	0.452	0.343	<u>0.467</u>	0.332	*0.126*	*0.088*	0.270	0.195	0.264	0.178
	WIG	0.248	**0.251**	**0.494**	**0.453**	**0.478**	**0.438**	**0.377**	**0.277**	**0.549**	<u>0.389</u>	**0.465**	0.320
	NQC	0.150	<u>0.235</u>	0.363	0.399	0.320	0.380	<u>0.339</u>	<u>0.261</u>	<u>0.544</u>	**0.404**	<u>0.463</u>	**0.357**
	σ_{max}	0.179	0.221	0.339	0.372	0.339	0.382	0.282	0.219	0.496	0.364	0.440	0.328
	n($\sigma_{x\%}$)	0.178	0.225	0.413	<u>0.422</u>	0.420	<u>0.410</u>	0.199	0.168	0.409	0.309	0.397	0.285
	SMV	0.139	0.219	0.362	0.400	0.333	0.387	0.275	0.216	0.503	0.380	0.454	<u>0.352</u>
	NQA-QPP (warm-up)	**0.274**	0.170	<u>0.471</u>	0.362	0.466	0.370	0.315	0.218	0.310	0.237	0.324	0.223
	BERTQPP (warm-up)	0.207	0.171	0.404	0.301	0.364	0.246	0.253	0.183	0.349	0.242	0.221	0.133
	qppBERT-PL (warm-up)	0.228	0.213	0.367	0.305	0.312	0.287	0.218	0.164	0.378	0.272	0.313	0.229
	qppBERT-PL	*0.131*	0.125	0.310	0.251	0.363	0.275	0.185	0.144	0.301	0.217	0.263	0.196
OR-QuAC	Clarity	0.090	0.085	0.197	0.196	0.362	0.312	-0.050	-0.029	-0.029	-0.015	0.053	0.057
	WIG	0.247	0.235	0.376	0.370	0.482	0.450	0.137	0.107	0.195	0.130	0.298	0.261
	NQC	0.251	0.274	0.356	0.409	0.414	0.461	0.227	0.163	0.302	0.194	0.402	0.333
	σ_{max}	0.317	0.279	0.418	0.393	0.438	0.437	0.442	0.339	0.490	0.359	0.434	<u>0.370</u>
	n($\sigma_{x\%}$)	0.181	0.172	0.295	0.302	0.415	0.401	-0.032	*-0.003*	*-0.001*	*0.010*	0.102	0.106
	SMV	0.204	0.239	0.311	0.383	0.396	0.456	0.098	0.076	0.170	0.109	0.313	0.277
	NQA-QPP	**0.781**	**0.566**	**0.783**	**0.602**	**0.603**	**0.486**	**0.615**	**0.479**	**0.644**	**0.475**	0.446	0.323
	BERTQPP	<u>0.678</u>	<u>0.434</u>	<u>0.767</u>	0.551	<u>0.589</u>	<u>0.484</u>	<u>0.481</u>	<u>0.417</u>	<u>0.595</u>	<u>0.453</u>	<u>0.447</u>	0.313
	qppBERT-PL	0.594	0.507	0.655	<u>0.552</u>	0.451	0.440	0.391	0.250	0.449	0.277	**0.455**	**0.383**

to reformulate queries or seek alternative information sources if the results are expected to be poor. This can be achieved by presenting visual feedback on the predicted performance of a query, in addition to top results. Query refinement [56] and personalization [50] can also benefit from performance prediction methods; QPP methods can be used to estimate the expected utility of different refinement terms and identifying queries that can benefit from personalization, thereby improving users' search experiences [3, 6, 60].

QPP methods can be classified into pre- and post-retrieval methods. Pre-retrieval methods estimate query performance based on the query and corpus statistics before retrieval takes place. Post-retrieval methods use additional information from the ranked list to predict query performance after retrieval. Consequently, post-retrieval QPP methods have shown superior performance compared to the pre-retrieval metrics when predicting retrieval performance [6]. We focus on the performance of post-retrieval QPP methods. To the best of our knowledge, all the current pre-retrieval QPP methods adopt an unsupervised approach. Post-retrieval QPP methods include both supervised and unsupervised methods.

Traditional query performance prediction methods have mostly relied on an unsupervised approach where query term frequency and corpus statistics are used as indicators for query performance [24–27, 47, 48, 62]. These statistics include measures such as the similarity between the query and the retrieved documents [48], the divergence between the retrieved documents and the corpus [10], and the distribution of the relevance scores obtained for the retrieved documents [9, 62]. More recent approaches use deep learning-based models to train supervised QPP methods. Supervised methods for query performance prediction are more effective than unsupervised approaches in an ad-hoc retrieval setting. However, these supervised methods require a significant amount of data and training instances, such as the MS MARCO dataset [41], in order to perform QPP effectively [4, 16, 23, 59]. Hashemi et al. [23] escaped this

limitation by exploring the ability of QPP methods to predict performance when addressing non-factoid question-answering tasks. Studies of the performance of QPP methods in other settings, such as CS, have been limited.

In this reproducibility paper, we contribute an analysis of the effectiveness of QPP methods in predicting the performance of retrieval methods in the context of CS. We do so by providing the results of a comprehensive benchmark of state-of-the-art QPP methods and highlighting the drawbacks and strengths of QPP methods on three different conversational search datasets.

Conversational search. Conversational search (CS) is the task of retrieving relevant passages in response to user queries in a multi-turn conversation [13]. A unique challenge in CS is that a user query in a conversation is context-dependent, i.e., it may contain omissions, coreferences, or ambiguities, making it challenging for ad-hoc search methods to capture the underlying information need [44]. Recovering the underlying information need from the conversational history is crucial [34]. To address this challenge, there are two main groups of CS methods, namely, *query-rewriting-based retrieval* and *conversational dense retrieval*. Query-rewriting-based retrieval methods first rewrite a query that is part of a conversation into a self-contained query and then feed it to an ad-hoc retriever [33, 36, 51, 53, 54, 57]. Query rewriting can be conducted by either term expansion or sequence generation. The former adds terms from the conversational history to the current query, e.g., by designing rules [36] or training a binary term classifier [51], while the latter directly generates the reformulated queries using pre-trained generative language models, e.g., GPT-2 [57] and T5 [33].

Conversational dense retrieval methods train a query encoder to encode the current query and the conversational history into a contextualized query embedding; the contextualized query embedding is expected to implicitly represent the information need of the current query in a latent space [29, 32, 34, 35, 43, 58]. Lin et al. [32] train the query encoder by optimizing a ranking loss over a large number of pseudo-relevance judgments. Yu et al. [58] train the query encoder to mimic the embeddings of human-written queries output by the query encoder of the ad-hoc dense retriever ANCE [55]. Mao et al. [34] train the query encoder to denoise noisy turns in the conversation history by contrastive learning.

Little research has been done into QPP for CS. Arabzadeh et al. [5], Roitman et al. [45] explore QPP in single-turn CS, where they use QPP to help a CS system take the next appropriate action given a user query. Specifically, they use QPP to assess the retrieved answer quality to determine whether the system should return the answer to the user. Al-Thani et al. [1], Lin et al. [33] use QPP to improve the retrieval quality of a CS system. Lin et al. [33] use a QPP method to determine whether the current query should be expanded with keywords from the previous turns. Al-Thani et al. [1] use QPP methods to select the better query rewrite from different ones. Meng et al. [37] investigate the performance of pre-retrieval QPP methods when they estimate the retrieval quality of BM25 fed with T5-generated query rewrites. Also, Meng et al. [37] propose to incorporate query rewriting quality to improve QPP effectiveness. Additionally, Vlachou and Macdonald [52] explore QPP in the context of conversational fashion recommendation, which differs from CS.

What we add to the studies listed above, is a comprehensive reproducibility study where we reproduce various QPP methods designed for ad-hoc search systems in the setting of multi-turn CS.

6 CONCLUSION

In this reproducibility study, we examined whether four key findings for QPP in ad-hoc search hold in CS. We experimented with QPP methods designed for ad-hoc search in three CS settings: (i) predicting the retrieval quality of BM25 while studying the impact of query rewriting; (ii) predicting the retrieval quality of a conversational dense retrieval method, namely ConvDR; and (iii) predicting the retrieval quality for top ranks vs. deeper-ranked lists.

We found that the three findings on QPP for ad-hoc search do not generalize to CS very well. Specifically, we found (i) supervised QPP methods distinctly outperform their unsupervised counterparts only when a large amount of training data is available, while unsupervised QPP methods show strong performance when being in a few-shot setting and predicting the retrieval quality for deeper ranked lists; (ii) point-wise supervised QPP methods outperform their list-wise counterparts in most cases; however, list-wise QPP methods are more data-efficient, show a slight advantage in predicting the retrieval quality for deeper ranked lists; and (iii) retrieval score-based unsupervised QPP methods show high effectiveness in estimating the retrieval quality of a conversational dense retrieval method, ConvDR, either for top ranks or deeper ranked lists.

Our paper reveals the drawbacks of QPP methods designed for ad-hoc search in the context of CS, motivating the next direction for the modeling of QPP for CS. E.g., we show that the data sparsity problem in CS severely reduces the performance of supervised QPP methods. Thus, designing QPP methods using few-shot learning techniques, e.g., prompt learning, to solve the data sparsity problem in CS is one possible way. Also, we show that the quality of query rewriting and conversation context modeling is of great importance.

We point to two limitations of our study, namely, (i) we only consider estimating the retrieval quality of one conversational dense retrieval method, and (ii) we only use correlation metrics to evaluate the performance of QPP methods. In future work, we plan to (i) consider more conversational dense retrieval methods such as CQE [32] as well as other dense retrieval methods for CS, such as T5-based rewriter+ANCE [55], and (ii) introduce QPP-specific evaluation metrics, such as scaled Absolute Ranking Error (sARE) and scaled Mean Absolute Ranking Error (sMARE) [20, 21].

ACKNOWLEDGMENTS

We would like to thank our reviewers for their feedback. This research was partially supported by the China Scholarship Council (CSC) under grant number 202106220041, and the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl.

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Haya Al-Thani, Tamer Elsayed, and Bernard J Jansen. 2022. Improving Conversational Search with Query Reformulation Using Selective Contextual History. *Data and Information Management* (2022), 100025.

[2] Mohammad Aliannejadi, Manajit Chakraborty, Esteban Andrés Ríssola, and Fabio Crestani. 2020. Harnessing Evolution of Multi-Turn Conversations for Effective Answer Retrieval. In *CHIIR*. 33–42.

[3] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *SIGIR*. 475–484.

[4] Negar Arabzadeh, Maryam Khodabakhsh, and Ebrahim Bagheri. 2021. BERT-QPP: Contextualized Pre-trained Transformers for Query Performance Prediction. In *CIKM*. 2857–2861.

[5] Negar Arabzadeh, Mahsa Seifikar, and Charles LA Clarke. 2022. Unsupervised Question Clarity Prediction Through Retrieved Item Coherency. In *CIKM*. 3811–3816.

[6] 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.

[7] Xiaoyang Chen, Ben He, and Le Sun. 2022. Groupwise Query Performance Prediction with Bert. In *ECIR*. Springer, 64–74.

[8] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *EMNLP*. Association for Computational Linguistics, 2174–2184.

[9] Steve Cronen-Townsend, Yun Zhou, and W Bruce Croft. 2002. Predicting Query Performance. In *SIGIR*. 299–306.

[10] Steve Cronen-Townsend, Yun Zhou, and W Bruce Croft. 2006. Precision Prediction Based on Ranked List Coherence. *Information Retrieval* 9, 6 (2006), 723–755.

[11] Ronan Cummins, Joemon Jose, and Colm O'Riordan. 2011. Improved Query Performance Prediction Using Standard Deviation. In *SIGIR*. 1089–1090.

[12] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. CAsT 2020: The Conversational Assistance Track Overview. In *Text Retrieval Conference*.

[13] Jeffrey Dalton, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan. 2020. Cast-19: A Dataset for Conversational Information Seeking. In *SIGIR*. 1985–1988.

[14] Suchana Datta, Debasis Ganguly, Derek Greene, and Mandar Mitra. 2022. Deep-QPP: A Pairwise Interaction-based Deep Learning Model for Supervised Query Performance Prediction. In *WSDM*. 201–209.

[15] Suchana Datta, Debasis Ganguly, Mandar Mitra, and Derek Greene. 2022. A Relative Information Gain-based Query Performance Prediction Framework with Generated Query Variants. *TOIS* (2022).

[16] Suchana Datta, Sean MacAvaney, Debasis Ganguly, and Derek Greene. 2022. A 'Pointwise-Query, Listwise-Document' based Query Performance Prediction Approach. In *SIGIR*. 2148–2153.

[17] Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can You Unpack That? Learning to Rewrite Questions-in-Context. In *EMNLP*. 5918–5924.

[18] Guglielmo Faggioli, Marco Ferrante, Nicola Ferro, Raffaele Perego, and Nicola Tonellotto. 2021. Hierarchical Dependence-aware Evaluation Measures for Conversational Search. In *SIGIR*. 1935–1939.

[19] Guglielmo Faggioli, Marco Ferrante, Nicola Ferro, Raffaele Perego, and Nicola Tonellotto. 2022. A Dependency-Aware Utterances Permutation Strategy to Improve Conversational Evaluation. In *ECIR*. Springer, 184–198.

[20] Guglielmo Faggioli, Oleg Zendel, J Shane Culpepper, Nicola Ferro, and Falk Scholer. 2021. An Enhanced Evaluation Framework for Query Performance Prediction. In *ECIR*. Springer, 115–129.

[21] Guglielmo Faggioli, Oleg Zendel, J Shane Culpepper, Nicola Ferro, and Falk Scholer. 2022. sMARE: A New Paradigm to Evaluate and Understand Query Performance Prediction Methods. *Information Retrieval Journal* 25, 2 (2022), 94–122.

[22] Debasis Ganguly, Suchana Datta, Mandar Mitra, and Derek Greene. 2022. An Analysis of Variations in the Effectiveness of Query Performance Prediction. In *ECIR*. Springer, 215–229.

[23] Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2019. Performance Prediction for Non-factoid Question Answering. In *ICTIR*. 55–58.

[24] Claudia Hauff, Leif Azzopardi, Djoerd Hiemstra, and Franciska de Jong. 2010. Query Performance Prediction: Evaluation Contrasted with Effectiveness. In *ECIR*. Springer, 204–216.

[25] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. 2008. A Survey of Pre-retrieval Query Performance Predictors. In *CIKM*. 1419–1420.

[26] Ben He and Iadh Ounis. 2006. Query Performance Prediction. *Information Systems* 31, 7 (2006), 585–594.

[27] Jiyin He, Martha Larson, and Maarten de Rijke. 2008. Using Coherence-based Measures to Predict Query Difficulty. In *ECIR*. Springer, 689–694.

[28] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *SIGIR*. 39–48.

[29] Sungdong Kim and Gangwoo Kim. 2022. Saving Dense Retriever from Shortcut Dependency in Conversational Search. In *EMNLP*. 10278–10287.

[30] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[31] Victor Lavrenko and W Bruce Croft. 2001. Relevance-Based Language Models. In *SIGIR*. 120–127.

[32] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. Contextualized Query Embeddings for Conversational Search. In *EMNLP*. 1004–1015.

[33] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2021. Multi-stage Conversational Passage Retrieval: An Approach to Fusing Term Importance Estimation and Neural Query Rewriting. *TOIS* 39, 4 (2021), 1–29.

[34] Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022. Curriculum Contrastive Context Denoising for Few-shot Conversational Dense Retrieval. In *SIGIR*. 176–186.

[35] Kelong Mao, Zhicheng Dou, Hongjin Qian, Fengran Mo, Xiaohua Cheng, and Zhao Cao. 2022. ConvTrans: Transforming Web Search Sessions for Conversational Dense Retrieval. In *EMNLP*. 2935–2946.

[36] Ida Mele, Cristina Ioana Muntean, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, and Ophir Frieder. 2020. Topic Propagation in Conversational Search. In *SIGIR*. 2057–2060.

[37] Chuan Meng, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Performance Prediction for Conversational Search Using Perplexities of Query Rewrites. In *QPP++2023*. 25–28.

[38] Chuan Meng, Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. RefNet: A Reference-aware Network for Background Based Conversation. In *AAAI*.

[39] Chuan Meng, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tengxiao Xi, and Maarten de Rijke. 2021. Initiative-Aware Self-Supervised Learning for Knowledge-Grounded Conversations. In *SIGIR*. 522–532.

[40] Chuan Meng, Pengjie Ren, Zhumin Chen, Weiwei Sun, Zhaochun Ren, Zhaopeng Tu, and Maarten de Rijke. 2020. DukeNet: A Dual Knowledge Interaction Network for Knowledge-Grounded Conversation. In *SIGIR*. 1151–1160.

[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] Joaquín Pérez-Iglesias and Lourdes Araujo. 2010. Standard Deviation as a Query Hardness Estimator. In *SPIRE*. Springer, 207–212.

[43] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval Conversational Question Answering. In *SIGIR*. 539–548.

[44] Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *CHIIR*. 117–126.

[45] Haggai Roitman, Shai Erera, and Guy Feigenblat. 2019. A Study of Query Performance Prediction for Answer Quality Determination. In *ICTIR*. 43–46.

[46] Surendra Sarnikar, Zhu Zhang, and J Leon Zhao. 2014. Query-performance Prediction for Effective Query Routing in Domain-specific Repositories. *Journal of the Association for Information Science and Technology* 65, 8 (2014), 1597–1614.

[47] Anna Shtok, Oren Kurland, and David Carmel. 2010. Using Statistical Decision Theory and Relevance Models for Query-performance Prediction. In *SIGIR*. 259–266.

[48] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting Query Performance by Query-Drift Estimation. *TOIS* 30, 2 (2012), 1–35.

[49] Yongquan Tao and Shengli Wu. 2014. Query Performance Prediction by Considering Score Magnitude and Variance Together. In *CIKM*. 1891–1894.

[50] Jaime Teevan, Susan T Dumais, and Daniel J Liebling. 2008. To Personalize or Not to Personalize: Modeling Queries with Variation in User Intent. In *SIGIR*. 163–170.

[51] Svitlana Vakulenko, Nikos Voskarides, Zhucheng Tu, and Shayne Longpre. 2021. A Comparison of Question Rewriting Methods for Conversational Passage Retrieval. In *ECIR*. 418–424.

[52] Maria Vlachou and Craig Macdonald. 2022. Performance Predictors for Conversational Fashion Recommendation. In *KaRS*.

[53] Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query Resolution for Conversational Search with Limited Supervision. In *SIGIR*. 921–930.

[54] Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, and Gaurav Singh Tomar. 2022. CONQRR: Conversational Query Rewriting for Retrieval with Reinforcement Learning. In *EMNLP*. 10000–10014.

[55] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *ICLR*.

[56] Elad Yom-Tov, Shai Fine, David Carmel, and Adam Darlow. 2005. Learning to Estimate Query Difficulty: Including Applications to Missing Content Detection and Distributed Information Retrieval. In *SIGIR*. 512–519.

[57] Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot Generative Conversational Query Rewriting. In *SIGIR*. 1933–1936.

[58] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot Conversational Dense Retrieval. In *SIGIR*. 829–838.

[59] Hamed Zamani, W Bruce Croft, and J Shane Culpepper. 2018. Neural Query Performance Prediction Using Weak Supervision from Multiple Signals. In *SIGIR*.

105–114.

[60] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating Clarifying Questions for Information Retrieval. In *WWW*. 418–428.

[61] Hamed Zamani, Johanne R Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational Information Seeking. *arXiv preprint arXiv:2201.08808* (2022).

[62] Yun Zhou and W Bruce Croft. 2007. Query Performance Prediction in Web Search Environments. In *SIGIR*. 543–550.

[63] Jie Zou, Mohammad Aliannejadi, Evangelos Kanoulas, Maria Soledad Pera, and Yiqun Liu. 2022. Users Meet Clarifying Questions: Toward a Better Understanding of User Interactions for Search Clarification. *TOIS* (2022).