UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Query Performance Prediction Using Relevance Judgments Generated by Large Language Models

Meng, Chuan; Arabzadeh, Negar; Askari, Arian; Aliannejadi, Mohammad; de Rijke, Maarten

DOI
10.1145/3736402

Publication date
2025

Document Version
Final published version

Published in
ACM Transactions on Information Systems

License
CC BY

Link to publication

Citation for published version (APA):
Meng, C., Arabzadeh, N., Askari, A., Aliannejadi, M., & de Rijke, M. (2025). Query Performance Prediction Using Relevance Judgments Generated by Large Language Models. *ACM Transactions on Information Systems*, *43*(4), Article 106. https://doi.org/10.1145/3736402

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:12 Oct 2026

Query Performance Prediction Using Relevance Judgments Generated by Large Language Models

CHUAN MENG, University of Amsterdam, Amsterdam, Netherlands
NEGAR ARABZADEH, University of Waterloo, Waterloo, Ontario, Canada
ARIAN ASKARI, Leiden University, Leiden, Netherlands
MOHAMMAD ALIANNEJADI and MAARTEN DE RIJKE, University of Amsterdam, Amsterdam, Netherlands

Query performance prediction (QPP) aims to estimate the retrieval quality of a search system for a query without human relevance judgments. Previous QPP methods typically return a single scalar value and do not require the predicted values to approximate a specific information retrieval (IR) evaluation measure, leading to certain drawbacks: (i) a single scalar is insufficient to accurately represent different IR evaluation measures, especially when metrics do not highly correlate, and (ii) a single scalar limits the interpretability of QPP methods because solely using a scalar is insufficient to explain QPP results. To address these issues, we propose a QPP framework using automatically *gen*erated *re*levance judgments (QPP-GenRE), which decomposes QPP into independent subtasks of predicting the relevance of each item in a ranked list to a given query. This allows us to predict any IR evaluation measure using the generated relevance judgments as pseudo-labels. This also allows us to interpret predicted IR evaluation measures, and identify, track, and rectify errors in generated relevance judgments to improve QPP quality. We predict an item's relevance by using *open source* large language models (LLMs) to ensure scientific reproducibility.

We face two main challenges: (i) excessive computational costs of judging an entire corpus for predicting a metric considering recall, and (ii) limited performance in prompting open source LLMs in a zero-/few-shot manner. To solve the challenges, we devise an approximation strategy to predict an IR measure considering recall and propose to fine-tune open source LLMs using human-labeled relevance judgments. Experiments on the TREC 2019–2022 deep learning tracks and CAsT-19–20 datasets show that QPP-GenRE achieves state-of-the-art QPP quality for both lexical and neural rankers.

CCS Concepts: • **Information systems** → **Evaluation of retrieval results**;

This research was partially supported by the China Scholarship Council (CSC) under Grant No. 202106220041, 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, project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21, which is (partly) financed by the Dutch Research Council (NWO), project ROBUST with project number KICH3.LTP.20.006, which is (partly) financed by the Dutch Research Council (NWO) and the Dutch Ministry of Economic Affairs and Climate Policy (EZK) under the program LTP KIC 2020-2023, and the FINDHR (Fairness and Intersectional Non-Discrimination in Human Recommendation) project that received funding from the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101070212. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Authors' Contact Information: Chuan Meng (corresponding author), University of Amsterdam, Amsterdam, Netherlands; e-mail: c.meng@uva.nl; Negar Arabzadeh, University of Waterloo, Waterloo, Ontario, Canada; e-mail: narabzad@uwaterloo.ca; Arian Askari, Leiden University, Leiden, Netherlands; e-mail: a.askari@liacs.leidenuniv.nl; Mohammad Aliannejadi, University of Amsterdam, Amsterdam, Netherlands; e-mail: m.aliannejadi@uva.nl; Maarten de Rijke, University of Amsterdam, Amsterdam, Netherlands; e-mail: m.derijke@uva.nl.

This work is licensed under Creative Commons Attribution International 4.0.

© 2025 Copyright held by the owner/author(s).
ACM 1558-2868/2025/7-ART106
https://doi.org/10.1145/3736402

Additional Key Words and Phrases: Query performance prediction, Large language models, Relevance judgments, Relevance prediction, Re-ranking, Conversational search

ACM Reference format:
Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi, and Maarten de Rijke. 2025. Query Performance Prediction Using Relevance Judgments Generated by Large Language Models. *ACM Trans. Inf. Syst.* 43, 4, Article 106 (July 2025), 35 pages.
https://doi.org/10.1145/3736402

1 Introduction

Query performance prediction (QPP), a.k.a. query difficulty prediction, has attracted the attention of the **information retrieval (IR)** community throughout the years [8–10, 17, 49]. QPP aims to estimate the retrieval quality of a search system for a query without using human-labeled relevance judgments [47]. Effective QPP benefits various downstream applications [42], e.g., query variant selection [36, 112, 124], selective query expansion [4], IR system configuration selection [34, 127], enriching query features for learning-to-rank [80], and query-specific pool depth prediction [50] to reduce human relevance judgment costs.

Current Limitations. QPP methods can be applied in various domains and scenarios [42, 88]. We are usually concerned with the predicted retrieval quality w.r.t. various IR measures across different scenarios, e.g., our emphasis might be on precision [41, 44] for conversational search [2, 72] and on recall for legal search [126]. However, existing QPP approaches typically predict only a single real-valued score that indicates the retrieval quality for a query [49] and do not require the predicted score to approximate a specific IR evaluation measure [6, 114, 115, 122, 146].

These properties results in two key limitations: (i) While predicted performance scores have been shown to correlate with some IR evaluation metrics [32, 49], relying on a single value to represent different IR evaluation measures leads to a "one size fits all" approach, which is problematic because the literature shows that some IR metrics do not correlate well and the agreement varies across scenarios and queries [54, 62]. Although some studies train regression-based QPP models to predict a specific IR evaluation measure [7, 20, 30, 55, 63], they require training separate models to predict different measures, leading to lots of storage and running costs. (ii) A single-score prediction limits the interpretability of QPP. It is insufficient to explain QPP outputs or to analyze and fix inaccurate QPP results based solely on a single score. We argue that more in-depth and interpretable insights into QPP outputs are required.

A Novel QPP Framework. We propose a QPP framework using automatically *gen*erated *r*elevance judgments (QPP-GenRE), in which we decompose QPP into independent subtasks of automatically predicting the relevance of each item in a ranked list for a given query. QPP-GenRE comes with various advantages: (i) It allows us to directly predict any desired IR evaluation measure at no additional cost, using automatically generated relevance judgments as pseudo-labels. Compared to most existing QPP methods that only outputting a single scalar value as an indicator of a ranker's overall performance, our method provides a multi-dimensional assessment of system effectiveness by enabling the calculation of various metrics from the same set of relevance judgments. Leveraging predicted relevance judgments, our method allows the computation of metrics such as **normalized discounted cumulative gain (nDCG)**@k, Precision@k, **reciprocal rank (RR)**, and so on. This flexibility is particularly advantageous because it allows us to use QPP to predict retrieval quality in terms of a specific evaluation metric that is prioritized in different scenarios. For example, we use predicted relevance judgments to calculate precision-oriented metrics in conversational search,

while focusing on recall-oriented metrics for tasks like legal search. (ii) The generated relevance judgments provide an explanation beyond simply gauging how difficult or easy a query is by offering information about why the query is predicted as being difficult or easy; moreover, we can translate the "QPP errors" into easily observable "relevance judgment errors," e.g., false positives or negatives, informing potential ways of improving QPP quality by fixing observed relevance judgment errors.

Integrating QPP-GenRE with Large Language Model (LLM)-Labeled Relevance Judgments. QPP-GenRE can be integrated with various approaches for judging relevance. The success of QPP-GenRE depends fundamentally on the accuracy of relevance judgment predictions. Therefore, it is crucial to equip QPP-GenRE with an approach capable of accurately generating relevance judgments. Recently, numerous studies [40, 52, 75, 79, 120, 125, 130] have shown the potential effectiveness of using LLMs to generate relevance judgments. Therefore, it is natural to explore equipping QPP-GenRE with LLMs for judging relevance. However, those studies have certain limitations: Several authors have prompted commercial LLMs (e.g., ChatGPT, GPT-3.5/4, GPT-4o) to generate relevance judgments (e.g., [19, 40, 75, 125, 129, 131, 140]); commercial LLMs come with limitations like non-reproducibility, non-deterministic outputs, and potential data leakage between pretraining and evaluation data, impeding their value in scientific research [102, 103, 142]. Although MacAvaney and Soldaini [79] prompt small-scale open source language models (e.g., Flan-T5 [21] with 3B parameters) for generating relevance judgments, they focus on a setting wherein the model is already given one relevant item for each query, which does not apply to QPP as we typically do not know any relevant item for a query in advance. In this article, we focus on the use of *open source* LLMs for generating relevance judgments in a realistic setting where we lack prior knowledge of any relevant items for a query. There are only few studies [64, 109, 129, 140] attempting to prompt open source LLMs in this setting.

Challenges. We face two challenges when using QPP-GenRE for QPP: (i) predicting IR metrics that not only consider precision but also take recall into account, ideally, entails identifying all relevant items in the entire corpus for a query; however, using an LLM to judge the entire corpus per query is impractical due to the significant computational overhead; (ii) our experiments reveal that directly prompting open source LLMs in a zero-/few-shot manner yields limited effectiveness in predicting relevance, resulting in limited QPP quality; this aligns with recent findings indicating limited success in prompting open source LLMs for specific tasks [104]. Also, incorporating **in-context learning (ICL)** examples in few-shot prompting leads to high inference costs [70].

Solutions. To address the challenges listed above, (i) we devise an approximation strategy to predict IR measures considering recall by only judging a few items in the ranked list for a query and using them to estimate the metric, hence avoiding the cost of traversing the entire corpus to identify all relevant items for a query; the approximation strategy also enables us to investigate the impact of various judging depths in the ranked list on QPP quality; and (ii) we enhance an open source LLM's ability to generate relevance judgments by training it with **parameter-efficient fine-tuning (PEFT)** [33] on human-labeled relevance judgments; unlike previous supervised QPP methods that need to train separate models for predicting different IR evaluation measures, training LLMs to judge relevance is agnostic to a specific IR metric.

Experiments. Experiments on datasets from the **TREC 2019–2022 deep learning (TREC-DL)** tracks [22–25] show that QPP-GenRE achieves state-of-the-art QPP quality in estimating the retrieval quality of a lexical ranker (BM25) and two neural rankers, ANCE [134] and TAS-B [57], in terms of RR@10, a precision-oriented IR metric, and nDCG@10, an IR metric considering recall (see Sections 6.1 and 6.2).

We also find that using LLMs to directly model QPP, i.e., asking LLMs to directly generate values of IR evaluation metrics, performs much worse than QPP-GenRE. This finding reveals that QPP-GenRE is a more effective way of modeling QPP using LLMs. Furthermore, our experiments demonstrate the effectiveness of our devised approximation strategy in nDCG@10: QPP-GenRE achieves state-of-the-art QPP quality at the shallow judging depth 10, and QPP-GenRE's QPP quality reaches saturation when it further judges up to 100–200 retrieved items in a ranked list (see Section 7.1).

Moreover, we conduct an in-depth analysis to investigate the impact of fine-tuning and the choice of LLMs on the quality of generated relevance judgments and QPP. We consider two families of LLMs, Llama and Mistra, with sizes ranging from 1B to 70B, under both few-shot and fine-tuned settings. We find that fine-tuning markedly improves the quality of relevance judgment generation and QPP for all LLMs. In particular, a fine-tuned 3B model (Llama-3.2-3B-Instruct) provides the best tradeoff between QPP quality and computational efficiency: It not only significantly outperforms 70B few-shot models, but also achieves QPP quality comparable to that of fine-tuned 7B and 8B models. This suggests that, compared to few-shot prompting, fine-tuning LLMs for relevance prediction can yield higher effectiveness in both relevance prediction and QPP, even with relatively small model sizes; this, in turn, implies that fine-tuning can offer strong performance at lower inference costs. Moreover, the performance of fine-tuned LLMs in terms of judging relevance exceeds that of a commercial LLM (GPT-3.5) [40] (see Section 7.2).

Additionally, to show QPP-GenRE's compatibility with other types of relevance prediction methods, we adapt a state-of-the-art pointwise LLM-based re-ranker, RankLLaMA [76], into a relevance judgment generator by applying a threshold to its re-ranking scores. Our results indicate that QPP-GenRE integrated with RankLLaMA achieves high QPP quality, at the cost of tuning a proper threshold. The high QPP quality achieved by RankLLaMA demonstrates QPP-GenRE's compatibility with other types of relevance prediction methods (see Section 7.3).

To demonstrate the generalizability of QPP-GenRE to a new domain, we conduct experiments of applying QPP-GenRE to conversational search [89, 94] in a zero-shot manner. Specifically, we evaluate QPP-GenRE and baselines when predicting the performance of a conversational dense retriever [137] on the CAsT-19 [29] and 20 [28] datasets. We found that QPP-GenRE consistently outperforms all baselines on both datasets, demonstrating strong generalizability (see Section 7.4).

We also analyze QPP errors based on automatically generated relevance judgments, and provide a case study for a specific example, demonstrating QPP-GenRE's interpretability (see Section 7.5).

Finally, our computational cost analysis shows that QPP-GenRE shows lower latency than some supervised QPP baselines when predicting multiple measures because multiple measures can be derived from the same set of relevance judgments. Although QPP-GenRE shows higher latency than other QPP baselines when predicting only one metric, QPP-GenRE's latency is still 20 times smaller than the state-of-the-art GPT-4-based listwise re-ranker [119]. To further enhance the efficiency of QPP-GenRE, we have proposed a *relevance judgment caching mechanism*. Our experimental results show that the mechanism can reduce LLM calls for relevance prediction by about 30%. Specifically, this mechanism reuses previously predicted relevance judgments for the same query when predicting the performance of new rankers. As a result, this mechanism helps conserve computational resources by avoiding recompute relevance judgments that are shared among multiple rankers (see Section 7.6).

Application Scenarios. Given QPP-GenRE's high QPP quality and interpretability, it is well-suited for some knowledge-intensive professional search scenarios, where accurate QPP is prioritized, interpretable QPP results are preferred, and users may have a higher tolerance level for latency than

users in web search. Plus, QPP-GenRE can be used to analyze how well a search system performs in offline settings [44], where latency is not necessarily an issue.

One might argue, if QPP-GenRE needs to be integrated with an LLM to predict ranking quality, why not directly use the LLM for re-ranking? However, we reveal that QPP-GenRE integrated with LLaMA-7B already achieves high QPP quality and remains significantly more efficient than costly state-of-the-art LLM-based re-rankers (e.g., the GPT-4-based listwise re-ranker [119]). Calling those expensive LLM-based re-rankers is often unnecessary, as many initial rankings are good enough and either do not require re-ranking or only need very shallow re-ranking depths [87]. Therefore, sufficiently accurate QPP for initial rankings is needed to guide the decision on whether to use the expensive re-ranker, or to determine the optimal re-ranking depth that does not waste computational resources. Given QPP-GenRE's substantial improvements in QPP quality over previous QPP methods and significantly lower latency compared to those expensive re-rankers, it is valuable to make QPP-GenRE work with state-of-the-art, yet much more costly, LLM-based re-rankers [119] to achieve a better balance between effectiveness and efficiency in re-ranking.

Another advantage of QPP-GenRE, which makes it applicable to real-world scenarios, is that unlike traditional approaches that depend heavily on the specific properties of rankers, our method is ranker-agnostic, e.g., conventional baselines often rely on score distributions tied to the type of their rankers, making their predictions inherently ranker-dependent. Previous study has demonstrated that the effectiveness of such score-based QPP methods varies across different rankers due to the differences in score distributions produced by each ranker [46]. In contrast, QPP-GenRE operates on individual query–document pairs and evaluating them independently of a ranker. This eliminates the dependency on specific ranker characteristics and score distributions, ensuring that our framework can be applied generally across various retrieval settings. Furthermore, QPP-GenRE can leverage the reusability of predicted relevance judgments. Since each query–document pair is judged only once, our method allows for predicting the performance of multiple rankers effectively due to their potential overlaps in their top ranked documents. This implies that QPP-GenRE can become more efficient over time as it is used in practice.

Reproducibility. To facilitate future research, we release our data, scripts for fine-tuning/inference, sampled demonstration examples for few-shot prompting, and fine-tuned checkpoints of various LLMs at https://github.com/ChuanMeng/QPP-GenRE.

Contributions. Our main contributions are as follows:

- We propose a novel QPP framework using automatically generated relevance judgments (QPP-GenRE), which decomposes QPP into independent subtasks of predicting the relevance of each item in a ranked list to the query, and predicts different IR evaluation measures based on the relevance predictions.
- We devise an approximation strategy to predict IR measures that account for both precision and recall, avoiding the cost of traversing the entire corpus to identify all relevant items for a query.
- We fine-tune leading *open source* LLMs from the Llama and Mistral families, covering a range of model sizes, for the task of automatically generating relevance judgments. Our results show that fine-tuning much smaller LLMs for relevance judgment prediction can yield more effective relevance prediction and QPP than few-shot prompting with much larger models.
- We conduct experiments on four datasets, showing that QPP-GenRE outperforms the state-of-the-art QPP baselines on the TREC-DL 19–22 datasets in predicting RR@10 and nDCG@10 in terms of Pearson's ρ and Kendall's τ.

2 Related Work

Our work is relevant to four strands of research: QPP (Section 2.1), zero/few-shot prompting and PEFT for LLMs (Section 2.2), LLMs for generating relevance judgments (Section 2.3), and LLMs for re-ranking (Section 2.4).

2.1 QPP

QPP has attracted lots of attention in the IR and NLP community and has been widely studied in *ad-hoc* search [32, 45, 46, 115], conversational search [3, 43, 44, 83–86, 118], question answering [55, 110], and image retrieval [100]. This article focuses on QPP for *ad-hoc* search.

Typically, QPP methods are divided into two categories: pre- and post-retrieval methods [17]. The former predicts the difficulty of a given query by using features of the query and corpus, while the latter further uses features of a ranked list returned by a ranker for the query [17]. This article focuses on post-retrieval QPP methods.

A large number of unsupervised and supervised post-retrieval QPP methods have been proposed [17] for predicting the performance of lexical rankers, such as query likelihood [66] and BM25 [106]. Unsupervised QPP methods can be classified into clarity-based [26], robustness-based [14, 145, 146], coherence-based [5, 37], and score-based [27, 99, 114, 122, 146]. More recently, a set of supervised QPP methods have been proposed [7, 20, 30, 31, 55, 63, 138]. NeuralQPP [138] and Deep-QPP [30] are optimized from scratch. NQA-QPP [55] and BERT-QPP [7] fine-tune BERT [35] to improve QPP effectiveness. Further, Datta et al. [32] propose qppBERT-PL, which considers list-wise-document information, while Chen et al. [20] propose BERT-groupwise-QPP that considers both cross-query and cross-document information. Khodabakhsh and Bagheri [63] propose a **multi-task query performance prediction framework (M-QPPF)**, learning document ranking and QPP simultaneously.

Post-retrieval QPP methods designed for lexical rankers struggle to predict the retrieval quality of neural rankers [46, 55], motivating several new unsupervised post-retrieval QPP methods designed for neural rankers. Datta et al. [31] propose a weighted relative information gain-based model, which assesses a neural ranker for a given query by considering the relative difference of predicted performance between the given query and its variants; Zendel et al. [139] assess a neural re-ranker by measuring the entropy of scores returned by it; Faggioli et al. [45] propose neural-ranker-specific ways of calculating regularization terms used by unsupervised post-retrieval QPP methods; Vlachou and Macdonald [132] propose an unsupervised coherence-based QPP method that employs neural embedding representations to assess dense retrievers; and Singh et al. [115] propose **pairwise rank preference-based QPP (QPP-PRP)** for predicting the performance of a neural ranker by measuring the degree to which a pairwise neural re-ranker (e.g., DuoT5 [101]) agrees with the ranked list returned by the neural ranker.

We present a novel QPP perspective: We start by automatically generating relevance judgments for a ranked list for a query and then proceed to predict IR evaluation measures for the ranked list. To the best of our knowledge, no prior work addresses QPP from this perspective.

Unlike regression-based QPP models [7, 20, 30, 55, 63], which require training separate models to predict different IR evaluation measures, the training of LLMs for judging relevance in the QPP-GenRE method that we propose is agnostic to a specific IR evaluation measure, and different measures can be derived from the same set of generated relevance judgments.

We also differ from qppBERT-PL [32], which first predicts the number of relevant items for each chunk in a ranked list and then aggregates those numbers into a general QPP score. However, qppBERT-PL's output is still presented as a single scalar, which is insufficient to accurately represent

different evaluation measures; also, it is infeasible to predict arbitrary IR measures only using the number of relevant items in a ranked list.

The work closest to QPP-GenRE, which is still different, is QPP using **effectiveness evaluation without relevance judgments (EEwRJ)** [90]. The goal of EEwRJ methods is to predict search system effectiveness in a TREC-like environment, e.g., a method proposed by Soboroff et al. [116] randomly samples items from a pool for a query and treats these items as relevant; the intuition is that if an item is ranked highly by many search systems, it is likely to be pooled and therefore considered relevant. Mizzaro et al. [90] explore applying QPP EEwRJ [90] methods to QPP. However, QPP using EEwRJ suffers from two limitations: (i) EEwRJ requires obtaining ranked lists returned by all search systems in a given TREC edition to predict the difficulty of a query, and (ii) EEwRJ encounters normalization challenges when predicting the ranking quality for a ranked list returned by a specific search system [90]. QPP-GenRE does not face these limitations.

2.2 Zero/Few-Shot Prompting and PEFT for LLMs

While fine-tuning pre-trained language models has given rise to many state-of-the-art results [35], fully fine-tuning LLMs for a specific task on consumer-level hardware is typically infeasible [147] because of the large number of parameters of LLMs. As a result, there are three prevailing ways to adapt LLMs to a specific task: zero-shot prompting, few-shot prompting [12, 13], a.k.a. ICL [16, 38], and PEFT [33, 59, 70].

There is limited success in only prompting open source LLMs for certain tasks [104]. Zero-shot prompting instructs an LLM to perform a specific task by inputting a text instruction. To get a promising result, zero-shot prompting is usually based on instruction-tuned LLMs [104, 141], such as Flan-T5 [21] and Flan-UL2 [123]. However, Sun et al. [117] show that the performance of zero-shot prompting degrades considerably if an LLM is fed an instruction that was not observable during its training. ICL inputs a few input-target pairs (a.k.a. demonstrations) to an LLM, which would make an LLM learn from analogy [38] without updating its parameters. However, ICL has a high computational cost because it needs to feed input-target pairs to an LLM for each prediction; also, ICL requires substantial manual prompt engineering because an LLM's performance [70] is sensitive to the formatting of the prompt (e.g., the wording and the order of input-target pairs).

PEFT can solve the above limitations; it aims to adapt an LLM to a specific task by training only a small fraction of its parameters. **Low-rank adaptation (LoRA)**, a widely used PEFT method [51, 71, 111, 143], has been shown to achieve comparable performance to full-model fine-tuning [33, 74]; LoRA adds learnable low-rank adapters to each network layer of an LLM [59] while all original parameters of the LLM are frozen. QLoRA [33] further reduces the memory usage of LoRA without sacrificing performance; QLoRA first quantizes an LLM model to 4-bits before adding and optimizing low-rank adapters. Our work explores the possibility of training open source LLMs with QLoRA to generate relevance judgments.

2.3 LLMs for Generating Relevance Judgments

Automatically generating relevance judgments is a long-standing goal in IR that has been studied for multiple decades [81, 82, 97, 98, 105, 116]. Recent studies have demonstrated promising results of using LLMs for the automatic generation of relevance judgments [40, 125]. In this article we focus on studies into generating relevance judgments with discrete classes (e.g., "Relevant" or "Irrelevant"), instead of generating continuous relevance labels in real numbers [136]. We discuss related studies into LLM-based automatic generation of relevance judgments from two dimensions: (i) how LLMs are used to generate relevance judgments, and (ii) their applications.

Recent studies have explored prompting commercial LLMs (e.g., GPT-3.5 and GPT 4) or open source LLMs in zero- or few-shot manners. Specifically, Faggioli et al. [40] use zero- and few-shot

prompting to instruct GPT-3.5 to predict the relevance of an item to a query. Thomas et al. [125] instruct GPT-4 by zero-shot prompting, and add to the prompt a detailed query description and consider chain-of-thought [133]. Ma et al. [75] instruct GPT-3.5 to generate relevance judgments for a domain-specific scenario, i.e., legal case retrieval [78]; they use prompts specifically designed for this scenario. More recently, Upadhyay et al. [131] prompt GPT-4o in a zero-shot manner. Besides using commercial LLMs, only few studies [64, 109, 129, 140] explore prompting open source LLMs to generate relevance judgments, e.g., Khramtsova et al. [64], Upadhyay et al. [129], and Salemi and Zamani [109] prompt Flan-T5 [21], Vicuña-7B [144], and Mistral [61], respectively, in either zero-shot or few-shot manners. MacAvaney and Soldaini [79] focus on a special scenario where a relevant item for a given query is already known and use Flan-T5 [21] to estimate the relevance of another item to the query given the known relevant item.

Recent studies have explored using LLM-generated relevance judgments to benefit (i) search system evaluation [40, 79, 125, 129], (ii) ranker selection [64], (iii) item selection and retrieval quality evaluation in **retrieval-augmented generation (RAG)** [109, 140], and (iv) retriever fine-tuning [75]. Concerning (i), recent studies [1, 40, 79, 125, 129] explore evaluating search systems either entirely using LLM-generated relevance judgments or partially using LLM-generated relevance judgments (a.k.a. filling holes). They have demonstrated a high correlation between search system rankings based on LLM- and human-labeled relevance judgments. As to (ii), given a pool of dense retrievers, Khramtsova et al. [64] select a suitable one for a target corpus by estimating their performance using LLM-generated queries and relevance judgments specific to the target corpus. For (iii), for item selection, Zhang et al. [140] prompt LLMs to generate relevance judgments for retrieved candidate items in RAG; the items that are predicted as "relevant" are used for text generation. Zhang et al. [140] observe that items selected via relevance prediction resulted in sub-optimal text generation quality. For retrieval quality evaluation, Salemi and Zamani [109] generate relevance judgments for retrieved candidate items and aggregate those judgments into a score. However, Salemi and Zamani [109] found that the aggregated score based on the LLM-generated relevance judgments achieves a low correlation with the text generation quality of RAG. Concerning (iv), Ma et al. [75] fine-tune a legal case retriever on a training set augmented with LLM-generated relevance judgments. They show that fine-tuning a legal case retriever using the generated relevance judgments results in enhanced performance.

Our work differs from the studies mentioned above: (i) we explore the possibility of *fine-tuning* open source LLMs for generating relevance judgments; unlike MacAvaney and Soldaini [79], we focus on a more practical scenario wherein no relevant item is known in advance for each query; and (ii) we focus on QPP and predict the ranking quality of a ranked list for a query using LLM-generated relevance judgments, which previous studies have not explored.

2.4 LLMs for Re-Ranking

Recent studies on using LLMs for re-ranking have witnessed remarkable progress [11, 15, 39, 58, 76, 77, 87, 102, 103, 108, 119, 121, 142, 148–150]. There are four paradigms of LLM-based re-ranking: pointwise, pairwise, listwise, and setwise [150]. Given a query, pointwise re-rankers produce a relevance score for each item independently, and the final ranking is formed by sorting items by relevance score [39, 76, 108, 148]. The pairwise paradigm [104] eliminates the need for computing relevance scores; given a query and a pair of items, a pairwise re-ranker estimates whether one item is more relevant than the other for the query. Listwise re-rankers [77, 102, 103, 119, 121, 142] frame re-ranking as a pure generation task and directly output the reordered ranked list given a query and a ranked list return by first-stage retriever [77, 102, 103, 119, 121, 142]. Given the low efficiency of pairwise (multiple inference passes) and listwise (multiple decoding steps) re-rankers, the setwise paradigm [150] is meant to improve the efficiency while retaining re-ranking effectiveness. Given a

query and set of items, an LLM is asked which item is the most relevant one to the query; these items are reordered according to the LLM's output logits of each item being chosen as the most relevant item to the query, which only requires one decoding step of an LLM.

Our work differs from this line of research because we generate explicit relevance judgments with discrete classes (e.g., "Relevant" or "Irrelevant"), whereas studies into LLMs for re-ranking aim to predict the relevance order of items. However, using LLMs for generating relevance judgments and for re-ranking are intrinsically the same task: relevance prediction. Thus, an LLM-based re-ranker has the potential to serve as a relevance judgment generator.

Our *main contribution* in this article is the introduction of QPP-GenRE, a novel QPP framework, which, in theory, can be integrated with various relevance prediction approaches. To demonstrate the compatibility of QPP-GenRE with various relevance prediction approaches, we adapt a state-of-the-art pointwise LLM-based re-ranker, RankLLaMA [76], into a relevance judgment generator by applying a threshold for its re-ranking scores; we then integrate QPP-GenRE with this adapted RankLLaMA. It is important to note that exploring the use of other types of LLM-based re-rankers (e.g., pairwise and listwise) as relevance judgment generators falls outside the scope of this article.

3 Task Definition

In this article, we focus on post-retrieval QPP [17]. Generally, a post-retrieval QPP method ψ aims to estimate the retrieval quality of a ranked list $L = [d_1, \ldots, d_i, \ldots, d_{|L|}]$ with $|L|$ retrieved items induced by a ranker M over a corpus C in response to query q without human-labeled relevance judgments, formally:

$$p = \psi(q, L, C) \in \mathbb{R}, \tag{1}$$

where p indicates the predicted retrieval quality of the ranker M in response to the query q; typically, p is expected to be correlated with an IR evaluation measure, such as RR.

4 Method

4.1 Overview of QPP-GenRE

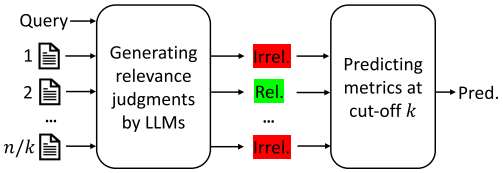
We propose QPP-GenRE, which consists of two steps: (i) generating relevance judgments using LLMs, and (ii) predicting IR evaluation measures (see Figure 1). In (i), we employ an LLM to generate relevance judgments for the top-n retrieved items in the ranked list for a given query; to improve LLMs' effectiveness in generating relevance judgments, we fine-tune an LLM with PEFT using human-labeled relevance judgments. In (ii), we regard the generated relevance judgments as pseudo-labels to calculate different IR evaluation measures.

4.2 Generating Relevance Judgments Using LLMs

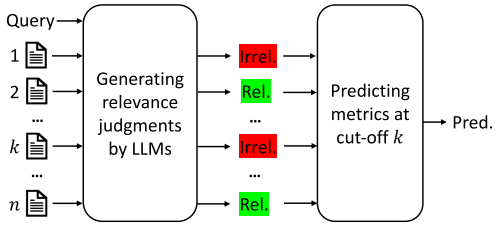
4.2.1 Inference. Given the ranked list $L = [d_1, \ldots, d_i, \ldots, d_{|L|}]$ with $|L|$ items returned by a ranker M for a query q, an LLM is employed to automatically predict the relevance of each item in the top-n positions of the ranked list L to the query q, formally:

$$\hat{r}_i = \text{LLM}(\text{prompt}(q, d_i)), \tag{2}$$

where $\text{prompt}(\cdot, \cdot)$ is a prompt to instruct an LLM on the task of automatic generation of relevance judgments, as illustrated in Figure 2. We follow the design proposed by Faggioli et al. [40] to create a prompt that explicitly instructs the LLM to output either "Relevant" or "Irrelevant." In our preliminary experiments, we also tested the prompt from Sun et al. [119], which asks the LLM the question, "Does the passage answer the query?" and expects a response of either "Yes" or "No." However, we found that this alternative prompt produced inferior results compared to our chosen design. $\hat{r}_i$ is a predicted relevance value for the item d_i at rank i. $\hat{r}_i \in \{1, 0\}$, where "1"



(a) Predicting a precision-based metric. In this case, we predicts the top-n items in the given ranked list, n is equal to k, which is the cut-off point for a precision-oriented metric, such as reciprocal rank (RR). E.g., the predicted $RR@k$ for the ranked list in this figure is 0.5 as the second item is predicted as relevant.



(b) Predicting a metric considering recall using our proposed *approximation strategy*. It only judges the top-n ($n \ll$ the total number of items in the corpus) items in the given ranked list and uses the items predicted as relevant to approximate all relevant items in the corpus.

Fig. 1. The framework of QPP-GenRE.

Instruction: Please assess the relevance of the provided passage to the following question.
Please output "Relevant" or "Irrelevant".
Question: {question}
Passage: {passage}
Output: Relevant/Irrelevant

Fig. 2. Prompt used by LLMs for automatic generation of relevance judgments.

indicates relevant and "0" irrelevant. We leave the prediction of multi-graded labels as future work. After automatically judging the top-n items in the ranked list L, we get a list of generated relevant judgments $\hat{\mathcal{R}}_{L_{1:n}} = [\hat{r}_1, \ldots, \hat{r}_i, \ldots, \hat{r}_n]$, where $\hat{r}_i$ is the predicted relevance value for d_i in L.

4.2.2 PEFT. To further improve an LLM's effectiveness in generating relevance judgments, we use human-labeled relevance judgments to train an LLM with an effective PEFT method, QLoRA [33]. Specifically, we first quantize an LLM model to 4-bit, add learnable low-rank adapters to each network layer of the LLM, and then optimize low-rank adapters. Formally, given the query q and an item d_i in the ranked list L, we optimize the LLM to generate the human-labeled relevance value r_i for the item d_i:

$$\mathcal{L}(\theta_{LoRA}) = -\frac{1}{M}\sum_{i=1}^{M} \log P(r_i \mid \text{prompt}(q, d_i)), \tag{3}$$

where θ_{LoRA} stands for learnable low-rank adapters added to the LLM; M is the number of training examples. See Section 5.6 for more details.

4.3 Predicting IR Evaluation Measures

4.3.1 Predicting Precision-Oriented Measures. We compute a precision-oriented measure based on LLM-generated relevant judgments $\hat{\mathcal{R}}_{L_{1:n}}$ for the top-n items in the ranked list L, as shown in Figure 1(a). Note that in this case, $n = k$. The following is an example to compute RR@k:

$$RR@k = 1/\min_i\{\hat{r}_i > 0\}, \tag{4}$$

where $0 < i \leqslant k$. For instance, as illustrated in Figure 1(a), the first item in the ranked list is predicted as irrelevant, while the second item is predicted as relevant. In this case, the predicted

RR@k value would be 0.5. *RR@k* would be equal to 0 if there is no top-k item that is predicted as relevant to the query q.

4.3.2 An Approximation Strategy to Predict Measures Considering Recall. As the computation of a measure considering recall requires the information of all relevant items in the corpus C for a given query q, we need to automatically assess every item in corpus C, which is infeasible due to the high computational cost. To address this issue, we devise an approximation strategy for predicting an IR measure considering recall, which only judges the top-n ($n \ll$ the total number of items in the corpus) items in the ranked list L and uses the items predicted as relevant to approximate all relevant items in the corpus, to avoid the cost of judging the entire corpus. Fröbe et al. [48], Lu et al. [73], and Moffat [96] define nDCG [60] at a cutoff k as a recall-oriented IR evaluation metric because it is normalized by a recall-oriented "best possible" ranking.[1] nDCG@10 is also the most primary official IR evaluation metric in TREC-DL 19–22 [22–25]. Thus, here we show an example of predicting nDCG@k [60], formally:

$$nDCG@k = DCG@k/IDCG@k, \tag{5}$$

where *DCG@k* can be computed easily using the generated relevance judgments for the top-k items in the ranked list L, namely:[2]

$$DCG@k = \hat{r}_1 + \sum_{i=2}^{k} \hat{r}_i / \log_2 i. \tag{6}$$

IDCG@k is the ideal ranked list with k items, which requires knowing all the relevant items in the corpus C. We approximate all relevant items in the corpus by considering the items that are predicted as relevant at the top-n ranks in the ranked list L, and compute *IDCG@k* based on that. First, we reorder the LLM-generated relevant judgments $\hat{\mathcal{R}}_{L_{1:n}} = [\hat{r}_1, \ldots, \hat{r}_i, \ldots, \hat{r}_n]$ for the ranked list L into $\hat{\mathcal{R}}_{iL_{1:n}} = [\hat{ir}_1, \ldots, \hat{ir}_i, \ldots, \hat{ir}_n]$ in descending order of predicted relevance; then, we compute *IDCG@k* based on $\hat{\mathcal{R}}_{iL_{1:n}}$, namely:

$$IDCG@k = \hat{ir}_1 + \sum_{i=2}^{k} \hat{ir}_i / \log_2 i. \tag{7}$$

5 Experimental Setup

5.1 Research Questions (RQs)

In this section, we study the following RQs:

- *RQ1* To what extent does QPP-GenRE improve QPP quality for lexical and neural rankers in terms of RR@10, a precision-oriented IR metric, compared to state-of-the-art baselines?
- *RQ2* To what extent does QPP-GenRE improve QPP quality for lexical and neural rankers in terms of nDCG@10, an IR metric that not only considers precision but also takes recall into account, compared to state-of-the-art baselines?
- *RQ3* How does judging depth in a ranked list affect the prediction of nDCG@10, an IR metric that considers both precision and recall? In other words, how does varying the number of top-ranked documents submitted for relevance judgments impact QPP quality?

[1] In this article, we employ nDCG@10 and believe that nDCG@10 is a metric considering recall: Figure 4 illustrates that to reach saturation in predicting nDCG@10 values for ANCE and BM25, judgments up to the top 100 and 200 retrieved items are needed, respectively. If it were a precision-based metric, saturation could be achieved by judging around 10 items.
[2] Note that we consider the definition of *DCG@k* for binary relevance labels.

RQ4 To what extent do fine-tuning and the choice of LLMs affect the quality of generated relevance judgments and QPP?

5.2 Datasets

We experiment with four widely used IR datasets from the TREC-DL tracks [22–25]. These datasets provide relevance judgments in multi-graded relevance scales per query. TREC-DL 19, 20, 21, and 22 have 43, 54, 53, and 76 queries, respectively. TREC-DL 19/20 and TREC-DL 21/22 are based on the MS MARCO V1 and MS MARCO V2 passage ranking collections, respectively. In the V1 edition, the corpus comprises 8.8 million passages while the V2 edition has over 138 million passages.

5.3 Retrieval Approaches

We consider BM25 [106] as a lexical ranker; we also consider ANCE [134] and TAS-B [57] as neural-based dense retrievers. To increase the comparability and reproducibility of our article, we get the retrieval results of both rankers using the publicly available resource from Pyserini [69]. We get BM25's retrieval result with top-1,000 retrieved items per query on the TREC-DL 19–22 datasets using the default parameters ($k1 = 0.9$, $b = 0.4$). BM25's actual nDCG@10 values are 0.506, 0.480, 0.446, and 0.269 on TREC-DL 19, 20, 21, and 22, respectively. We get the retrieval results of ANCE and TAS-B with top-1,000 retrieved items per query on TREC-DL 19–20, using the publicly available dense vector index of ANCE on MS MARCO V1. ANCE's actual nDCG@10 values are 0.645 and 0.646 on TREC-DL 19 and 20, respectively; TAS-B's actual nDCG@10 values are 0.721 and 0.685 on TREC-DL 19 and 20, respectively. We rely on the publicly available dense vector index of ANCE/TAS-B; at the time of writing, there is no dense vector index of ANCE/TAS-B publicly available on MS MARCO V2 for TREC-DL 21 and 22.[3]

5.4 QPP Baselines

We consider three groups of baselines: unsupervised post-retrieval QPP methods, supervised post-retrieval QPP methods, and the LLM-based QPP methods. Specifically, we consider the following unsupervised QPP approaches that already showed high correlation with actual retrieval performance in previous work:

- Clarity [26] computes the KL divergence between language models [68] induced from the top-k items in a ranked list and the corpus.
- **Weighted information gain (WIG)** [146] calculates the difference between retrieval scores of the top-k items in a ranked list and the retrieval score of the entire corpus.
- **Normalized query commitment (NQC)** [114] calculates the standard deviation of retrieval scores of the top-k items in a ranked list to a query; the standard deviation is normalized by the retrieval score of the entire corpus to the query.
- σ_{max} [99] computes the standard deviation of retrieval scores from the first item to each point in a ranked list and outputs the maximum standard deviation.
- n($\sigma_{x\%}$) [27] calculates the standard deviation for each query by considering the items whose retrieval scores are at least $x\%$ of the top retrieval score in a ranked list.
- **Score magnitude and variance (SMV)** [122] considers both the magnitude of retrieval scores (WIG) and their variance (NQC).
- UEF(NQC) [113] uses a pseudo-effective reference list to improve QPP quality; we follow [6, 7, 32] to use NQC as a base predictor.

[3]Building the dense vector index on MS MARCO V2 with over 138 million passages is resource-intensive and beyond the scope of our work.

Instruction: Evaluate the relevance of the ranked list of passages to the given query by providing a numerical score between 0 and 1. A score of "1" indicates that the ranked passages are highly relevant to the query, while a score of "0" means no relevance between the passages and the query.
Query: { }
Passage 1: { }
Passage 2: { }
...
Passage k: { }
Output:

Fig. 3. Prompt used by QPP-LLM.

—RLS(NQC) [107] generates and selects both pseudo-effective and pseudo-ineffective reference lists; we use NQC as a base predictor because Roitman [107] shows that RLS works better with NQC.
—QPP-PRP [115] measures the degree to which a pairwise neural re-ranker (DuoT5 [101]) agrees with the ranked list for the query.
—Dense-QPP [6] is robustness-based and designed for dense retrievers only: It injects noise neural representation of the given query, and then measures the similarity between ranked lists for the original query and perturbed query representations. Note that Dense-QPP [6] is designed for predicting the ranking quality of neural-based retrievers; it cannot predict the ranking quality of BM25.

Since studies show that BERT-based post-retrieval supervised QPP methods [7, 20, 32, 55] perform better than their neural-based counterparts, we only consider BERT-based supervised QPP approaches:

—NQA-QPP [55] is a regression-based method, which predicts a QPP score by using BERT representations for the query and query–item pairs, and the standard deviation of retrieval scores.
—BERTQPP [7] is a regression-based method, which predicts a QPP score by using BERT representations for the query and the top-ranked item. We use the cross-encoder version of BERTQPP because of its promising results.
—qppBERT-PL [32] first splits the ranked list into chunks, predicts the number of relevant items in each chunk, and calculates a weighted average of the number of relevant items in all chunks.
—M-QPPF [63] is also regression-based and models QPP and document ranking jointly, by adopting a shared BERT layer to learn representations for query-document pairs, and using two layers to model QPP and document ranking, respectively.

While to the best of our knowledge there is no LLM-based QPP method yet, to have a fair comparison with LLM-based approaches, we propose two LLM-based QPP baselines. Research on using LLMs for arithmetic tasks shows that LLaMA treats numbers as distinct tokens and can understand and generate numerical values [71]. Inspired by this, we prompt LLaMA-7B to directly generate a numerical score given a query and the ranked list with k passages for the query; the prompt is shown in Figure 3. We consider two variants:

— QPP-LLM (few-shot) uses ICL and inserts several demonstration examples after the instruction in the prompt; each example is composed of a query, k passages, and the actual performance in terms of an IR evaluation measure.
— QPP-LLM (fine-tuned) fine-tune LLaMA-7B to learn to directly generate numerical values of an IR metric, similar to the way other regression-based supervised QPP methods are trained.

5.5 QPP Evaluation and Target IR Evaluation Measures

We follow established best practices [17, 26, 32, 56, 138] to evaluate QPP by measuring linear correlation by Pearson's ρ as well as ranked-based correlation through Kendall's τ correlation coefficients between the actual and predicted performance of a query set. As for target IR metrics, we consider the two primary official IR metrics used in TREC DL 19–22 [22–25], RR@10 (precision-oriented), and nDCG@10 (considering recall); recent QPP studies [6, 45, 63] consider either or both of these metrics as their target metrics. Following [32], we use relevance scale ≥ 2 as positive to compute actual binary IR measures (e.g., RR). When calculating correlation for nDCG@10, the actual values of nDCG@10 are calculated by human-labeled and multi-graded relevance judgments, while the nDCG@10 values predicted by QPP-GenRE are based on its generated binary judgments.

5.6 Implementation Details

For all unsupervised QPP baselines, we tune the hyper-parameters for predicting the ranking quality of a ranker (either BM25 or ANCE) on TREC-DL 19 (TREC-DL 21) based on Pearson's ρ correlation for predicting the ranking quality of the same ranker on TREC-DL 20 (TREC-DL 22), and vice versa. We select the cut-off value k for Clarity, NQC, WIG, SMV, and so on from $\{5, 10, 15, 20, 25, 50, 100, 300, 500, 1{,}000\}$. n($\sigma_{x\%}$) has a hyper-parameter x, which we choose from the set $\{0.25, 0.4, 0.5, 0.6, 0.75, 0.9\}$.

To predict the performance of a certain ranker (any of BM25, ANCE, or TAS-B), we train all supervised QPP baselines based on the ranked list returned by the target ranker. To predict a certain IR evaluation measure, regression-based methods [7, 55, 63] are trained to learn to output the target evaluation measure during training. However, our preliminary result shows that training supervised QPP baselines, especially for regression-based supervised methods [7, 55, 63], on the training set of MS MARCO V1 leads to inferior QPP quality for predicting the performance of the neural rankers (ANCE and TAS-B). We hypothesize that this is because they were originally trained on the training set of MS MARCO V1 [57, 134], and so the ranked list returned by them on the training set of MS MARCO V1 would have higher quality than the ranked list returned by them on the evaluation sets; therefore, supervised QPP methods that share the same training set as the neural rankers tend to predict inflated performance on the evaluation sets, leading to degraded QPP quality. To solve the issue and ensure the consistency of the article, we train all supervised QPP methods (including QPP-GenRE) on the development set of MS MARCO V1 (6,980 queries) for predicting the performance of BM25, ANCE, or TAS-B. We train all supervised QPP methods for 5 epochs and pick the best checkpoint for predicting the performance of a ranker on TREC-DL 19 (TREC-DL 21) based on Pearson's ρ correlation for predicting the performance of the same ranker on TREC-DL 20 (TREC-DL 22) and vice versa. All supervised QPP baselines use bert-base-uncased,[4] a constant learning rate (0.00002), and the Adam optimizer [65].

For QPP-LLM, we prompt LLaMA-7B with the top-k retrieved items, where k is set to 10. For QPP-LLM (few-shot), we randomly sample demonstration examples from the development set of MS MARCO V1; our preliminary experiments show that sampling two demonstrations works best. For QPP-LLM (fine-tuned), we fine-tune LLaMA-7B using PEFT as QPP-GenRE fine-tunes LLMs.

[4] https://github.com/huggingface/transformers.

Table 1. Correlation Coefficients (Pearson's ρ and Kendall's τ) between Actual Retrieval Quality, in Terms of RR@10, of BM25, and Performance Predicted by QPP-GenRE/Baselines, on TREC-DL 19–22

	Ranker: BM25							
	TREC-DL 19		TREC-DL 20		TREC-DL 21		TREC-DL 22	
QPP method	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.135	0.028	0.050	0.021	0.183	0.161	0.253[a]	0.099
WIG	0.113	0.164	0.286[a]	0.218[a]	0.237	0.206[a]	0.029	0.082
NQC	0.194	0.117	0.152	0.191	0.227	0.195	0.223	0.048
σ_{max}	0.195	0.164	0.200	0.211[a]	0.278[a]	0.174	0.038	0.048
n($\sigma_{x\%}$)	0.144	0.181	0.187	0.123	0.127	0.140	0.169	0.113
SMV	0.141	0.097	0.126	0.193	0.240	0.189	0.227[a]	0.094
UEF(NQC)	0.235	0.256[a]	0.270[a]	0.211[a]	0.231	0.111	0.216	0.065
RLS(NQC)	0.272	0.122	0.290[a]	0.193	0.234	0.195	0.224	0.095
QPP-PRP	0.292	0.189	0.163	0.184	−0.080	−0.017	0.122	0.091
NQA-QPP	0.181	0.122	0.062	0.069	0.161	0.163	0.224	0.177[a]
BERTQPP	0.281	0.136	0.237	0.155	0.206	0.134	0.148	0.122
qppBERT-PL	0.145	0.138	0.166	0.152	0.339[a]	0.244[a]	0.131	0.206[a]
M-QPPF	0.317[a]	0.208	0.335[a]	0.273[a]	0.282[a]	0.209[a]	0.161	0.187[a]
QPP-LLM (few-shot)	0.008	0.003	−0.081	−0.129	−0.053	−0.053	−0.241	−0.155
QPP-LLM (fine-tuned)	0.171	0.158	0.228	0.206	0.030	0.099	−0.038	0.009
QPP-GenRE ($n = 10$)	**0.538**[a,b]	**0.486**[a,b]	**0.560**[a,b]	**0.475**[a,b]	**0.524**[a,b]	**0.435**[a,b]	**0.350**[a,b]	**0.262**[a,b]

The best value in each column is marked in bold. n denotes QPP-GenRE's judgment depth in a ranked list.
[a]Statistically significant correlation coefficients (p-value < 0.05).
[b]The statistically significant improvement of QPP-GenRE compared to all the baselines (paired t-test; p-value < 0.001 with Bonferroni correction for multiple testing).

We equip QPP-GenRE with an LLM for judging relevance. We use a recent PEFT method, 4-bit QLoRA [33], to fine-tune an LLM. To maintain a comparable setup with the baselines, we fine-tune an LLM for 5 epochs on the development set of MS MARCO V1. Note that we use LLaMA-7B for BM25 and ANCE, and Mistral-7B-Instruct-v0.3 for TAS-B. The training of judging relevance needs positive and negative items per query. For positive items, we use the items annotated as relevant in *qrels* per query; we randomly sample one negative item from the ranked list (1,000 items) returned by BM25 per query. There are 6,980 queries in our training set. Each query may have multiple relevant items annotated in the *qrels*, and has one negative item we sampled. As a result, we have 7,437 positive training examples and 6,980 negative training examples. All experiments are conducted on an NVIDIA A100 GPU (40 GB).

One might argue why we choose QLoRA fine-tuning instead of distilling an oracle model into a smaller model [119]. The decision is based on the following three reasons. First, model distillation requires an existing oracle model, such as GPT-4, for relevance prediction. However, this work focuses on exclusively using open source LLMs, avoiding using powerful commercial models like GPT-4 to ensure reproducibility and deterministic outputs. Second, one might wonder why we did not distill larger open source LLMs into smaller models. As demonstrated in Figure 5, a 1-billion-parameter Llama model fine-tuned using QLoRA on human-labeled relevance judgments outperforms a 70-billion-parameter Llama model using few-shot prompting. Therefore, if we choose to distill the 70-billion-parameter model into a smaller model, the performance of the distilled model would be inferior to that achieved by the 1-billion-parameter model fine-tuned via QLoRA, because the performance of distilled model is inherently limited by the larger model's capabilities.

Table 2. Correlation Coefficients (Pearson's ρ and Kendall's τ) between Actual Retrieval Quality, in Terms of RR@10, of ANCE/TAS-B, and Performance Predicted by QPP-GenRE/Baselines, on TREC-DL 19 and 20

	Ranker: ANCE				Ranker: TAS-B			
	TREC-DL 19		TREC-DL 20		TREC-DL 19		TREC-DL 20	
QPP method	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	−0.078	−0.012	−0.074	−0.048	−0.212	−0.148	0.148	0.133
WIG	0.313[a]	0.228	0.059	0.048	−0.066	−0.125	0.024	0.020
NQC	0.350[a]	0.200	0.145	0.112	0.248	0.213	0.260	0.194
σ_{max}	0.384[a]	0.287[a]	0.171	0.118	0.015	0.021	0.312[a]	0.245[a]
n($\sigma_{x\%}$)	0.200	0.176	−0.008	0.022	−0.030	−0.079	0.080	0.086
SMV	0.352[a]	0.256[a]	0.182	0.161	0.249	0.205	0.263	0.198
UEF(NQC)	0.340[a]	0.260[a]	0.131	0.108	0.260	0.228	0.281[a]	0.213
RLS(NQC)	0.359[a]	0.273[a]	0.178	0.139	0.257	0.217	0.283[a]	0.217[a]
QPP-PRP	0.259	0.246	0.100	−0.008	0.155	0.113	0.203	0.116
Dense-QPP	0.452[a]	0.280[a]	0.209	0.139	0.251	0.213	0.146	0.012
NQA-QPP	−0.026	−0.009	−0.059	−0.080	0.172	0.144	−0.058	−0.075
BERTQPP	0.330[a]	0.214	0.046	−0.012	0.202	0.194	0.077	0.037
qppBERT-PL	0.092	0.025	−0.224	−0.218	0.276	0.269	0.004	−0.002
M-QPPF	0.292	0.200	0.068	0.038	0.277	0.236	0.103	0.022
QPP-LLM (few-shot)	−0.008	0.005	−0.226	−0.207	−0.080	0.002	0.054	−0.024
QPP-LLM (fine-tuned)	−0.073	0.011	−0.022	0.069	0.155	0.113	0.043	−0.020
QPP-GenRE (n = 10)	**0.567**[a,b]	**0.440**[a,b]	**0.293**[a,b]	**0.257**[a,b]	**0.538**[a,b]	**0.481**[a,b]	**0.356**[a,b]	**0.289**[a,b]

The best value in each column is marked in bold. n denotes QPP-GenRE's judgment depth in a ranked list.
[a]Statistically significant correlation coefficients (p-value < 0.05).
[b]The statistically significant improvement of QPP-GenRE compared to all the baselines (paired t-test; p-value < 0.001 with Bonferroni correction for multiple testing).

Third, we have a large amount of human-labeled relevance judgments available. Directly using these labels to fine-tune LLMs via QLoRA allows us to make the most efficient use of this data.

6. Results

6.1 Predicting a Precision-Oriented IR Measure

To answer RQ1, we compare QPP-GenRE and all baselines in predicting the performance of BM25, ANCE, and TAS-B w.r.t. a widely used precision-oriented metric, RR@10; see Tables 1 and 2. We have three main observations.

First, our proposed method, QPP-GenRE, outperforms all baselines in terms of both correlation coefficients on all datasets when predicting the performance of all rankers. In particular, we observe that QPP-GenRE outperforms QPP-PRP [115], which is a recently proposed baseline by 84% (0.292 vs. 0.538) in terms of Pearson's ρ when predicting RR@10 for BM25 on TREC-DL 19.

Second, QPP-LLM (few-shot) gets the worst result compared to other approaches. While QPP-LLM (fine-tuning) performs slightly better than QPP-LLM (few-shot), its performance is still limited in most cases. This indicates that it is ineffective for an LLM to model QPP in a straightforward way of directly predicting a score.

Third, there is no clear winner among the baselines, and the performance of baselines shows a bigger variance than QPP-GenRE across different datasets and rankers, e.g., the unsupervised method WIG achieves a good result among baselines for assessing BM25 on TREC-DL 20, while it gets nearly zero correlation coefficients on TREC-DL 22 when assessing BM25. Conversely,

Table 3. Correlation Coefficients (Pearson's ρ and Kendall's τ) between Actual Retrieval Quality, in Terms of nDCG@10, of BM25, and Performance Predicted by QPP-GenRE/Baselines, on TREC-DL 19–22

	Ranker: BM25							
	TREC-DL 19		TREC-DL 20		TREC-DL 21		TREC-DL 22	
QPP method	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.091	0.056	0.358[a]	0.250[a]	0.137	0.078	0.202	0.090
WIG	0.520[a]	0.331[a]	0.615[a]	0.423[a]	0.311[a]	0.281[a]	0.350[a]	0.249[a]
NQC	0.468[a]	0.300[a]	0.508[a]	0.401[a]	0.134	0.221[a]	0.360[a]	0.156[a]
σ_{max}	0.478[a]	0.327[a]	0.529[a]	0.440[a]	0.298[a]	0.258[a]	0.142[a]	0.196[a]
n($\sigma_{x\%}$)	0.532[a]	0.311[a]	0.622[a]	0.443[a]	0.328[a]	0.234[a]	0.336[a]	0.228[a]
SMV	0.376[a]	0.271[a]	0.463[a]	0.383[a]	0.327[a]	0.236[a]	0.338[a]	0.155[a]
UEF(NQC)	0.499[a]	0.322[a]	0.517[a]	0.356[a]	0.153	0.232[a]	0.311[a]	0.145
RLS(NQC)	0.469[a]	0.169	0.522[a]	0.376[a]	0.272[a]	0.223[a]	0.337[a]	0.157[a]
QPP-PRP	0.321	0.181	0.189	0.157	0.027	0.004	0.077	0.012
NQA-QPP	0.210	0.147	0.244	0.210[a]	0.286[a]	0.201[a]	0.312[a]	0.194[a]
BERTQPP	0.458[a]	0.207	0.426[a]	0.300[a]	0.351[a]	0.223[a]	0.369[a]	0.229[a]
qppBERT-PL	0.171	0.175	0.410[a]	0.279[a]	0.277[a]	0.182	0.300[a]	0.242[a]
M-QPPF	0.404[a]	0.254[a]	0.435[a]	0.297[a]	0.265	0.226[a]	0.345[a]	0.204[a]
QPP-LLM (few-shot)	−0.024	−0.031	0.167	0.138	0.238	0.201	−0.073	−0.077
QPP-LLM (fine-tuned)	0.313[a]	0.215	0.309[a]	0.254[a]	0.264	0.198	−0.075	−0.009
QPP-GenRE (n = 200)	**0.724**[a,b]	0.474[a,b]	**0.638**[a,b]	**0.469**[a,b]	0.546[a,b]	0.435[a,b]	**0.388**[a]	**0.251**[a]
QPP-GenRE (n = 10)	0.605[a]	**0.482**[a]	0.490[a]	0.323[a]	0.462[a]	0.350[a]	0.316[a]	0.245[a]
QPP-GenRE (n = 100)	0.712[a]	0.472[a]	0.609[a]	0.457[a]	0.545[a]	0.427[a]	0.332[a]	0.246[a]
QPP-GenRE (n = 1,000)	0.715[a]	0.477[a]	0.627[a]	0.459[a]	**0.547**[a]	**0.436**[a]	**0.388**[a]	**0.251**[a]

n denotes QPP-GenRE's judgment depth in a ranked list. The best value in each column is marked in bold.
[a]Statistically significant correlation coefficients (p-value < 0.05).
[b]The statistically significant improvement of QPP-GenRE (n = 200) compared to all the baselines (paired t-test; p-value < 0.001 with Bonferroni correction for multiple testing).

QPP-GenRE consistently achieves the best performance across datasets and rankers, thus showing robust performance.

6.2 Predicting an IR Measure Considering Recall

To answer *RQ2*, Tables 3 and 4 list the performance of QPP-GenRE along with all the baselines on assessing BM25, ANCE, and TAS-B in terms of nDCG@10. For QPP-GenRE, we universally set the judging depth n to 200 for all evaluation sets. The result reveals that by judging only 200 items per query, we can achieve state-of-the-art QPP quality in terms of nDCG@10 for all rankers on all evaluation sets; we will investigate the impact of judging depth on QPP-GenRE's performance in the next section. Also, QPP-LLM (few-shot) and QPP-LLM (fine-tuning) are among the worst-performing baselines, showing that the LLMs struggle to generate numerical scores. Different from the results for *RQ1*, most QPP methods tend to perform better when predicting nDCG@10 than RR@10; this observation indicates that predicting RR@10 is a more challenging task.

7 Analysis

7.1 Judging Depth Analysis

RQ3 examines how varying the number of top-ranked documents submitted for relevance judgments impacts QPP quality. To answer *RQ3*, as detailed in Section 4.3.2, for predicting IDCG, we devise

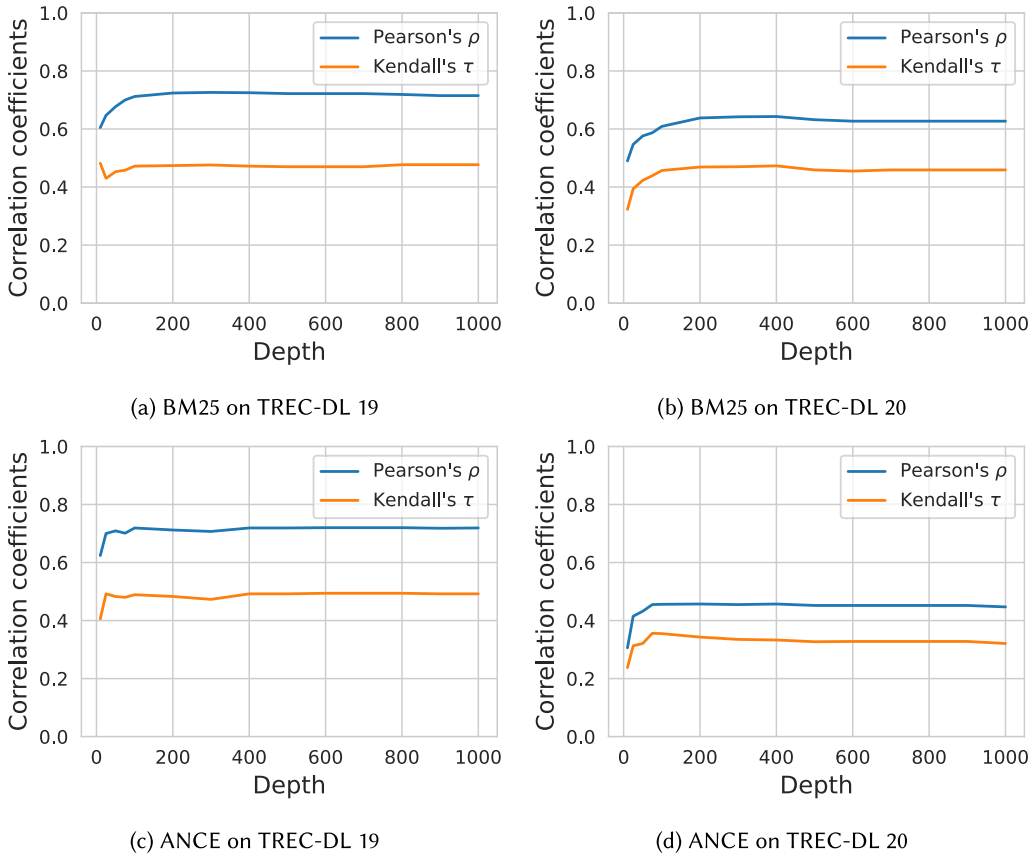
Table 4. Correlation Coefficients (Pearson's ρ and Kendall's τ) between Actual Retrieval Quality, in Terms of nDCG@10, of ANCE/TAS-B, and Performance Predicted by QPP-GenRE/Baselines, on TREC-DL 19 and 20

	Ranker: ANCE				Ranker: TAS-B			
	TREC-DL 19		TREC-DL 20		TREC-DL 19		TREC-DL 20	
QPP method	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	−0.088	−0.062	−0.091	−0.045	0.153	0.049	0.162	0.087
WIG	0.515[a]	0.368[a]	0.218	0.150	0.228	0.146	0.227	0.169
NQC	0.548[a]	0.372[a]	0.411[a]	0.290[a]	0.330[a]	0.233[a]	0.406[a]	0.264[a]
σ_{max}	0.455[a]	0.339[a]	0.403[a]	0.288[a]	0.220	0.126	0.428[a]	0.284[a]
n($\sigma_{x\%}$)	0.388[a]	0.315[a]	0.103	0.075	−0.008	−0.031	0.002	−0.020
SMV	0.496[a]	0.359	0.380[a]	0.283[a]	0.349[a]	0.253[a]	0.425[a]	0.285[a]
UEF(NQC)	0.548[a]	0.372[a]	0.413[a]	0.290[a]	0.321[a]	0.246[a]	0.425[a]	0.271[a]
RLS(NQC)	0.466[a]	0.346[a]	0.333[a]	0.271[a]	0.314[a]	0.246[a]	0.404[a]	0.272[a]
QPP-PRP	0.129	0.049	0.216	0.121	0.220	0.126	0.267	0.237[a]
Dense-QPP	0.565[a]	0.389[a]	0.419[a]	0.318[a]	0.429[a]	0.244[a]	0.126	0.012
NQA-QPP	0.089	−0.038	0.186	0.113	−0.020	0.060	0.031	0.024
BERTQPP	0.222	0.117	0.137	0.089	0.043	0.027	0.178	0.086
qppBERT-PL	0.116	0.098	−0.119	−0.046	0.304[a]	0.187	0.057	0.057
M-QPPF	0.287	0.160	0.225	0.177	0.163	0.051	0.304[a]	0.171
QPP-LLM (few-shot)	0.136	0.120	−0.130	−0.094	−0.020	0.060	0.108	0.048
QPP-LLM (fine-tuned)	0.203	0.117	0.081	0.097	0.262	0.195	0.162	0.111
QPP-GenRE (n = 200)	0.712[a,b]	0.483[a,b]	**0.457**[a,b]	0.343[a,b]	0.501[a,b]	0.346[a,b]	0.449[a]	0.315[a]
QPP-GenRE (n = 10)	0.624[a]	0.406[a]	0.306[a]	0.238[a]	0.490[a]	0.309[a]	0.421[a]	0.290[a]
QPP-GenRE (n = 100)	**0.719**[a]	0.489[a]	0.456[a]	**0.355**[a]	0.501[a]	0.336[a]	**0.450**[a]	**0.317**[a]
QPP-GenRE (n = 1,000)	**0.719**[a]	**0.492**[a]	0.447[a]	0.321[a]	**0.505**[a]	**0.348**[a]	0.449[a]	0.315[a]

n denotes QPP-GenRE's judgment depth in a ranked list. The best value in each column is marked in bold.
[a]Statistically significant correlation coefficients (p-value < 0.05).
[b]The statistically significant improvement of QPP-GenRE (n = 200) compared to all the baselines (paired t-test; p-value < 0.001 with Bonferroni correction for multiple testing).

an approximation strategy and use the items in the top n ranks of the ranked list L that are predicted as relevant by QPP-GenRE to approximate all the relevant items for a query in the corpus. To investigate the impact of the value of n on the quality of the prediction, we investigate the relationship between the QPP quality of predicting nDCG@10 and the judgment depth to answer the following question: *What depth of relevance judgment n do we need to consider to get a satisfactory performance for predicting nDCG@10?* In Figure 4, we plot the correlation coefficients between actual nDCG@10 values and nDCG@10 values predicted by QPP-GenRE for different judging depths in {10, 25, 50, 75, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1,000} on TREC-DL 19 and 20. We also show exact QPP results with depths at 10, 100, and 1,000 in Tables 3 and 4.

Tables 3 and 4 reveal that, by judging only 10 items in the ranked list, we can already outperform all the baselines and achieve state-of-the-art QPP quality on half of the evaluation sets we used, e.g., assessing BM25 on TREC-DL 19/21, ANCE on TREC-DL 19, and TAS-B on TREC-DL 19. While judging deeper in the ranked list is essential for predicting recall-oriented measures, satisfactory QPP quality is still attainable with a relatively shallow depth. Moreover, Figure 4 illustrates that judging the top 200 items in a ranked list already reaches the saturation point for assessing BM25, i.e., there is no significant improvement by judging a higher number of items, while judging less than 100 top items reaches the saturation point for ANCE. We speculate that this is because ANCE



(a) BM25 on TREC-DL 19

(b) BM25 on TREC-DL 20

(c) ANCE on TREC-DL 19

(d) ANCE on TREC-DL 20

Fig. 4. Relationship between the QPP effectiveness of predicting nDCG@10 and the judging depth for a ranked list.

has better retrieval quality than BM25, and more relevant items would appear earlier in the ranked list of ANCE than BM25; therefore, a shallower judging depth suffices to approximate all relevant items in the corpus. This emphasizes the need to consider retrieval quality when determining the optimal judgment depth for various rankers.

7.2 Impact of Fine-Tuning and the Choice of LLMs

To answer *RQ4*, we analyze the impact of fine-tuning and the choice of LLMs on the quality of generated relevance judgments and QPP. We evaluate two widely used families of LLMs, Llama and Mistral, spanning from 1B to 70B under two settings: (i) trained with PEFT on human relevance labels (following the same fine-tuning setup as in Section 5.6), and (ii) few-shot prompted (ICL).[5] For the Llama family, besides LLaMA-7B [128], our evaluation includes Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct, Llama-3-8B, Llama-3-8B-Instruct, and Llama-3-70B-Instruct. For the Mistral family, we focus on Mistral-7B-Instruct-v0.3 and Mistral-22B-Instruct (a.k.a. Mistral-Small-Instruct-2409).

[5]We randomly sample human-labeled demonstration examples from the same set used for fine-tuning LLMs; each example is a triplet (query, passage, relevant/irrelevant); our experiments show that two examples work best, one with relevant passages and one with irrelevant passages.

Table 5. Relevance Judgment Agreement (Cohen's κ) between TREC Assessors and Each LLM, and Pearson's ρ Correlation Coefficients between BM25's Actual nDCG@10 Values and Those Predicted by QPP-GenRE Integrated with Each LLM on TREC-DL 19–22

	TREC-DL 19		TREC-DL 20		TREC-DL 21		TREC-DL 22	
LLM	κ	P-ρ	κ	P-ρ	κ	P-ρ	κ	P-ρ
GPT-3.5 (text-davinci-003) [40]	-	-	-	-	0.260	-	-	-
Llama-3.2-1B-Instruct (few-shot)	0.013	0.152	0.029	0.099	0.009	0.249	0.079	0.087
Llama-3.2-3B-Instruct (few-shot)	0.186	0.293	0.114	0.020	0.165	0.289	0.055	0.443
Mistral-7B-Instruct-v0.3 (few-shot)	0.224	0.271	0.174	0.499	0.245	0.414	0.042	0.243
LLaMA-7B (few-shot)	−0.001	−0.062	−0.003	0.087	0.003	−0.002	−0.010	0.214
Llama-3-8B (few-shot)	0.018	0.042	0.027	0.087	0.021	0.180	−0.035	0.087
Llama-3-8B-Instruct (few-shot)	0.315	0.510	0.227	0.372	0.238	0.462	0.049	0.388
Mistral-22B-Instruct (few-shot)	0.281	0.412	0.238	0.535	0.276	0.528	0.083	0.473
Llama-3-70B-Instruct (few-shot)	0.321	0.526	0.245	0.557	0.279	0.545	**0.086**	0.483
Llama-3.2-1B-Instruct (fine-tuned)	0.351	0.610	0.211	0.596	0.197	0.570	0.042	0.428
Llama-3.2-3B-Instruct (fine-tuned)	0.383	0.710	0.273	0.722	0.306	0.608	0.042	0.511
Mistral-7B-Instruct-v0.3 (fine-tuned)	0.403	0.734	0.328	0.720	0.373	0.592	0.076	0.411
LLaMA-7B (fine-tuned)	0.258	**0.715**	0.238	0.627	0.333	0.547	0.038	0.388
Llama-3-8B (fine-tuned)	0.381	0.544	**0.342**	0.681	0.347	0.612	0.082	0.568
Llama-3-8B-Instruct (fine-tuned)	0.397	0.647	0.316	**0.743**	**0.418**	**0.699**	0.066	**0.573**
Mistral-22B-Instruct (fine-tuned)	**0.407**	0.682	0.276	0.640	0.310	0.591	0.071	0.462

The best value in each column is marked in bold. We do not fine-tune Llama-3-70B-Instruct due to budget constraints.

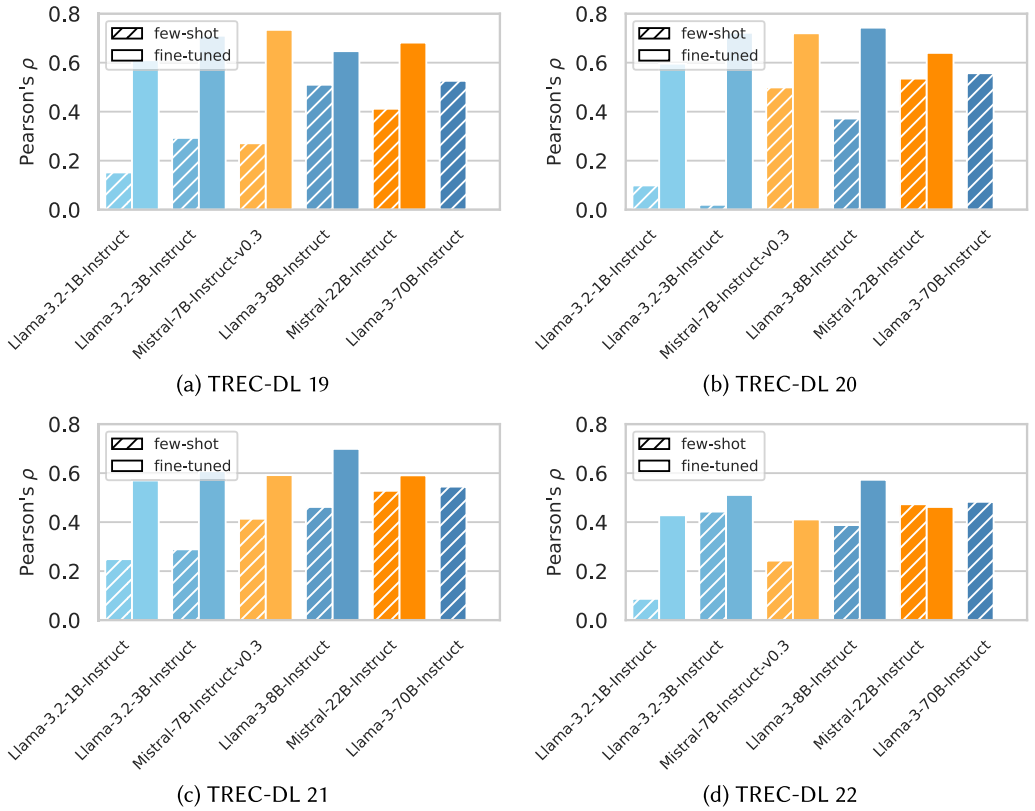
We do not report the results for a zero-shot setting because our preliminary experiments show that prompting these LLMs in a zero-shot way yields pretty poor performance. Note that we do not fine-tune Llama-3-70B-Instruct due to budget constraints.

To evaluate the performance of judging relevance, we compute Cohen's κ metric to measure the agreement between relevance judgments made by the TREC assessors (i.e., relevance judgments in the *qrels*) and relevance judgments automatically generated by a fine-tuned or few-shot LLM, on TREC-DL 19–22. Faggioli et al. [40] reported the relevance judgment agreement in terms of Cohen's κ between TREC assessors and GPT-3.5 (text-davinci-003) on TREC-DL 21; we also consider their Cohen's κ value for comparison. To evaluate QPP quality, we compute the Pearson's ρ correlation coefficients between BM25's actual nDCG@10 values and those predicted by QPP-GenRE using relevance judgments generated by an LLM, on TREC-DL 19–22.[6] The judging depth is set to 1,000 in a ranked list. We show the results in Table 5 and Figure 5.

We have three observations. First, fine-tuning generally markedly improves the quality of relevance judgment generation and QPP, particularly for LLMs with sizes ranging from 1 billion to 8 billion parameters. Specifically, almost all of fine-tuned LLMs exhibit improved relevance judgment agreement with the TREC assessors on TREC-DL 19–22. After fine-tuning, LLaMA-7B and Llama-3-8B achieve "fair" agreement with the TREC assessors on TREC-DL 19, 20, and 21,[7] and Llama-3-8B-Instruct (fine-tuned) even achieves "moderate" agreement on TREC-DL 21 (a

[6]We do not report the Pearson's ρ correlation for GPT-3.5 (text-davinci-003) because the relevance judgments generated by Faggioli et al. [40] are not available to us.

[7]Note that unlike the qrels files for TREC-DL 19, 20, and 21 which are fully manually annotated, the qrels file for TREC-DL 22 is constructed by first detecting near-duplicate items and manually judging only one representative item from each near-duplicate cluster for a given query [24]; this difference may result in variation in Cohen's κ values of LLMs across TREC-DL 19, 20, 21, and TREC-DL 22.



(a) TREC-DL 19 (b) TREC-DL 20

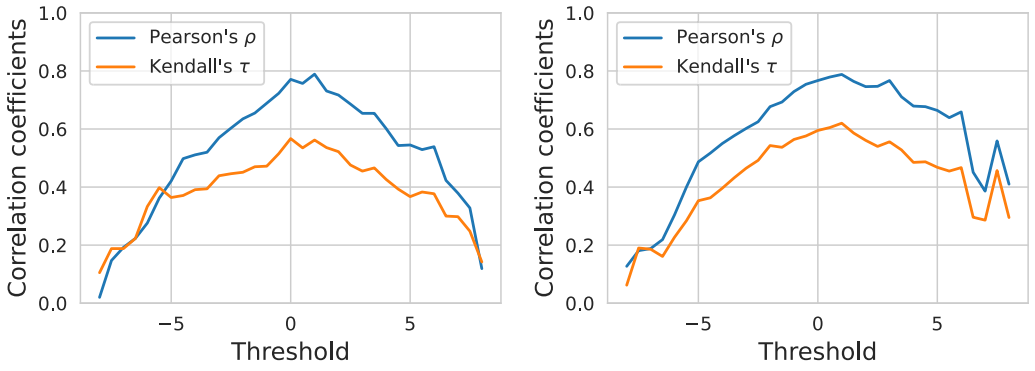
(c) TREC-DL 21 (d) TREC-DL 22

Fig. 5. Pearson's ρ correlation coefficients between BM25's actual nDCG@10 values and those predicted by QPP-GenRE integrated with various LLMs with different sizes, under both few-shot and fine-tuned settings, on TREC-DL 19–22. From left to right along the x-axis, as the size of the LLMs increases, the inference efficiency correspondingly decreases. To aid visual comparison, LLMs from the same family share a consistent color scheme: Llama models are shown in blue, and Mistral models in orange. Note that, for simplicity, we only retain the results of LLMs with "instruct" versions. We do not fine-tune Llama-3-70B-Instruct due to budget constraints.

Cohen's κ value of 0.418). All fine-tuned LLMs (except for Llama-3.2-1B-Instruct) exhibit a higher Cohen's κ value than the commercial LLM, GPT-3.5 (text-davinci-003). All fine-tuned LLMs (except for Mistral-22B-Instruct on TREC-DL 22) surpass their corresponding few-shot counterpart on all datasets in terms of Pearson's ρ. This reveals that fine-tuning is an effective way to improve the quality of LLMs in generating relevance judgments, which finally translates to better QPP quality.

Second, larger LLMs with over 22 billion parameters demonstrate significantly greater effectiveness than their smaller counterparts in the few-shot setting. Specifically, in this setting, Llama-3-70B-Instruct achieves the best overall performance. Mistral-22B-Instruct consistently outperforms Mistral-7B-Instruct-v0.3 across all datasets. However, a fine-tuned Llama model with only 3 billion parameters markedly outperforms both of these larger few-shot LLMs across all datasets.

Third, Instruction-tuned LLMs generally perform better than their standard counterparts. Llama-3-8B-Instruct further enhances relevance judgment generation and QPP quality over both Llama-3-8B and LLaMA-7B across most cases. Notably, Llama-3-8B-Instruct (few-shot) even performs better



(a) BM25 on TREC-DL 19 (b) BM25 on TREC-DL 20

Fig. 6. QPP quality of QPP-GenRE integrated with RankLLaMA [76] in predicting nDCG@10 values for BM25, w.r.t. threshold values ranged from −8 to 8, on TREC-DL 19 and 20. An item is predicted as "relevant" if its re-ranking score meets or exceeds a given threshold value.

than or equally as well as LLaMA-7B (fine-tuned) on TREC-DL 19 and 22. This finding implies that with a more effective LLM QPP-GenRE has the potential to achieve improved QPP performance.

The above observations provide insights into the minimum requirements needed to achieve reliable QPP quality. Our findings show that a fine-tuned 3B model (Llama-3.2-3B-Instruct) offers the best tradeoff between QPP quality and computational overhead: It not only markedly outperforms few-shot 70B LLMs but also delivers QPP quality comparable to that of fine-tuned 7/8B LLMs.

7.3 Integrating QPP-GenRE with an LLM-Based Re-Ranker

To show QPP-GenRE's compatibility with other types of relevance prediction methods instead of directly asking an LLM to explicitly generate explicit relevance judgments, we adapt a state-of-the-art pointwise LLM-based re-ranker, RankLLaMA [76], into a relevance judgment generator, and then integrate QPP-GenRE with the adapted RankLLaMA. Specifically, we translate a re-ranking score into a relevance judgment by applying a threshold: An item is deemed as "relevant" if its re-ranking score meets or exceeds a given threshold value. We analyze Pearson's ρ and Kendall's τ correlation coefficients between BM25's actual nDCG@10 values and those predicted by QPP-GenRE integrated with RankLLaMA w.r.t. different threshold values on TREC-DL 19 and 20. We employ RankLLaMA (7B) from Tevatron.[8] RankLLaMA's re-ranking scores for BM25 range from −12.93 to 89.90 for TREC-DL 19 and from −14.38 to 8.82 for TREC-DL 20. Thresholds are set at intervals of 0.5. The judging depth is set to 1,000 in a ranked list.

We report the results in Figure 6. We find that RankLLaMA achieves the highest QPP quality on both datasets when the threshold is 1. At this particular threshold, RankLLaMA achieves high Pearson's ρ values of 0.789 and 0.788 on TREC-DL 19 and 20, respectively. These values exceed those of fine-tuned LLaMA-7B, which achieves Pearson's ρ values of 0.715 and 0.627 on TREC-DL 19 and 20, respectively, as well as Llama-3-8B-Instruct, which achieves Pearson's ρ values of 0.647 and 0.743 on TREC-DL 19 and 20, respectively (see Figure 5).[9] This means that a state-of-the-art pointwise LLM-based re-ranker can be adapted into an effective relevance judgment generator.

[8] https://github.com/texttron/tevatron/tree/main/examples/rankllama.

[9] Note that the comparison is not fair because (i) LLMs and all other supervised QPP methods used in this article are trained on the development set of MS MARCO V1, while RankLLaMA [76] was trained on the training set of MS MARCO V1, which is much larger. (ii) We employ the official version of MS MARCO V1, while RankLLaMA [76] uses the Tevatron version of

Table 6. Correlation Coefficients (Pearson's ρ and Kendall's τ) between Actual Retrieval Quality, in Terms of nDCG@3, of ConvDR [137], and Its Performance Predicted by QPP-GenRE/Baselines, on CAsT-19 and 20

	Ranker: ConvDR							
	T5-generated query rewrites				Human-written query rewrites			
	CAsT-19		CAsT-20		CAsT-19		CAsT-20	
QPP method	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.257[a]	0.176[a]	0.126	0.088	0.257[a]	0.176[a]	0.126	0.088
WIG	0.387[a]	0.274[a]	0.377[a]	0.277[a]	0.412[a]	0.285[a]	0.384[a]	0.264[a]
NQC	0.431[a]	0.307[a]	0.339[a]	0.261[a]	0.431[a]	0.307[a]	0.339[a]	0.261[a]
σ_{max}	0.378[a]	0.267[a]	0.282[a]	0.219[a]	0.378[a]	0.267[a]	0.282[a]	0.219[a]
n($\sigma_{x\%}$)	0.187[a]	0.175[a]	0.199[a]	0.168[a]	0.216[a]	0.196[a]	0.201[a]	0.156[a]
SMV	0.386[a]	0.285[a]	0.275[a]	0.216[a]	0.386[a]	0.285[a]	0.275[a]	0.216[a]
UEF(NQC)	0.435[a]	0.312[a]	0.343[a]	0.265[a]	0.427[a]	0.310[a]	0.341[a]	0.263[a]
RLS(NQC)	0.429[a]	0.311[a]	0.337[a]	0.267[a]	0.413[a]	0.308[a]	0.342[a]	0.259[a]
QPP-PRP	0.350[a]	0.270[a]	0.280[a]	0.210[a]	0.345[a]	0.265[a]	0.275[a]	0.205[a]
NQA-QPP	0.175	0.115	0.082	0.075	0.142	0.091	0.065	0.058
BERTQPP	0.243[a]	0.170[a]	0.236[a]	0.185[a]	0.256[a]	0.172[a]	0.262[a]	0.209[a]
qppBERT-PL	0.203[a]	0.169[a]	0.181[a]	0.165[a]	0.105	0.090	0.166[a]	0.161[a]
M-QPPF	0.242[a]	0.174[a]	0.285[a]	0.219[a]	0.262[a]	0.190[a]	0.313[a]	0.254[a]
QPP-GenRE (n = 200)	**0.623**[a,b]	**0.505**[a,b]	0.484[a,b]	0.395[a,b]	**0.645**[a,b]	**0.529**[a,b]	0.678[a,b]	0.551[a,b]
QPP-GenRE (n = 10)	0.617[a]	0.504[a]	0.471[a]	0.388[a]	0.619[a]	0.506[a]	0.659	0.534
QPP-GenRE (n = 100)	**0.623**[a]	**0.505**[a]	0.485[a]	0.396[a]	0.644[a]	**0.529**[a]	0.675[a]	0.547[a]
QPP-GenRE (n = 1,000)	**0.623**[a]	**0.505**[a]	**0.487**[a]	**0.398**[a]	**0.645**[a]	**0.529**[a]	**0.684**[a]	**0.556**[a]

Following Meng et al. [86], for QPP methods requiring queries as input, we feed them with either T5-generated or human-written query rewrites. The best value in each column is marked in bold. n denotes QPP-GenRE's judgment depth in a ranked list.

[a]Statistically significant correlation coefficients (p-value < 0.05).

[b]The statistically significant improvement of QPP-GenRE ($n = 200$) compared to all the baselines (paired t-test; p-value < 0.001 with Bonferroni correction for multiple testing).

The high QPP quality achieved by RankLLaMA demonstrates QPP-GenRE's compatibility with other types of relevance prediction methods besides directly using LLMs as relevance judgment generators (i.e., asking an LLM to explicitly generate explicit relevance judgments).

However, compared to directly regarding an LLM as a relevance judgment generator, adapting an LLM-based re-ranker into a relevance judgment generator requires tuning an appropriate threshold. As demonstrated, re-ranking scores are not normalized and their ranges vary across datasets. Directly using a re-ranker as a relevance judgment generator can cause issues in real-world scenarios. Extra calibration work for re-ranking scores might be necessary.

7.4 Generalization to Conversational Search

To assess the generalizability of QPP-GenRE to new domains, we apply it to the conversational search scenario [89, 91–95] in a zero-shot manner. Specifically, we evaluate QPP-GenRE and other baselines on predicting the performance of ConvDR [137], a widely used conversational dense

MS MARCO V1, where passages are enriched with document titles; Lassance and Clinchant [67] reveal that incorporating titles leads to enhanced ranking performance.

retriever. Given the findings in Section 7.2, which show that the fine-tuned Llama-3.2-3B-Instruct model achieves high relevance prediction quality at low inference cost, we equip QPP-GenRE with this model fine-tuned on MS MARCO V1 for relevance prediction. For all supervised QPP baselines, we directly use their checkpoints trained on MS MARCO V1 for assessing ANCE (see Section 6.2). Because a user query in a conversation depends on the conversational context, i.e., a query may contain omissions, coreferences, or ambiguities, it is hard for existing QPP methods to capture users' information need from such context-dependent queries. Therefore, we follow Meng et al. [86] to provide QPP methods (including QPP-GenRE) with self-contained query rewrites as input. These rewrites are either generated by the T5 query generator[10] or written by humans. Table 6 presents the performance of QPP-GenRE along with all the baselines on assessing ConvDR [137] in terms of nDCG@3 on the CAsT-19 [29] and 20 [28] datasets. Note that nDCG@3 is the primary evaluation metric officially adopted by TREC CAsT [29]. We have two main observations.

First, QPP-GenRE significantly outperforms all QPP baselines on both datasets when provided with either type of query input. This demonstrates the ability of QPP-GenRE to generalize effectively to the conversational search domain. Second, QPP-GenRE achieves higher performance when provided with human-written query rewrites compared to T5-generated rewrites on both datasets. This finding highlights the critical role of high-quality query rewrites in effectively adapting QPP methods to conversational search scenarios; this finding also aligns with prior research [86].

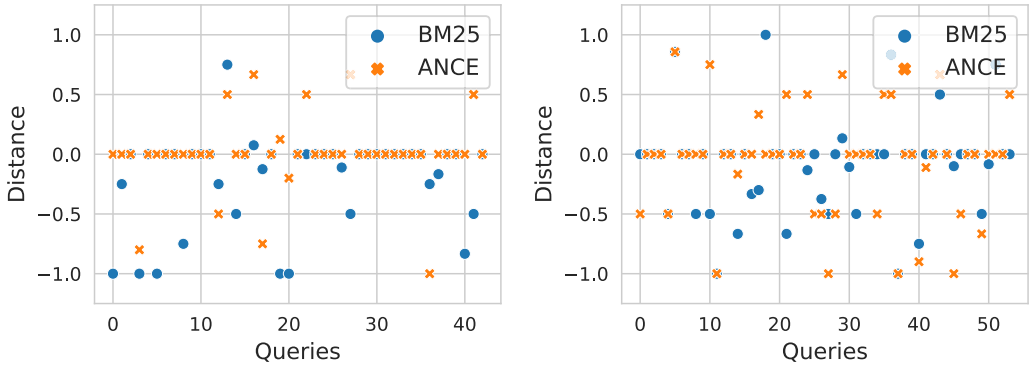
7.5 QPP-GenRE's Interpretability

As QPP-GenRE computes QPP based on generated relevance judgments, we analyze QPP errors from the perspective of relevance judgment generation. Figure 7 shows the QPP errors of QPP-GenRE integrated with LLaMA-7B in predicting the performance of BM25 and ANCE in terms of RR@10 on TREC-DL 19 and 20; the error is defined as the distance between the RR@10 values predicted by QPP-GenRE and actual RR@10 values, namely "predicted RR@10 minus actual RR@10." We find that most RR@10 values predicted by QPP-GenRE tend to be smaller than the actual RR@10 values, indicating that QPP-GenRE performs less effectively in identifying relevant items than irrelevant ones in the top of the ranked list. Table 7 shows the confusion matrices that compare relevance judgments made by TREC assessors (i.e., relevance judgments in *qrels*) and QPP-GenRE integrated with LLaMA-7B on TREC-DL 19 and 20. Table 8 provides a detailed breakdown of QPP-GenRE's prediction performance for each class, including metrics such as Precision, Recall, and F1 score. We find that QPP-GenRE tends to wrongly predict some relevant items as irrelevant (false negatives), which provides a further interpretation of the QPP errors we found above. Therefore, reducing false negatives in generating relevance judgments is a potential way to improve the QPP quality of QPP-GenRE. We leave this exploration for future work.

To show the superior interpretability of QPP-GenRE compared to other baselines, we provide a case study shown in Tables 9 and 10. Table 9 lists the predicted or ground-truth retrieval quality in terms of RR@10 of BM25 for the query "who is Robert Gra" on TREC-DL 2019. The predictions are made using widely used unsupervised QPP methods (WIG, NQC), supervised QPP methods (BERTQPP, qppBERT-PL), and QPP-GenRE.

We observe that the ground-truth RR@10 score for the query is 1, while the score predicted by QPP-GenRE is 0.5. QPP-GenRE infers RR@10 directly from the predicted relevance judgments for items in BM25's ranked list, we can conclude that while the top-ranked item is actually relevant to the query, QPP-GenRE mistakenly classified it as irrelevant. Table 10 provides supporting evidence by displaying the relevance judgments assigned by human annotators and QPP-GenRE for each item in BM25's ranked list. We found that QPP-GenRE fails to identify the top-ranked item as

[10] https://huggingface.co/castorini/t5-base-canard.



(a) TREC-DL 19 (b) TREC-DL 20

Fig. 7. The QPP errors of QPP-GenRE integrated with LLaMA-7B in predicting the performance of BM25 and ANCE in terms of RR@10 on TREC-DL 19 and 20. The distance is defined as "predicted RR@10 minus actual RR@10." The closer a query point is to 0 on the Y axis, the more accurately QPP-GenRE predicts its difficulty.

Table 7. Confusion Matrices Comparing Relevance Judgments Made by TREC Assessors and QPP-GenRE Integrated with LLaMA-7B on TREC-DL 19 and 20

QPP-GenRE	TREC-DL 19 assessors		TREC-DL 20 assessors	
	Relevant	Irrelevant	Relevant	Irrelevant
Relevant	752	553	486	763
Irrelevant	1,749	6,206	1,180	8,957

Table 8. Performance of Each Class for QPP-GenRE with LLaMA-7B on the TREC-DL 2019 and 2020 Qrels

	TREC-DL 19			TREC-DL 20		
Class	Precision	Recall	F1	Precision	Recall	F1
Relevant	0.576	0.301	0.395	0.389	0.292	0.333
Irrelevant	0.780	0.918	0.844	0.884	0.922	0.902

relevant. Specifically, QPP-GenRE does not identify the key part "Captain Robert Gray" in this item. This suggests that we could potentially improve QPP-GenRE's performance by further fine-tuning it to predict relevance on query–item pairs specifically related to influential historical figures.

However, all baselines lack interpretability compared to QPP-GenRE. WIG and NQC do not directly predict values for a specific IR metric, and their scores are difficult to interpret in isolation without comparing them to the scores for other queries. BERTQPP is trained to predict RR@10, but in this case, it returns a score of 0.3, indicating inaccurate performance prediction. Unfortunately, BERTQPP does not provide any intermediate outputs to help understand why it made this error or how its performance can be improved, limiting its interpretability and actionable insights. qppBERT-PL first predicts the number of relevant items in each chunk of the ranked list and then aggregates these numbers into an overall score. While it is possible to check the intermediate predictions of the number of relevant items per chunk, this information is too coarse to provide

Table 9. Retrieval Quality, in Terms of RR@10, Predicted by Various QPP Methods for the BM25 Retrieval of the Query "Who Is Robert Gra" (Query ID 1037798) on TREC-DL 19

QPP methods	Retrieval quality
WIG	2.427
NQC	0.106
BERTQPP	0.172
qppBERT-PL	0.368
QPP-GenRE	0.500
Ground-truth RR@10	1.000

Table 10. Ranked List Returned by BM25 for the Query "Who Is Robert Gra" (Query ID 1037798), Human-Labeled Relevance Judgments, and Ones Predicted by QPP-GenRE with LLaMA-7B on TREC-DL 19

Rank	Passage	Human	QPP-GenRE
1	Captain Robert Gray, May 1972. Discovering the Columbia River, May 1792 ... The Columbia River was given the name it bears today in May 1792...	Relevant	Irrelevant
2	Robert Gray. A surprise came on the Democratic side in the race for Mississippi Governor. Robert Gray, a retired firefighter and truck driver...	Irrelevant	Relevant
3	Team Mississippi Robert Gray for Governor Official Page. Robert Gray never would have made it without God...	Irrelevant	Irrelevant
4	I'm not a politician, said Gray in a Wednesday interview. I'm not a person who really wanted to run for Governor. Robert Gray is a 46-year-old truck driver...	Irrelevant	Relevant

detailed insights. In contrast, QPP-GenRE predicts the relevance of each individual item, offering more granular and informative insights.

7.6 Computational Cost Analysis

Table 11 shows the online QPP latency of QPP-GenRE integrated with LLaMA-7B and other BERT-based supervised QPP baselines, on TREC-DL 19, on a single NVIDIA A100 GPU. We compute the inference latency when queries are processed individually. For QPP-GenRE, we consider judging depths at 10, 100, and 200; QPP-GenRE can use batch acceleration for judging items for the same query because each item in a ranked list for a query is independent of each other.[11] Although QPP-GenRE is more expensive than all baselines when predicting one measure due to the much larger parameter size of LLaMA-7B compared to BERT, QPP-GenRE has lower latency compared to some baselines when predicting multiple IR evaluation measures because multiple measures can be derived from the same set of relevance judgments at no additional cost, e.g., while QPP-GenRE is 56% more expensive than M-QPPF for predicting one measure, it becomes more efficient when predicting two or more metrics than M-QPPF. Nevertheless, we acknowledge that QPP-GenRE

[11]qppBERT-PL first splits a ranked list with 100 items into 25 chunks and then predicts the number of relevant items in each chunk. For a fair comparison, we put 25 chunks into one batch for acceleration.

Table 11. Inference Efficiency of Supervised QPP Baselines and QPP-GenRE Integrated with LLaMA-7B on TREC-DL 19 to Predict One to Four Different IR Metrics

	Inference latency per query (ms)			
QPP method	1	2	3	4
NQA-QPP	118.40	236.80	355.20	473.60
BERTQPP	30.29	60.58	90.87	121.16
qppBERT-PL	316.80	316.80	316.80	316.80
M-QPPF	289.27	578.54	867.81	1,157.08
QPP-GenRE ($n = 10$)	452.60	452.60	452.60	452.60
QPP-GenRE ($n = 100$)	1,566.25	1,566.25	1,566.25	1,566.25
QPP-GenRE ($n = 200$)	2,845.43	2,845.43	2,845.43	2,845.43

n denotes QPP-GenRE's judgment depth in a ranked list. Cases with higher latency than QPP-GenRE ($n = 10$) are underlined.

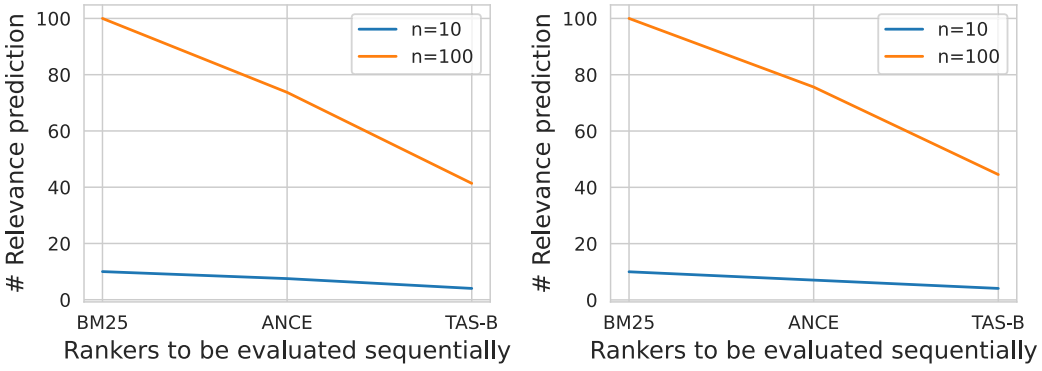
has higher computational costs than supervised QPP methods when predicting a single measure. Conversely, regression-based QPP baselines (NQA-QPP, BERTQPP, and M-QPPF) need to train separate models for different IR evaluation metrics. Although qppBERT-PL is not optimized to learn to output one specific IR evaluation measure, qppBERT-PL does not achieve a promising QPP quality (see Sections 6.1 and 6.2).

We argue that QPP-GenRE's latency is still much smaller than some high-performing LLM-based re-rankers, e.g., Sun et al. [119] show that a GPT-4-based listwise re-ranker needs 10 API calls (one call takes 3,200 ms) to re-rank 100 items for a query, resulting in 32,000 ms in total, which is around 20 times worse than QPP-GenRE's latency with a judging depth of 100. QPP-GenRE can well fit some knowledge-intensive professional search scenarios where QPP quality is prioritized or users may have a higher tolerance level for latency. Besides using QPP online, QPP can also be used to analyze a search system's performance in an offline setting [44].

Lastly, in order to enhance the efficiency of QPP-GenRE, we propose a *relevance judgment caching mechanism*. It reuses previously predicted relevance judgments for the same query when predicting the performance of new rankers. As a result, this mechanism helps conserve computational resources by eliminating the need to recompute relevance judgments that are shared among multiple rankers. Figure 8 shows the number of actual relevance predictions required for sequentially predicting the performance of BM25, ANCE, and TAS-B using the *relevance judgment caching mechanism*, with a judging depth of n equal to 10 or 100, on TREC-DL 19 and 20. We found that our proposed relevance judgment caching mechanism can reduce the number of LLM calls for relevance prediction by approximately 30%. For instance, on TREC-DL 19, with a judging depth of 10, the caching mechanism results in 21.15 LLM calls on average when sequentially predicting the performance of the three rankers (10 for BM25, 7.06 for ANCE, and 4.09 for TAS-B). In contrast, without using this mechanism, 30 LLM calls would be required (3×10).

8 Conclusions and Future Work

We have proposed a new QPP framework, QPP-GenRE, which models QPP from the perspective of predicting IR evaluation measures based on automatically generated relevance judgments. We have devised an approximation strategy for predicting an IR evaluation measure considering recall, which only judges a limited number of items in a given ranked list for a query, to avoid the cost of



(a) TREC-DL 19 (b) TREC-DL 20

Fig. 8. The average number of actual relevance predictions required for sequentially predicting the performance of BM25, ANCE, and TAS-B using the *relevance judgment caching mechanism*, with a judging depth of n equal to 10 or 100, on TREC-DL 19 and 20.

traversing the entire corpus to find all relevant items; the approximation strategy also enables us to study into the impact of various judging depths on QPP quality. We have explored using open source LLMs for generating relevance judgments, to ensure scientific reproducibility. In addition, we have examined training open source LLMs with PEFT on human-labeled relevance judgments, to improve the quality of relevance judgment generation and QPP.

Main Findings. Experiments on datasets from the TREC-DL 19–22 tracks demonstrate that QPP-GenRE significantly surpasses existing QPP approaches, achieving state-of-the-art QPP quality in assessing lexical and neural rankers for either a precision-oriented IR metric or an IR metric considering recall. Moreover, we have shown that fine-tuning open source LLMs on human-labeled relevance judgments is crucial for obtaining reliable relevance prediction and QPP results. Fine-tuning much smaller LLMs for relevance judgment prediction can yield more effective relevance prediction and QPP than few-shot prompting with much larger models. In particular, a fine-tuned 3B model (Llama-3.2-3B-Instruct) offers the best tradeoff between QPP quality and computational efficiency: It significantly outperforms 70B few-shot models and delivers performance comparable to fine-tuned 7B and 8B models. It implies that fine-tuning can offer strong performance at low inference costs. Furthermore, QPP-GenRE has the potential to conduct QPP more accurately when integrated with a more effective LLM, has a good compatibility with other types of relevance prediction methods (e.g., an LLMs-based re-ranker). Additionally, we have demonstrated that QPP-GenRE has great generalizability to the conversational search scenario. We have shown that QPP-GenRE exhibits good interpretability. Finally, we have found that our proposed *relevance judgment caching mechanism* can reduce LLM calls for relevance prediction by about 30%.

Broader Implications. QPP-GenRE has the potential to facilitate the practical use of QPP. The limited accuracy and interpretability of current QPP methods make them difficult to use in practical applications [8]. However, QPP-GenRE demonstrates significantly improved QPP accuracy and better interpretability, enhancing the reliability of QPP results and potentially facilitating the practical use of QPP. Especially, QPP-GenRE has the potential to benefit some knowledge-intensive professional search scenarios. In such scenarios, accurate QPP is prioritized, interpretable QPP results are needed, and users may have a higher tolerance level for latency. QPP-GenRE also has the potential for practical application in commercial search engines: Commercial search engines receive many frequent and repeated queries, and QPP-GenRE can improve QPP efficiency by reusing stored

relevance judgments for repeated query–item pairs and only generating relevance judgments for new query–item pairs. Moreover, QPP-GenRE can be used to analyze the ranking quality of a search system in a purely offline setting [44], where latency is not necessarily an issue.

Limitations and Future Work. First, we only consider predicting the ranking quality of widely used lexical and dense retrievers, and have not investigated QPP-GenRE's bias towards LLMs-based rankers [76]. Given that QPP-GenRE is based on LLM-based relevance predictors, it would be particularly interesting to explore QPP-GenRE's potential biases when it predicts the ranking quality of LLM-based rankers.

Second, QPP-GenRE is a QPP framework that can be integrated with various relevance prediction approaches. We show the success of QPP-GenRE equipped with various open source LLMs as well as a state-of-the-art pointwise LLM-based re-ranker, RankLLaMA [76]. Exploring various LLMs to find the optimal one for relevance prediction is beyond the scope of our work. However, in future, we believe it is valuable to investigate QPP-GenRE's performance integrated with other open source LLMs as relevance judgment generators. It is also interesting to adapt pairwise or listwise LLM-based re-rankers into relevance judgment generators and integrate QPP-GenRE with them.

Third, we only show QPP-GenRE's high effectiveness in predicting two primary metrics (RR@10 and nDCG@10) used at TREC DL 19–22 [22–25]. It is worthwhile to consider other metrics at various cutoffs in future work, e.g., nDCG@20 and MAP@100.

Fourth, while QPP-GenRE exhibits a promising QPP quality and can be used in scenarios where QPP quality is prioritized and users have a higher tolerance level for latency, e.g., patent search or post analysis, it is worth improving QPP-GenRE's efficiency in future to widen its scope of applications. We plan to investigate (i) the use of multiple GPUs because judging each item in a ranked list is independent of each other, (ii) distilling knowledge from LLMs to smaller language models [53], (iii) compressing LLMs by using lower-bit (e.g., 2-bit) quantization [18] or using low-rank factorization [135], and (iv) proposing an adaptive sampling approach that selects only a subset of documents from a ranked list for LLM-based relevance prediction to optimize the tradeoff between judgment cost and QPP performance.

Acknowledgment

We thank our reviewers and associate editor for their constructive feedback and suggestions.

References

[1] Zahra Abbasiantaeb, Chuan Meng, Leif Azzopardi, and Mohammad Aliannejadi. 2024. Can we use large language models to fill relevance judgment holes? arXiv:2405.05600. Retrieved from https://arxiv.org/abs/2405.05600

[2] Zahra Abbasiantaeb, Chuan Meng, Leif Azzopardi, and Mohammad Aliannejadi. 2025. Improving the reusability of conversational search test collections. In *ECIR*, 196–213.

[3] Zahra Abbasiantaeb, Chuan Meng, David Rau, Antonis Krasakis, Hossein A. Rahmani, and Mohammad Aliannejadi. 2023. LLM-based retrieval and generation pipelines for TREC interactive knowledge assistance track (iKAT). In *TREC*.

[4] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. 2004. Query difficulty, robustness, and selective application of query expansion. In *ECIR*, 127–137.

[5] Negar Arabzadeh, Amin Bigdeli, Morteza Zihayat, and Ebrahim Bagheri. 2021. Query performance prediction through retrieval coherency. In *ECIR*, 193–200.

[6] Negar Arabzadeh, Radin Hamidi Rad, Maryam Khodabakhsh, and Ebrahim Bagheri. 2023. Noisy perturbations for estimating query difficulty in dense retrievers. In *CIKM*, 3722–3727.

[7] Negar Arabzadeh, Maryam Khodabakhsh, and Ebrahim Bagheri. 2021. BERT-QPP: Contextualized pre-trained transformers for query performance prediction. In *CIKM*, 2857–2861.

[8] Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi, and Ebrahim Bagheri. 2024. Query performance prediction: From fundamentals to advanced techniques. In *ECIR*, 381–388.
[9] Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi, and Ebrahim Bagheri. 2024. Query performance prediction: Techniques and applications in modern information retrieval. In *SIGIR-AP*, 291–294.
[10] Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi, and Ebrahim Bagheri. 2025. Query performance prediction: Theory, techniques and applications. In *WSDM*, 991–994.
[11] Arian Askari, Mohammad Aliannejadi, Chuan Meng, Evangelos Kanoulas, and Suzan Verberne. 2023. Expand, highlight, generate: RL-driven document generation for passage reranking. In *EMNLP*, 10087–10099.
[12] Arian Askari, Chuan Meng, Mohammad Aliannejadi, Zhaochun Ren, Evangelos Kanoulas, and Suzan Verberne. 2024. Generative retrieval with few-shot indexing. arXiv:2408.02152. Retrieved from https://arxiv.org/abs/2408.02152
[13] Arian Askari, Roxana Petcu, Chuan Meng, Mohammad Aliannejadi, Amin Abolghasemi, Evangelos Kanoulas, and Suzan Verberne. 2025. Self-seeding and multi-intent self-instructing LLMs for generating intent-aware information-seeking dialogs. In *NAACL*.
[14] Javed A. Aslam and Virgil Pavlu. 2007. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In *ECIR*, 198–209.
[15] Rishi Bommasani, Percy Liang, and Tony Lee. 2023. Holistic evaluation of language models. *Annals of the New York Academy of Sciences* 1525, 1 (2023), 140–146.
[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*, 1877–1901.
[17] 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.
[18] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. 2023. QuIP: 2-Bit quantization of large language models with guarantees. arXiv:2307.13304. Retrieved from https://arxiv.org/abs/2307.13304
[19] Nuo Chen, Jiqun Liu, Xiaoyu Dong, Qijiong Liu, Tetsuya Sakai, and Xiao-Ming Wu. 2024. AI can be cognitively biased: An exploratory study on threshold priming in LLM-based batch relevance assessment. arXiv:2409.16022. Retrieved from https://arxiv.org/abs/2409.16022
[20] Xiaoyang Chen, Ben He, and Le Sun. 2022. Groupwise query performance prediction with BERT. In *ECIR*, 64–74.
[21] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. arXiv:2210.11416. Retrieved from https://arxiv.org/abs/2210.11416
[22] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 deep learning track. In *TREC*.
[23] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2019. Overview of the TREC 2019 deep learning track. In *TREC*.
[24] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Fernando Campos, and Jimmy Lin. 2021. Overview of the TREC 2021 deep learning track. In *TREC*.
[25] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Fernando Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2022. Overview of the TREC 2022 deep learning track. In *TREC*.
[26] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *SIGIR*, 299–306.
[27] Ronan Cummins, Joemon Jose, and Colm O'Riordan. 2011. Improved query performance prediction using standard deviation. In *SIGIR*, 1089–1090.
[28] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. CAsT 2020: The conversational assistance track overview. In *TREC*.
[29] Jeffrey Dalton, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan. 2020. Cast-19: A dataset for conversational information seeking. In *SIGIR*, 1985–1988.
[30] 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.
[31] 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 (2022), 1–31.
[32] Suchana Datta, Sean MacAvaney, Debasis Ganguly, and Derek Greene. 2022. A 'pointwise-query, listwise-document' based query performance prediction approach. In *SIGIR*, 2148–2153.
[33] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. arXiv:2305.14314. Retrieved from https://arxiv.org/abs/2305.14314
[34] Romain Deveaud, Josiane Mothe, and Jian-Yun Nie. 2016. Learning to rank system configurations. In *CIKM*, 2001–2004.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 4171–4186.
[36] Giorgio Maria Di Nunzio and Guglielmo Faggioli. 2021. A study of a gain based approach for query aspects in recall oriented tasks. *Applied Sciences* 11, 19 (2021), 9075.
[37] Fernando Diaz. 2007. Performance prediction using spatial autocorrelation. In *SIGIR*, 583–590.
[38] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. arXiv:2301.00234. Retrieved from https://arxiv.org/abs/2301.00234
[39] Andrew Drozdov, Honglei Zhuang, Zhuyun Dai, Zhen Qin, Razieh Rahimi, Xuanhui Wang, Dana Alon, Mohit Iyyer, Andrew McCallum, Donald Metzler, et al. 2023. PaRaDe: Passage ranking using demonstrations with LLMs. In *Findings of EMNLP*, 14242–14252.
[40] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. 2023. Perspectives on large language models for relevance judgment. In *ICTIR*, 39–50.
[41] Guglielmo Faggioli, Marco Ferrante, Nicola Ferro, Raffaele Perego, and Nicola Tonellotto. 2021. Hierarchical dependence-aware evaluation measures for conversational search. In *SIGIR*, 1935–1939.
[42] Guglielmo Faggioli, Nicola Ferro, Josiane Mothe, Fiana Raiber, and Maik Fröbe. 2023. Report on the 1st workshop on query performance prediction and its evaluation in new tasks (QPP++ 2023) at ECIR 2023. In *ACM SIGIR Forum*, Vol. 57, 1–7.
[43] Guglielmo Faggioli, Nicola Ferro, Cristina Muntean, Raffaele Perego, and Nicola Tonellotto. 2023. A spatial approach to predict performance of conversational search systems. In *IIR*, 41–46.
[44] Guglielmo Faggioli, Nicola Ferro, Cristina Ioana Muntean, Raffaele Perego, and Nicola Tonellotto. 2023. A geometric framework for query performance prediction in conversational search. In *SIGIR*, 1355–1365.
[45] Guglielmo Faggioli, Thibault Formal, Simon Lupart, Stefano Marchesin, Stephane Clinchant, Nicola Ferro, and Benjamin Piwowarski. 2023. Towards query performance prediction for neural information retrieval: Challenges and opportunities. In *ICTIR*, 51–63.
[46] 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 *ECIR*, 232–248.
[47] Guglielmo Faggioli, Oleg Zendel, J. Shane Culpepper, Nicola Ferro, and Falk Scholer. 2021. An enhanced evaluation framework for query performance prediction. In *ECIR*, 115–129.
[48] Maik Fröbe, Lukas Gienapp, Martin Potthast, and Matthias Hagen. 2023. Bootstrapped nDCG estimation in the presence of unjudged documents. In *ECIR*, 313–329.
[49] Debasis Ganguly, Suchana Datta, Mandar Mitra, and Derek Greene. 2022. An analysis of variations in the effectiveness of query performance prediction. In *ECIR*, 215–229.
[50] Debasis Ganguly and Emine Yilmaz. 2023. Query-specific variable depth pooling via query performance prediction. In *SIGIR*, 2303–2307.
[51] Aryo Gema, Luke Daines, Pasquale Minervini, and Beatrice Alex. 2023. Parameter-efficient fine-tuning of LLaMA for the clinical domain. arXiv:2307.03042. Retrieved from https://arxiv.org/abs/2307.03042
[52] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd-workers for text-annotation tasks. arXiv:2303.15056. Retrieved from https://arxiv.org/abs/2303.15056
[53] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. arXiv:2306.08543. Retrieved from https://arxiv.org/abs/2306.08543
[54] Soumyajit Gupta, Mucahid Kutlu, Vivek Khetan, and Matthew Lease. 2019. Correlation, prediction and ranking of evaluation metrics in information retrieval. In *ECIR*, 636–651.
[55] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2019. Performance prediction for non-factoid question answering. In *ICTIR*, 55–58.
[56] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. 2008. A survey of pre-retrieval query performance predictors. In *CIKM*, 1419–1420.
[57] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *SIGIR*, 113–122.
[58] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large language models are zero-shot rankers for recommender systems. arXiv:2305.08845. Retrieved from https://arxiv.org/abs/2305.08845
[59] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. In *ICLR*.
[60] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.

[61] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. arXiv:2310.06825. Retrieved from https://arxiv.org/abs/2310.06825
[62] Timothy Jones, Paul Thomas, Falk Scholer, and Mark Sanderson. 2015. Features of disagreement between retrieval effectiveness measures. In *SIGIR*, 847–850.
[63] 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.
[64] Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, and Guido Zuccon. 2024. Leveraging LLMs for unsupervised dense retriever ranking. arXiv:2402.04853. Retrieved from https://arxiv.org/abs/2402.04853
[65] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
[66] John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *SIGIR*, 111–119.
[67] Carlos Lassance and Stéphane Clinchant. 2023. The tale of two MSMARCO—and their unfair comparisons. In *SIGIR*.
[68] Victor Lavrenko and W. Bruce Croft. 2001. Relevance-based language models. In *SIGIR*, 120–127.
[69] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *SIGIR*, 2356–2362.
[70] Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *NeurIPS*.
[71] Tiedong Liu and Bryan Kian Hsiang Low. 2023. Goat: Fine-tuned LLaMA outperforms GPT-4 on arithmetic tasks. arXiv:2305.14201. Retrieved from https://arxiv.org/abs/2305.14201
[72] Lili Lu, Chuan Meng, Federico Ravenda, Mohammad Aliannejadi, and Fabio Crestani. 2025. Zero-shot and efficient clarification need prediction in conversational search. In *ECIR*, 389–404.
[73] Xiaolu Lu, Alistair Moffat, and J. Shane Culpepper. 2016. The effect of pooling and evaluation depth on IR metrics. *Information Retrieval Journal* 19, 4 (2016), 416–445.
[74] Yadong Lu, Chunyuan Li, Haotian Liu, Jianwei Yang, Jianfeng Gao, and Yelong Shen. 2023. An empirical study of scaling instruct-tuned large multimodal models. arXiv:2309.09958. Retrieved from https://arxiv.org/abs/2309.09958
[75] Shengjie Ma, Chong Chen, Qi Chu, and Jiaxin Mao. 2024. Leveraging large language models for relevance judgments in legal case retrieval. arXiv:2403.18405. Retrieved from https://arxiv.org/abs/2403.18405
[76] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning LLaMA for multi-stage text retrieval. arXiv:2310.08319. Retrieved from https://arxiv.org/abs/2310.08319
[77] Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. arXiv:2305.02156. Retrieved from https://arxiv.org/abs/2305.02156
[78] Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. 2021. LeCaRD: A legal case retrieval dataset for Chinese law system. In *SIGIR*, 2342–2348.
[79] Sean MacAvaney and Luca Soldaini. 2023. One-shot labeling for automatic relevance estimation. In *SIGIR*, 2230–2235.
[80] Craig Macdonald, Rodrygo L. T. Santos, and Iadh Ounis. 2012. On the usefulness of query features for learning to rank. In *CIKM*, 2559–2562.
[81] Mireille Makary, Michael Oakes, Ruslan Mitkov, and Fadi Yammout. 2017. Using supervised machine learning to automatically build relevance judgments for a test collection. In *DEXA*. IEEE, 108–112.
[82] Mireille Makary, Michael Oakes, and Fadi Yamout. 2016. Towards automatic generation of relevance judgments for a test collection. In *ICDIM*. IEEE, 121–126.
[83] Chuan Meng. 2024. Query performance prediction for conversational search and beyond. In *SIGIR*.
[84] Chuan Meng, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Performance prediction for conversational search using perplexities of query rewrites. In *QPP++2023*, 25–28.
[85] Chuan Meng, Mohammad Aliannejadi, and Maarten de Rijke. 2023. System initiative prediction for multi-turn conversational information seeking. In *CIKM*, 1807–1817.
[86] Chuan Meng, Negar Arabzadeh, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Query performance prediction: From ad-hoc to conversational search. In *SIGIR*, 2583–2593.
[87] Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi, and Maarten de Rijke. 2024. Ranked list truncation for large language model-based re-ranking. In *SIGIR*, 141–151.
[88] Chuan Meng, Guglielmo Faggioli, Mohammad Aliannejadi, Nicola Ferro, and Josiane Mothe. 2025. QPP++ 2025: Query performance prediction and its applications in the era of large language models. In *ECIR*, 319–325.
[89] Chuan Meng, Francesco Tonolini, Fengran Mo, Nikolaos Aletras, Emine Yilmaz, and Gabriella Kazai. 2025. Bridging the gap: From ad-hoc to proactive search in conversations. In *SIGIR*.
[90] Stefano Mizzaro, Josiane Mothe, Kevin Roitero, and Md Zia Ullah. 2018. Query performance prediction and effectiveness evaluation without relevance judgments: Two sides of the same coin. In *SIGIR*, 1233–1236.

[91] Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024. CHIQ: Contextual history enhancement for improving query rewriting in conversational search. In *EMNLP*, 2253–2268.
[92] Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. 2024. A survey of conversational search. arXiv:2410.15576. Retrieved from https://arxiv.org/abs/2410.15576
[93] Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. ConvGQR: Generative query reformulation for conversational search. In *ACL*, 4998–5012.
[94] Fengran Mo, Chuan Meng, Mohammad Aliannejadi, and Jian-Yun Nie. 2025. Conversational search: From fundamentals to frontiers in the LLM era. In *SIGIR*.
[95] Fengran Mo, Jian-Yun Nie, Kaiyu Huang, Kelong Mao, Yutao Zhu, Peng Li, and Yang Liu. 2023. Learning to relate to previous turns in conversational search. In *KDD*, 1722–1732.
[96] Alistair Moffat. 2017. Computing maximized effectiveness distance for recall-based metrics. *IEEE Transactions on Knowledge and Data Engineering* 30, 1 (2017), 198–203.
[97] Rabia Nuray and Fazli Can. 2003. Automatic ranking of retrieval systems in imperfect environments. In *SIGIR*, 379–380.
[98] Rabia Nuray and Fazli Can. 2006. Automatic ranking of information retrieval systems using data fusion. *Information Processing and Management* 42, 3 (2006), 595–614.
[99] Joaquín Pérez-Iglesias and Lourdes Araujo. 2010. Standard deviation as a query hardness estimator. In *SPIRE*, 207–212.
[100] Eduard Poesina, Radu Tudor Ionescu, and Josiane Mothe. 2023. IQPP: A benchmark for image query performance prediction. In *SIGIR*, 2953–2963.
[101] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. arXiv:2101.05667. Retrieved from https://arxiv.org/abs/2101.05667
[102] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankVicuna: Zero-shot listwise document reranking with open-source large language models. arXiv:2309.15088. Retrieved from https://arxiv.org/abs/2309.15088
[103] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankZephyr: Effective and robust zero-shot listwise reranking is a breeze! arXiv:2312.02724. Retrieved from https://arxiv.org/abs/2312.02724
[104] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. arXiv:2306.17563. Retrieved from https://arxiv.org/abs/2306.17563
[105] Sri Devi Ravana, Prabha Rajagopal, and Vimala Balakrishnan. 2015. Ranking retrieval systems using pseudo relevance judgments. *Aslib Journal of Information Management* 67, 6 (2015), 700–714.
[106] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline M. Hancock-Beaulieu, and Mike Gatford. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.
[107] Haggai Roitman. 2017. An enhanced approach to query performance prediction using reference lists. In *SIGIR*, 869–872.
[108] Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *EMNLP*, 3781–3797.
[109] Alireza Salemi and Hamed Zamani. 2024. Evaluating retrieval quality in retrieval-augmented generation. arXiv:2404.13781. Retrieved from https://arxiv.org/abs/2404.13781
[110] Mohammadreza Samadi and Davood Rafiei. 2023. Performance prediction for multi-hop questions. arXiv:2308.06431. Retrieved from https://arxiv.org/abs/2308.06431
[111] Andrea Santilli and Emanuele Rodolà. 2023. Camoscio: An Italian instruction-tuned Llama. arXiv:2307.16456. Retrieved from https://arxiv.org/abs/2307.16456
[112] Harrisen Scells, Leif Azzopardi, Guido Zuccon, and Bevan Koopman. 2018. Query variation performance prediction for systematic reviews. In *SIGIR*, 1089–1092.
[113] Anna Shtok, Oren Kurland, and David Carmel. 2010. Using statistical decision theory and relevance models for query-performance prediction. In *SIGIR*, 259–266.
[114] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems* 30, 2 (2012), 1–35.
[115] Ashutosh Singh, Debasis Ganguly, Suchana Datta, and Craig McDonald. 2023. Unsupervised query performance prediction for neural models utilising pairwise rank preferences. In *SIGIR*, 2486–2490.
[116] Ian Soboroff, Charles Nicholas, and Patrick Cahan. 2001. Ranking retrieval systems without relevance judgments. In *SIGIR*, 66–73.
[117] Jiuding Sun, Chantal Shaib, and Byron C. Wallace. 2023. Evaluating the zero-shot robustness of instruction-tuned language models. arXiv:2306.11270. Retrieved from https://arxiv.org/abs/2306.11270

[118] Weiwei Sun, Chuan Meng, Qi Meng, Zhaochun Ren, Pengjie Ren, Zhumin Chen, and Maarten de Rijke. 2021. Conversations powered by cross-lingual knowledge. In *SIGIR*, 1442–1451.
[119] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? Investigating large language models as re-ranking agents. In *EMNLP*, 14918–14937.
[120] Rikiya Takehi, Ellen M. Voorhees, and Tetsuya Sakai. 2024. LLM-assisted relevance assessments: When should we ask LLMs for help? arXiv:2411.06877. Retrieved from https://arxiv.org/abs/2411.06877
[121] Raphael Tang, Xinyu Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2023. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. arXiv:2310.07712. Retrieved from https://arxiv.org/abs/2310.07712
[122] Yongquan Tao and Shengli Wu. 2014. Query performance prediction by considering score magnitude and variance together. In *CIKM*, 1891–1894.
[123] Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. 2022. Ul2: Unifying language learning paradigms. In *ICLR*.
[124] Paul Thomas, Falk Scholer, Peter Bailey, and Alistair Moffat. 2017. Tasks, queries, and rankers in pre-retrieval performance prediction. In *ADCS*, 1–4.
[125] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large language models can accurately predict searcher preferences. In *SIGIR*, 1930–1940.
[126] Stephen Tomlinson, Douglas W. Oard, Jason R. Baron, and Paul Thompson. 2007. Overview of the TREC 2007 legal track. In *TREC*.
[127] Nicola Tonellotto, Craig Macdonald, and Iadh Ounis. 2013. Efficient and effective retrieval using selective pruning. In *WSDM*, 63–72.
[128] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. arXiv:2302.13971. Retrieved from https://arxiv.org/abs/2302.13971
[129] Shivani Upadhyay, Ehsan Kamalloo, and Jimmy Lin. 2024. LLMs can patch up missing relevance judgments in evaluation. arXiv:2405.04727. Retrieved from https://arxiv.org/abs/2405.04727
[130] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. 2024. A large-scale study of relevance assessments with large language models: An initial look. arXiv:2411.08275. Retrieved from https://arxiv.org/abs/2411.08275
[131] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024. UMBRELA: UMbrela is the (open-source reproduction of the) Bing RELevance Assessor. arXiv:2406.06519. Retrieved from https://arxiv.org/abs/2406.06519
[132] Maria Vlachou and Craig Macdonald. 2023. On coherence-based predictors for dense query performance prediction. arXiv:2310.11405. Retrieved from https://arxiv.org/abs/2310.11405
[133] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, Vol. 35, 24824–24837.
[134] 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*.
[135] Mingxue Xu, Yao Lei Xu, and Danilo P. Mandic. 2023. TensorGPT: Efficient compression of the embedding layer in LLMs based on the tensor-train decomposition. arXiv:2307.00526. Retrieved from https://arxiv.org/abs/2307.00526
[136] Le Yan, Zhen Qin, Honglei Zhuang, Rolf Jagerman, Xuanhui Wang, Michael Bendersky, and Harrie Oosterhuis. 2024. Consolidating ranking and relevance predictions of large language models through post-processing. arXiv:2404.11791. Retrieved from https://arxiv.org/abs/2404.11791
[137] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *SIGIR*, 829–838.
[138] Hamed Zamani, W. Bruce Croft, and J. Shane Culpepper. 2018. Neural query performance prediction using weak supervision from multiple signals. In *SIGIR*, 105–114.
[139] Oleg Zendel, Binsheng Liu, J. Shane Culpepper, and Falk Scholer. 2023. Entropy-based query performance prediction for neural information retrieval systems. In *QPP++2023*, 37–44.
[140] Hengran Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. Are large language models good at utility judgments? In *SIGIR*, 1941–1951.
[141] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. arXiv:2308.10792. Retrieved from https://arxiv.org/abs/2308.10792
[142] Xinyu Zhang, Sebastian Hofstätter, Patrick Lewis, Raphael Tang, and Jimmy Lin. 2023. Rank-without-GPT: Building GPT-independent listwise rerankers on open-source large language models. arXiv:2312.02969. Retrieved from https://arxiv.org/abs/2312.02969

[143] Yue Zhang, Leyang Cui, Deng Cai, Xinting Huang, Tao Fang, and Wei Bi. 2023. Multi-task instruction tuning of LLaMa for specific scenarios: A preliminary study on writing assistance. arXiv:2305.13225. Retrieved from https://arxiv.org/abs/2305.13225

[144] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *NeurIPS*, Vol. 36.

[145] Yun Zhou and W. Bruce Croft. 2006. Ranking robustness: A novel framework to predict query performance. In *CIKM*, 567–574.

[146] Yun Zhou and W. Bruce Croft. 2007. Query performance prediction in web search environments. In *SIGIR*, 543–550.

[147] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. arXiv:2308.07107. Retrieved from https://arxiv.org/abs/2308.07107

[148] Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Berdersky. 2023. Beyond yes and no: Improving zero-shot LLM rankers via scoring fine-grained relevance labels. arXiv:2310.14122. Retrieved from https://arxiv.org/abs/2310.14122

[149] Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source large language models are strong zero-shot query likelihood models for document ranking. arXiv:2310.13243. Retrieved from https://arxiv.org/abs/2310.13243

[150] Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2023. A setwise approach for effective and highly efficient zero-shot ranking with large language models. arXiv:2310.09497. Retrieved from https://arxiv.org/abs/2310.09497

Received 16 June 2024; revised 9 February 2025; accepted 10 April 2025