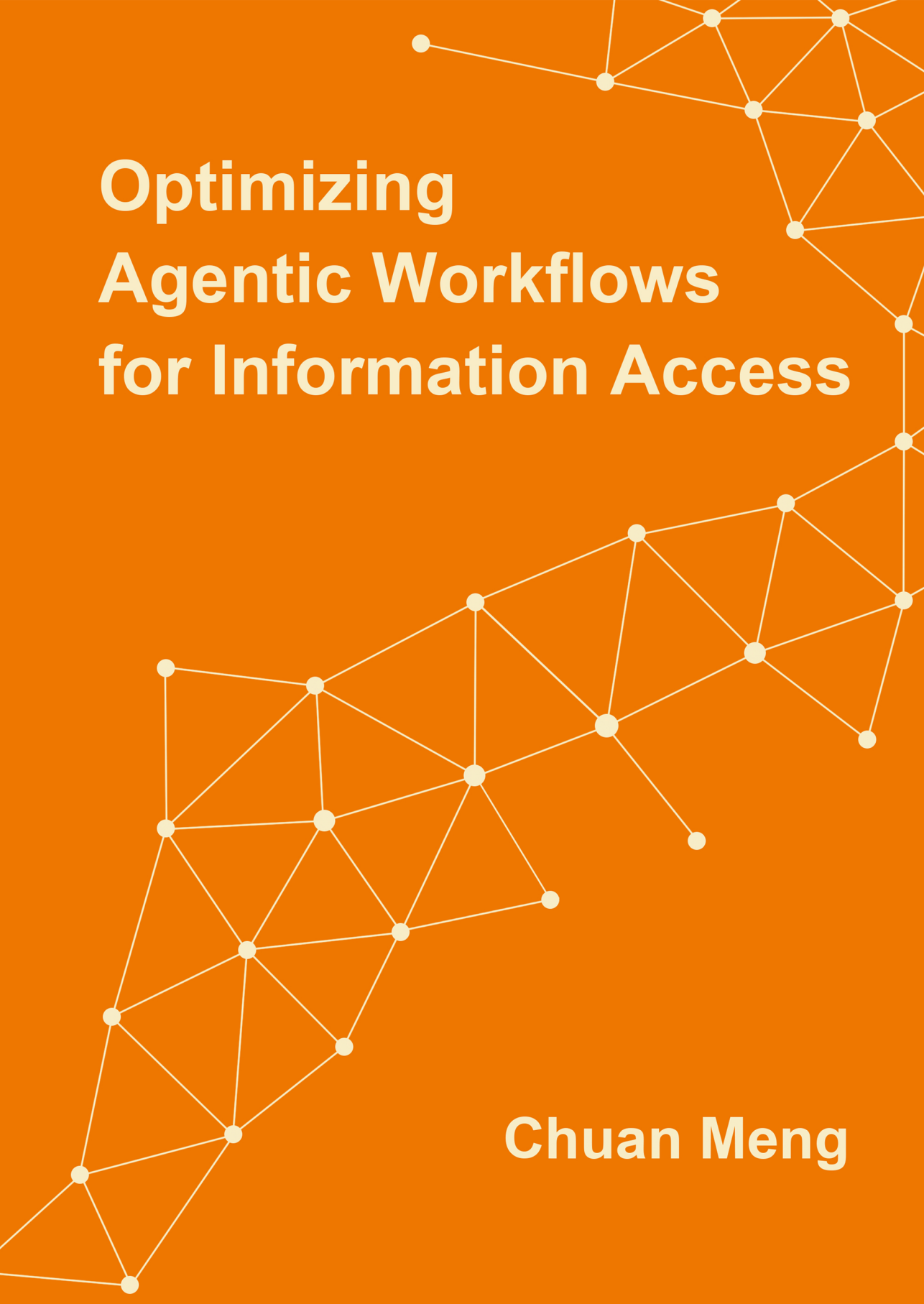
Optimizing Agentic Workflows for Information Access

Meng, C.

Publication date
2025
Document Version
Final published version

Optimizing Agentic Workflows for Information Access



Chuan Meng

Optimizing Agentic Workflows for Information Access

Chuan Meng

Optimizing Agentic Workflows for Information Access

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in
de Aula der Universiteit
op donderdag 19 juni 2025, te 14.00 uur

door

Chuan Meng

geboren te Shandong

Promotiecommissie

Promotor:	prof. dr. M. de Rijke	Universiteit van Amsterdam
Copromotor:	dr. M. Alian Nejadi	Universiteit van Amsterdam
Overige leden:	prof. dr. E. Kanoulas	Universiteit van Amsterdam
	prof. dr. C. Monz	Universiteit van Amsterdam
	dr. Z. Ren	Leiden University
	dr. A.C. Yates	Universiteit van Amsterdam
	prof. dr. E. Yilmaz	University College London

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The research was carried out at the Information Retrieval Lab (IRLab) at the University of Amsterdam, with support from the China Scholarship Council (CSC) under grant number 202106220041.

Cover by Chuan Meng
Printed by Ridderprint, The Netherlands

ISBN: 978-94-6522-323-0

To my beloved and great father, Xiangfeng Meng

The limits of my language mean the limits of my world.

— Ludwig Wittgenstein (1889–1951)

Acknowledgements

On the 3rd of October 2021, I took two suitcases and flew from Beijing to Amsterdam to formally begin my PhD journey. It was a rainy day in Amsterdam, as if foreshadowing the challenges ahead. Over the past three years, I have struggled with unpredictable winds and rain, language barriers, and limited food choices. But these challenges have never ever shaken my ambition to become a true researcher and scientist. To me, scientific research is always interesting. I feel addicted to it, and often forget about time when I am working. The happiness I feel when my ideas work, when my papers get published, and when my work inspires others has always been the force that keeps me moving forward.

First, I want to thank Maarten de Rijke, my supervisor. Maarten, you have taught me valuable lessons about how to become a professional researcher. The advice you shared deeply impressed me, really helped me throughout my PhD journey, and continues to influence how I mentor my own students. Some of your words have stayed especially close to my heart. "*If you feel there is a gap, there is a gap*" has made me much more confident when making decisions. "*Prestige would not really make you happy, but community and colleagues do*" helped me realize that belonging, support and connection with others are important forces that keep me going over the long term. "*A good PhD student should not only succeed themselves, but also help others succeed*" reminds me that making a positive impact on people around me is also an important mission of a researcher. I know these words will remain with me throughout my academic career.

Next, I want to thank Mohammad Aliannejadi, my co-supervisor. Mohammad, you are nice, approachable, and patient. What impressed me the most is that you always positively encouraged me. When I faced uncertainty or trouble, you often said, "*Try it. Don't limit yourself.*" That gave me the courage to explore and take action. When I experienced paper rejections and other setbacks, your words warmed my heart again and again and helped me recover quickly. One that I will always remember is: "*Be resilient to rejections. In academia, rejection is more common than acceptance.*" These moments helped me understand how important encouragement is for a PhD student. You set a great example for me. I want to be someone who supports my students like you supported me.

I am also grateful to Evangelos Kanoulas, Christof Monz, Zhaochun Ren, Andrew Yates, and Emine Yilmaz for being members of my PhD committee. Thank you for taking time to review my thesis.

In addition, I would like to thank Arian and Simon for agreeing to be my paranymphs. I am really grateful to have you by my side on such an important day.

Moreover, I am especially thankful to my fellow PhD collaborators, Negar, Arian, and Fengran. We worked side by side on papers, shared ideas, and chased deadlines. We went through both the frustration of rejections and the joy of acceptance together. We learned from each other and grew together as researchers. Working with you has been one of the most valuable parts of this journey.

Beyond that, I am also grateful to other collaborators during my PhD. Thanks to Amin, Antonis, David Rau, Evangelos, Fabio, Federico, Leif, Lili, Roxana, Saeed, Suzan, and Zahra for collaborating on research papers. Thanks to Jian-Yun for co-

organizing a tutorial on conversational search, and to Ebrahim for co-organizing tutorials on query performance prediction. I would also like to thank Guglielmo, Nicola, and Josiane for co-organizing a workshop on query performance prediction.

Furthermore, thank my other colleagues in my lab who accompanied me throughout my PhD journey: Ali, Ana, Arezoo, Barrie, Catherine, Chen Xu, Clara, Clemencia, Cosimo, Dan, Daniel, David Vos, Gabriel, Gabrielle, Hongyi, Ian, Ivana, Jasmin, Jia-Hong, Jia-Huei, Jiahuan, Jie Zou, Jin, Jingfen, Jingwei, Julien, Kidist, Kiki, Lu, Maarten Marx, Maartje, Maria, Mariya, Maurits, Maxime, Ming, Mohanna, Mozhdeh, Olivier, Panagiotis, Peibo, Petra, Philipp, Pooya, Romain, Roxana, Ruben, Ruqing, Samarth, Sami, Shaojie, Shashank, Siddharth, Simon, Songgaojun, Teng, Thilina, Thong, Vaishali, Vera, Weijia, Xinyi, Yangjun, Yibin, Yifei, Yixing, Yongkang, Yougang, Yuanna, Yuanxing, Yubao, Yueqing, Yuyue, Zhirui, Zihan, and Ziyi.

I am also grateful to the people I met during my internship as an Applied Scientist Intern at Amazon. Thank you to my manager, Gabriella Kazai, for your kind support throughout my internship. I still remember that on the day of the SIGIR 2025 submission deadline, I was touched when you thoughtfully brought me fruit to help me stay energized and get through the day. Thank you to my mentor, Francesco Tonolini. I learned a lot from your mathematical thinking and your ability to respond quickly. Thanks also to Nikolaos and Emine for your insightful ideas and your efforts in writing the paper. I am also grateful to other team members, including Daniele, Daniil, Jordan, Matej and Süha, for your help and support. Hi Matej, I really miss the fun we had playing board games together. Thank you to Chen Luo, Lei He, Nikos Voskarides, Preetam Dammu, Roi Blanco, Shervin Malmasi, and Tim Heithaus for the inspiring discussions and valuable feedback during my time at Amazon. I would also like to thank the other friends I met during the internship: Xinyu, Yulong, Zheng, Chengxi, and Yucheng. I was always happy when we were together, and I enjoyed the time we shared. Special thanks to Yulong for your helpful advice on my career development.

Beyond my lab and internship, I was also fortunate to meet other researchers. Special thanks to Iadh for inviting me to give talks at the University of Glasgow, Xi and Saeed for inviting me to give a talk at UCL, and Pablo Mendes for inviting me to give a talk at your startup. I am also grateful to Craig, Sean, Jiqun, Yiding, Ziming, Zhengyan, Zaiqiao, Hansi, Tao Yang, and Jie Yang for the insightful research discussions.

Looking further back, I would like to thank my supervisors during my master's studies at Shandong University: Zhumin Chen, Pengjie Ren, and Zhaochun Ren. You were the ones who first guided me into academic research and helped me develop good research habits. Thanks to your help, I was able to start my PhD journey smoothly.

In my daily life, I am grateful to my neighbours: Weijie, Xiaotian, Jiangyun, Fei Xue, and Weijian. Your help made my PhD journey easier and warmer.

Last but certainly not least, I want to express my deepest gratitude to my beloved and great father, Xiangfeng Meng. Dad, every year your birthday wishes to me arrived right on time, and they always brought tears to my eyes. Thank you for all your care and love over the past three years.

Chuan Meng
Amsterdam
April, 2025

Contents

1 Introduction

Information access systems play a vital role in connecting people to information that is crucial for decision-making and taking actions in the world [267]. Such systems have been embedded into the capillaries of human society, especially search engines and recommender systems [137]. Additionally, large language model (LLM)-based chatbots are increasingly emerging as a new gateway for individuals to access information [137, 214, 279].

Static workflows have been widely used in information access systems to fulfill users' information needs [227]. Figure 1.1 provides an example of a static workflow for information access, where a user query is first processed by a retriever to obtain candidate documents, followed by a re-ranker that refines the ranked results. The workflow can end at this stage, presenting the user with the final ranked list, often referred to as the "ten blue links" [141]. In recent years, the workflow has been further extended by introducing a response generator (e.g., LLMs [136]), which generates a natural response to the user grounded in both the user query and documents, a.k.a. retrieval-augmented generation (RAG) [4, 134]. However, this "one-size-fits-all" static workflow limits the ability of information access systems to address real-world user queries in complex scenarios that demand adaptive and case-by-case handling [227].

Agents offer a promising solution to the limitations of static workflows. In the context of artificial intelligence (AI), an agent is an autonomous entity that makes decisions and takes actions on users' behalf [220]. The idea of agents traces back to the 1950s with the emergence of symbolic AI [220]. More recently, agents have attracted significant attention, benefiting from capabilities of LLMs [47]. *Agentic workflows* are structured interaction sequences of autonomous agents [288]. Due to their autonomous decision-making abilities, agents can determine adaptive execution paths that dynamically respond to each user request, ultimately achieving better results in terms of effectiveness or efficiency [227].

This thesis explores agentic workflows for information access. To the best of our knowledge, there is only limited work on formally defining agentic workflows for information access. In this thesis, we consider an agentic workflow for information access to be a workflow in which one or multiple autonomous agents dynamically adjust execution paths to each user query. Agentic workflows, with their query-adaptive execution paths, stand in contrast to static workflows, which follow a fixed execution process for all user queries. We consider prior studies in information access that develop

Figure 1.1: Example of a static workflow for information access.

autonomous agents to enable query-adaptive execution paths as contributions to the broader landscape of agentic workflow research in information access.

Prior work has studied various components within agentic workflows for information access. This thesis focuses on three key components: mixed-initiative strategy planning, ranking strategy planning, and ranking result reflection.

First, we look at **mixed-initiative strategy planning**. As modern information access processes become increasingly interactive, systems must be capable of engaging in complex multi-turn conversations to meet users' information needs [12]. Furthermore, mixed-initiative is a key aspect of modern information access, where both the user and the system can take the initiative at different points in a conversation [203]. An effective information access system can ask clarifying questions in response to users' ambiguous queries [7, 11, 218, 284], elicit user preferences [204, 219], ask for feedback [246, 250], and so on. Because system initiative-taking at a wrong time could come at the risk of user frustration, hence hurting the overall user experience [263, 264, 305, 306], it is important for an information access system to plan an appropriate mixed-initiative strategy in response to a user query [27, 286], i.e., predict the right moment to take the initiative. Existing work has extensively studied clarification need prediction agents, which aim to predict the right time to ask a clarifying question [9, 11, 16, 263, 264, 273].

Second, we examine **ranking strategy planning**. Existing research has explored various ranking strategies that dynamically adapt to each user query, optimizing both efficiency and effectiveness, e.g., adaptive retrieval [116], where an agent decides whether to bypass retrieval entirely or select between single-step and multi-step retrieval based on the complexity of the user query. This thesis focuses on dynamic per-query re-ranking depth prediction [58, 135, 262, 285]. Re-ranking is a crucial step in refining retrieval results by re-ordering candidate documents in descending order of query–document relevance. Compared to retrievers, re-rankers are typically more precise in predicting query–document relevance but they come at a significantly higher computational cost [139]. A common practice in re-ranking is to apply a fixed re-ranking depth for all queries, i.e., a pre-determined number of top-retrieved documents are always passed to the re-ranking process. However, individual queries might need a shorter or a longer list of re-ranking candidates [36], i.e., a fixed re-ranking depth can lead to unnecessary computational cost for queries that need fewer candidates, while limiting the potential effectiveness for queries that benefit from a deeper re-ranking process. Dynamic per-query re-ranking depth prediction tackles the limitations of using fixed depths by predicting a dynamic re-ranking depth on a per-query basis (with a depth of zero meaning bypassing re-ranking entirely), potentially enhancing both efficiency [58, 135, 262] and effectiveness [285]. In this sense, a method for dynamic per-query re-ranking depth prediction can be seen as an agent that enables agentic workflows.

Third, we turn to **ranking result reflection**. Reflection is a critical component

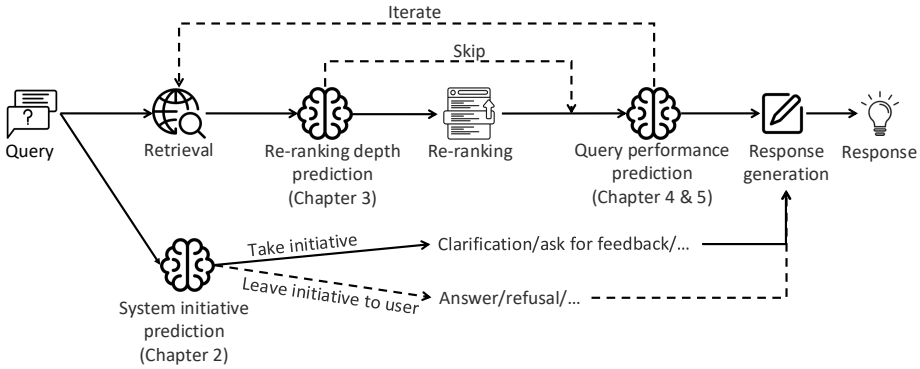


Figure 1.2: Example of an agentic workflow for information access.

in agentic workflows [267]; it enables systems to iteratively evaluate and refine their outputs [227, 267]. In information access scenarios, it is important to reflect not only on the quality of final generated responses but also on the quality of ranking results, as poor rankings can significantly degrade the subsequent response generation quality [277]. Therefore, it is crucial to effectively measure ranking result quality. If the ranking is predicted to be low-quality, the system can adjust the execution path to avoid passing the retrieved documents to the response generation phase, and instead iteratively refine the ranking quality, e.g., by reformulating the user query or leveraging alternative retrieval corpora [277]. This thesis focuses specifically on ranking result reflection. While various research directions exist for automatically evaluating ranking quality, this thesis is dedicated to exploring query performance prediction (QPP) [39] as a means of reflecting on ranking quality. QPP has been extensively studied in the information retrieval (IR) community for decades [15, 56, 102, 209, 223, 224, 239, 300]; it aims to predict the ranking quality of a search system for a query without relying on human-labeled relevance judgments [180]. Effective QPP has been used to adjust execution paths to optimize ranking quality on a per-query basis, e.g., query routing [215], selective query expansion [13], retriever selection [122], and IR system configuration selection [70, 244]. In this sense, a QPP method can be viewed as an agent that enables agentic workflows.

There are important limitations in the above three key components. In mixed-initiative strategy planning, **a narrow system-initiative scope is considered in predicting the timing of system-initiative taking.** While clarification need prediction has been widely studied, clarification is only one of many possible system-initiative actions [10, 29, 286], yet research on predicting the timing for other system-initiative actions has received little attention.

In ranking strategy planning, **only limited research explores dynamic per-query re-ranking depth prediction in LLM-based re-ranking.** Recently proposed LLM-based re-rankers [153, 195, 196, 198, 234, 293, 302] with billions of parameters [304] lead to a substantial increase in latency, making their deployment in real-world applications challenging. While dynamic per-query re-ranking depth prediction offers a promising solution for achieving a good effectiveness/efficiency trade-off in LLM-based

re-ranking, research in this area remains scarce.

In ranking result reflection, there are two key issues in using QPP:

(1) **Little research explores QPP in conversational search.** As mentioned earlier, modern information access systems should be capable of engaging in complex multi-turn conversations. This means that ranking result reflection must assess how well ranking results align with users' conversational queries. While QPP has been studied extensively in ad-hoc search, research on QPP in the emerging area of conversational search [182] is limited.

(2) **Little research explores improving QPP accuracy by leveraging LLMs' capabilities.** For QPP to be effective in ranking result reflection, it must provide highly accurate assessments of ranking results, ensuring that refinements are triggered at the right time. While LLMs have demonstrated state-of-the-art performance across various tasks in IR and natural language processing (NLP) [297, 301], no prior work has explored leveraging LLMs for enhancing QPP accuracy yet. It is unclear how LLMs can be effectively used to improve QPP accuracy.

This thesis aims to optimize agentic workflows for information access by improving the three critical components listed above: mixed-initiative strategy planning, ranking strategy planning, and ranking result reflection. Specifically, we enhance these components by addressing the four key limitations identified above.

1.1 Research Outline and Questions

This thesis aims to answer the following overarching question:

> *"What key aspects of agentic workflows for information access should be optimized, and how can we effectively optimize these aspects?"*

To address this question, this thesis identifies and addresses four limitations in three critical components of agentic workflows for information access. The thesis is structured into three parts and four chapters. Each part corresponds to one of the three components, and each chapter addresses a specific limitation. Figure 1.2 illustrates the themes across all chapters.

The first part of the thesis focuses on the mixed-initiative strategy planning component. This part contains a single chapter that focuses on resolving the issue of a narrow scope in system-initiative actions for predicting the timing of system initiative-taking.

The second part focuses on the ranking strategy planning component. This part consists of one chapter that aims to fill the research gap in dynamic per-query re-ranking depth prediction in the context of LLM-based re-ranking.

The third part focuses on the ranking result reflection component, comprising two chapters. One aims to fill the research gap in QPP in the emerging area of conversational search, while the other examines the under-explored area of LLM-enhanced QPP.

1.1.1 Mixed-initiative strategy planning

In the first part of the thesis, we focus on optimizing the mixed-initiative strategy planning component. It comprises Chapter 2, which aims to resolve the issue of a narrow scope of system-initiative actions in predicting the timing of system initiative-taking.

We broaden the scope of system-initiative actions by defining and modeling a new task, system initiative prediction (SIP). The SIP task is to predict the timing of system initiative that covers a broad range of specific system initiative-taking actions. Due to its broad coverage of specific system initiative-taking actions, SIP operates as a high-level strategic decision, potentially offering two primary benefits for downstream tasks within information access systems. First, SIP can capture knowledge shared across different system-initiative actions. The shared knowledge can be leveraged to enhance the prediction of a specific system-initiative action through transfer learning. For example, a model pre-trained on SIP can be further fine-tuned on the downstream task of clarification need prediction [11] to improve its performance. Second, SIP can narrow the decision space of some downstream tasks, thereby improving their accuracy; for instance, in action prediction, an action like requesting feedback is triggered only if the system decides to take the initiative. Given these potential compelling benefits, the next step is to explore modeling SIP and empirically validate its usefulness for downstream tasks. This leads us to the following research question:

RQ1 How can we effectively model system initiative prediction (SIP), and how does this prediction benefit downstream tasks?

To address this question, we first conduct an empirical analysis revealing structural dependencies between system-initiative decisions and other factors in multi-turn conversations. Motivated by these insights, we introduce a multi-turn system-initiative predictor (MuSIc) that builds on conditional random fields (CRFs), a class of probabilistic graphical models known for effectively capturing structural dependencies while offering strong interpretability and transparency. We further explore how SIP can benefit two downstream tasks: clarification need prediction and action prediction. For the former, we propose a SIP-to-clarification transfer learning method, which transfers knowledge gained from SIP to improve clarification need prediction performance. For action prediction, we introduce a SIP-aware hierarchical framework, where action prediction depends on SIP outcomes. Experimental results show that MuSIc achieves state-of-the-art SIP performance, and significantly improves both clarification need prediction and action prediction tasks. Additionally, a visual analysis highlights MuSIc's strong interpretability and transparency.

1.1.2 Ranking strategy planning

The second part of this thesis focuses on optimizing the ranking strategy planning component. It comprises Chapter 3, which aims to fill the research gap in dynamic per-query re-ranking depth prediction in LLM-based re-ranking. This thesis identified two key dimensions of the research gap. On the one hand, despite the growing importance of LLM-based re-ranking [153, 195, 196, 198, 234, 293, 302], in this scenario, there

remains a lack of systematic empirical analysis examining the potential advantages of adopting dynamic re-ranking depths over fixed ones in this context. On the other hand, effective methods for predicting dynamic re-ranking depths, specifically for LLM-based re-ranking, have not been clearly established. Existing studies predict dynamic re-ranking depths for non-LLM-based re-ranking by developing candidate pruning methods [58, 135, 262] or using ranked list truncation (RLT) [285] methods. This thesis focuses on RLT methods, and explores to what extent RLT methods can effectively predict dynamic re-ranking depths in the scenario of LLM-based re-ranking. The above considerations motivate the following research question.

RQ2 In the context of LLM-based re-ranking, what are the potential benefits of using dynamic per-query re-ranking depths over fixed ones, and to what extent can RLT methods effectively predict dynamic re-ranking depths?

To address this question, we begin by conducting a systematic empirical analysis to identify the limitations of fixed re-ranking depths and explore the potential advantages of dynamic re-ranking depths in the context of LLM-based re-ranking. Our analysis reveals that dynamic re-ranking depths not only enhance re-ranking efficiency but also improve re-ranking effectiveness. Next, we carry out a comprehensive study to examine how effectively RLT methods adapt to predicting dynamic re-ranking depths in the context of LLM-based re-ranking. Our leading experimental result indicates that RLT methods do not show a clear advantage over using a fixed re-ranking depth.

1.1.3 Ranking result reflection

In the third part of the thesis, we focus on the ranking strategy planning component. It comprises Chapters 4 and 5. Both chapters focus on QPP: Chapter 4 aims to fill the research gap in QPP for emerging conversational search, while Chapter 5 explores the underexplored area of LLM-enhanced QPP.

Chapter 4 decomposes the research gap in QPP for conversational search into two main aspects: the lack of evaluation of existing QPP methods in conversational search and the absence of effective adaptation strategies for these methods in this setting. While QPP methods for traditional ad-hoc search have been studied extensively, their applicability to the emerging field of conversational search remains largely unexplored. Conversational search introduces several distinct characteristics compared to ad-hoc search, such as conversational user queries [61, 62, 115] and a greater emphasis on the ranking quality of top-ranked results [286]. Given this research gap, it is critical to examine how well existing ad-hoc QPP methods generalize to the conversational search context. However, adapting these methods is not straightforward. The primary challenge arises from the context-dependent nature of conversational queries, which frequently include omissions, coreferences, and ambiguities [182]. Many QPP methods rely heavily on the input query [15, 45, 65, 67, 102, 283], yet they are designed for self-contained queries in ad-hoc search and lack the necessary capabilities to interpret context-dependent conversational queries effectively. The above concerns motivate our third research question:

RQ3 How can QPP methods originally designed for ad-hoc search be effectively

adapted to conversational search, and how well do QPP methods for ad-hoc search perform in conversational search?

To answer this question, we start by adapting existing QPP methods that rely on input queries for conversational search. Specifically, we feed self-contained query rewrites, generated by off-the-shelf query rewriting models, into these methods. Using this adaptation approach, we conduct a comprehensive study to evaluate how well QPP methods designed for ad-hoc search perform in conversational search. Below, we highlight a few key findings from our experiments. Our extensive results indicate that feeding query rewrites into QPP methods is an effective strategy for adapting them to conversational search. Additionally, predicting an IR evaluation metric with a shallow cut-off is generally more challenging than predicting one with a deep cut-off.

In Chapter 5, we focus on improving QPP accuracy by leveraging LLMs. While LLMs have demonstrated state-of-the-art performance across various tasks in IR and NLP [297, 301], limited research has explored how to use LLMs for enhancing QPP accuracy. In particular, it is unclear how LLMs can be effectively used to improve the accuracy of QPP methods. Hence, we address the following research question:

RQ4 How can LLMs be used to effectively enhance QPP accuracy?

To answer this question, we propose a framework for modeling QPP using automatically generated relevance judgments (QPP-GenRE), which decomposes QPP into independent subtasks of predicting the relevance of each document in a ranked list to the query, and then predicts different IR evaluation measures based on the relevance predictions. QPP-GenRE leverages the strong performance of LLMs in generating relevance judgments [249], with prior research demonstrating that LLMs can achieve accuracy comparable to human labelers [242]. To further enhance the quality of generated relevance judgments, we propose to fine-tune open-source LLMs on human-labeled relevance judgments.

Moreover, since predicting IR evaluation metrics considering recall typically requires identifying all relevant documents in the entire corpus, we propose an approximation strategy for QPP-GenRE; it predicts relevance for a limited subset of ranked documents and uses their relevance judgments to estimate recall-based IR metrics, circumventing the computational expense of traversing the entire corpus to find all relevant documents.

Additionally, to mitigate the efficiency issue that comes with calling LLMs, we introduce a relevance judgment caching mechanism that improves efficiency by reusing previously predicted relevance judgments. Extensive experiments show that QPP-GenRE achieves state-of-the-art QPP accuracy in assessing both lexical and neural retrievers across ad-hoc and conversational search scenarios. Also, fine-tuning significantly improves LLMs' performance in relevance judgment prediction, as well as the accuracy of QPP based on these judgments. The proposed caching mechanism markedly reduces the number of LLM calls for relevance prediction. Besides improved QPP accuracy, QPP-GenRE demonstrates strong interpretability, allowing QPP errors to be analyzed based on errors in generated relevance judgments.

1.2 Main Contributions

This section summarizes the main contributions of this thesis, from multiple perspectives: theoretical, algorithmic, empirical and resource-related.

1.2.1 Theoretical contributions

1. A formulation of the task of system initiative prediction (SIP) (Chapter 2).

2. A formulation of ranked list truncation (RLT) for dynamic per-query re-ranking depth prediction (Chapter 3).

3. A formulation of query performance prediction (QPP) in the scenario of conversational search (Chapter 4).

1.2.2 Algorithmic contributions

4. A multi-turn system-initiative predictor (MuSIc) that captures structural dependencies between system initiative-taking decisions and other factors (Chapter 2).

5. A SIP-to-clarification transfer learning method, which transfers knowledge gained from the SIP task to the clarification need prediction task (Chapter 2).

6. A framework for SIP-aware hierarchical action prediction, where action prediction depends on SIP (Chapter 2).

7. An algorithm for adapting RLT methods to predict re-ranking depths on a per-query basis (Chapter 3).

8. An algorithm for adapting QPP methods, designed for ad-hoc search, to the conversational search scenarios (Chapter 4).

9. A framework for modeling QPP 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 then predicts different IR evaluation measures based on the predicted relevance judgments (Chapter 5).

10. An approximation strategy for QPP-GenRE to predict IR evaluation measures that consider recall, by predicting relevance for only a limited set of ranked documents and then using their relevance judgments to estimate recall-based measures (Chapter 5).

11. A relevance judgment caching mechanism that increases QPP-GenRE efficiency by reusing previously predicted relevance judgments (Chapter 5).

1.2.3 Empirical contributions

12. An empirical analysis that uncovers structural dependencies between system initiative-taking decisions and various factors in multi-turn conversations (Chapter 2).

13. A systematic performance comparison of MuSIc against other methods for the SIP task (Chapter 2).

14. A visual analysis of MuSIc's interpretability and transparency to explicitly illustrate the structural dependencies it captures.

15. A performance comparison of clarification need prediction using SIP-to-clarification transfer learning against typical clarification need prediction (Chapter 2).

16. A performance comparison of SIP-aware hierarchical action prediction against typical action prediction (Chapter 2).

17. An empirical analysis in the context of LLM-based re-ranking that reveals how fixed re-ranking depths lead to unnecessary computational costs and degrade re-ranking quality, motivating the need for re-ranking depth prediction on a per-query basis (Chapter 3).

18. A comprehensive experiment designed to evaluate the performance of RLT methods in predicting re-ranking depths across various configurations in the context of LLM-based re-ranking (Chapter 3).

19. A comprehensive experiment to demonstrate the performance of QPP methods, originally designed for ad-hoc search, across various configurations of conversational search (Chapter 4).

20. A systematic performance comparison of QPP-GenRE, which uses fine-tuned LLMs for relevance prediction, against other QPP methods in assessing lexical and neural rankers across ad-hoc and conversational search scenarios (Chapter 5).

21. A comprehensive performance comparison of LLMs in relevance judgment prediction under fine-tuning and few-shot prompting settings, evaluating two families of open-source LLMs across different model sizes (Chapter 5).

22. An analysis of QPP-GenRE's performance in predicting a metric considering recall with respect to the number of documents used in the proposed approximation strategy (Chapter 5).

23. An efficiency comparison of QPP-GenRE with the proposed relevance judgment caching mechanism and QPP-GenRE without it (Chapter 5).

24. An analysis of QPP-GenRE's interpretability that shows QPP errors can be analyzed based on errors in generated relevance judgments (Chapter 5).

1.2.4 Resources contributions

25. An open-source repository that provides implementations of MuSIc and SIP-aware hierarchical action prediction, within a unified Python/PyTorch framework (Chapter 2). For further details, visit: `https://github.com/ChuanMeng/SIP`.

26. An open-source repository that offers an implementation of eight RLT methods, along with a full pipeline for adapting them to predicting re-ranking depths, within a unified Python/PyTorch framework (Chapter 3). For further details, visit: `https://github.com/ChuanMeng/RLT4Reranking`.

27. An open-source repository that provides an implementation of seventeen QPP methods, along with a full pipeline for adapting them to conversational search scenarios, within a unified Python/PyTorch framework (Chapter 4). For further details, visit: `https://github.com/ChuanMeng/QPP4CS`.

28. An open-source repository that provides scripts for fine-tuning open-source LLMs to generate relevance judgments, as well as an implementation of QPP-GenRE, within a Python/PyTorch framework (Chapter 5). For further details, visit: `https://github.com/ChuanMeng/QPP-GenRE`.

1.3 Thesis Overview

This section provides an overview of the thesis and offers recommendations for reading directions. The thesis is structured into an introduction chapter, four research chapters grouped into three thematic parts, and a conclusion chapter.

Each research chapter addresses one of the thesis-level research questions introduced in Section 1.1, along with additional chapter-specific research questions. The thesis-level research questions shape the overarching narrative of the work, while the chapter-specific questions focus on the individual contributions within each chapter.

The first chapter, which you are currently reading, introduces the subject of this thesis: optimizing agentic workflows for information access. It also presents the research questions that this thesis seeks to answer, outlines its key contributions, and provides context on its origins.

Part I titled *Mixed-Initiative Strategy Planning* contains Chapter 2, which resolves the issue of a narrow scope of system-initiative actions in predicting the timing of system initiative-taking. Chapter 2 expands the scope of system-initiative actions by defining and modeling a new task, system initiative prediction (SIP).

Part II titled *Ranking Strategy Planning* consists of Chapter 3, which explores dynamic per-query re-ranking depth prediction in LLM-based re-ranking, an area with limited prior research. Chapter 3 conducts a systematic empirical analysis that motivates the need for dynamic per-query re-ranking depths, and explores how to model the prediction in this context.

Part III titled *Ranking Result Reflection* comprises Chapter 4 and Chapter 5, which examine two underexplored areas: QPP in conversational search and LLM-enhanced

QPP. Chapter 4 adapts QPP methods, originally designed for ad-hoc search, to conversational search scenarios, and systematically investigates the performance of existing QPP methods in conversational search. Chapter 5 enhances QPP accuracy by leveraging LLMs' capabilities.

Finally, the thesis concludes with Chapter 6, summarizing the key findings and discussing potential future research directions.

The four research chapters in this thesis are self-contained, allowing them to be read independently. These chapters are based on previously published work, and to maintain fidelity to the original research, alternate versions have not been created. As a result, some notations and conventions may vary slightly across chapters.

1.4 Origins

We list the publications on which the research chapters were based. This thesis is built on four publications [175–177, 180].

Chapter 2 is based on the following paper:

- C. Meng, M. Aliannejadi, and M. de Rijke. System initiative prediction for multi-turn conversational information seeking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1807–1817, 2023.

  CM: Conceptualization, Methodology, Data curation, Software, Formal analysis, Writing – original draft; MA and MdR: Supervision, Conceptualization, Writing - review & editing.

Chapter 3 is based on the following paper:

- C. Meng, N. Arabzadeh, A. Askari, M. Aliannejadi, and M. de Rijke. Ranked list truncation for large language model-based re-ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 141–151, 2024.

  CM: Conceptualization, Methodology, Data curation, Software, Formal analysis, Writing – original draft; NA and AA: Writing - review & editing; MA and MdR: Supervision, Conceptualization, Writing - review & editing.

Chapter 4 is based on the following paper:

- C. Meng, N. Arabzadeh, M. Aliannejadi, and M. de Rijke. Query performance prediction: From ad-hoc to conversational search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2583–2593, 2023.

  CM: Conceptualization, Methodology, Data curation, Software, Formal analysis, Writing – original draft; NA: Software, Formal analysis, Writing - review & editing; MA and MdR: Supervision, Conceptualization, Writing - review & editing.

Chapter 5 is based on the following paper:

- C. Meng, N. Arabzadeh, A. Askari, M. Aliannejadi, and M. de Rijke. Query performance prediction using relevance judgments generated by large language models. *ACM Transactions on Information Systems (TOIS)*, to appear.

  CM: Conceptualization, Methodology, Data curation, Software, Formal analysis, Writing – original draft; NA and AA: Software, Writing - review & editing; MA and MdR: Supervision, Conceptualization, Writing - review & editing.

The writing of the thesis also benefited from work on other publications [1–3, 18–20, 23–25, 146, 159, 170–174, 178, 179, 183, 233]:

- C. Meng, F. Tonolini, F. Mo, N. Aletras, E. Yilmaz, and G. Kazai. Bridging the gap: From ad-hoc to proactive search in conversations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025.
- F. Mo, C. Meng, M. Aliannejadi, and J.-Y. Nie. Conversational search: From fundamentals to frontiers in the LLM era. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025.
- A. Askari, R. Petcu, C. Meng, M. Aliannejadi, A. Abolghasemi, E. Kanoulas, and S. Verberne. SOLID: Self-seeding and multi-intent self-instructing LLMs for generating intent-aware information-seeking dialogs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6390–6410, 2025.
- C. Meng, G. Faggioli, M. Aliannejadi, N. Ferro, and J. Mothe. QPP++ 2025: Query performance prediction and its applications in the era of large language models. In *European Conference on Information Retrieval*, pages 319–325, 2025.
- L. Lu, C. Meng, F. Ravenda, M. Aliannejadi, and F. Crestani. Zero-shot and efficient clarification need prediction in conversational search. In *European Conference on Information Retrieval*, pages 389–404, 2025.
- Z. Abbasiantaeb, C. Meng, L. Azzopardi, and M. Aliannejadi. Improving the reusability of conversational search test collections. In *European Conference on Information Retrieval*, pages 196–213, 2025.
- N. Arabzadeh, C. Meng, M. Aliannejadi, and E. Bagheri. Query performance prediction: Theory, techniques and applications. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 991–994, 2025.
- Z. Abbasiantaeb, C. Meng, L. Azzopardi, and M. Aliannejadi. Can we use large language models to fill relevance judgment holes? In *Joint Proceedings of the 1st Workshop on Evaluation Methodologies, Testbeds and Community for Information Access Research (EMTCIR 2024) and the 1st Workshop on User Modelling in Conversational Information Retrieval (UM-CIR 2024)*, 2024.

- N. Arabzadeh, C. Meng, M. Aliannejadi, and E. Bagheri. Query performance prediction: Techniques and applications in modern information retrieval. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 291–294, 2024.
- S. MacAvaney, A. Roegiest, A. Lipani, A. Parry, B. Engelmann, C. K. Kreutz, C. Meng, E. Frayling, E. Yang, F. Schlatt, G. Faggioli, H. Scells, I. Atanassova, J. Friese, J. Bevendorff, J. Sanz-Cruzado, J. Trippas, K. Pathak, K. Dhole, L. Azzopardi, M. Fröbe, M. Bertin, N. Prasad, S. Zerhoudi, S. Wang, S. Chatterjee, T. Jänich, U. Kruschwitz, X. Wang, and Z. Long. Report on the collab-a-thon at ECIR 2024. *SIGIR Forum*, 58(1):1–11, 2024.
- A. Askari, C. Meng, M. Aliannejadi, Z. Ren, E. Kanoulas, and S. Verberne. Generative retrieval with few-shot indexing. *arXiv preprint arXiv:2408.02152*, 2024.
- C. Meng. Query performance prediction for conversational search and beyond. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3077–3077, 2024.
- N. Arabzadeh, C. Meng, M. Aliannejadi, and E. Bagheri. Query performance prediction: From fundamentals to advanced techniques. In *European Conference on Information Retrieval*, pages 381–388, 2024.
- Z. Abbasiantaeb, C. Meng, D. Rau, A. Krasakis, H. A. Rahmani, and M. Aliannejadi. LLM-based retrieval and generation pipelines for TREC interactive knowledge assistance track (iKAT) 2023. In *Proceedings of the Thirty-Second Text REtrieval Conference (TREC 2023)*, 2023.
- A. Askari, M. Aliannejadi, C. Meng, E. Kanoulas, and S. Verberne. Expand, highlight, generate: RL-driven document generation for passage reranking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10087–10099, 2023.
- C. Meng, M. Aliannejadi, and M. de Rijke. Performance prediction for conversational search using perplexities of query rewrites. In *Proceedings of the QPP++ 2023: Query Performance Prediction and Its Evaluation in New Tasks Workshop co-located with the 45th European Conference on Information Retrieval*, pages 25–28, 2023.
- C. Meng, P. Ren, Z. Chen, Z. Ren, T. Xi, and M. de Rijke. Initiative-aware self-supervised learning for knowledge-grounded conversations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 522–532, 2021.
- W. Sun, C. Meng, Q. Meng, Z. Ren, P. Ren, Z. Chen, and M. de Rijke. Conversations powered by cross-lingual knowledge. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1442–1451, 2021.

- C. Meng, P. Ren, Z. Chen, W. Sun, Z. Ren, Z. Tu, and M. de Rijke. DukeNet: A dual knowledge interaction network for knowledge-grounded conversation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1151–1160, 2020.
- C. Meng, P. Ren, Z. Chen, C. Monz, J. Ma, and M. de Rijke. RefNet: A reference-aware network for background based conversation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8496–8503, 2020.

Part I

Mixed-Initiative Strategy Planning

2

Predicting the Timing of System Initiative

Research on mixed-initiative strategy planning has extensively explored the clarification need prediction task, which focuses on determining when an information access system should ask a clarifying question. However, clarification is just one of many possible system-initiative actions, and comparatively little attention has been given to predicting the timing of other actions. To address this limitation, this chapter introduces the system initiative prediction (SIP) task, which expands beyond the narrow focus of clarification need prediction. SIP aims to predict the timing of high-level system-initiative behaviours that cover a broad range of specific system-initiative actions. By operating at a higher level, SIP has the potential to enhance downstream tasks in mixed-initiative strategy planning. Given these potential advantages, it is crucial to explore how to effectively model SIP and assess its impact on downstream applications. Thus, this chapter seeks to answer the following thesis-level research question:

RQ1 How can we effectively model system initiative prediction (SIP), and how does this prediction benefit downstream tasks?

2.1 Introduction

An essential part of conversational information seeking (CIS) is to identify the right moment for a CIS system to take the initiative [27, 286], given that system initiative-taking risks frustrating the user and hurting the user experience [263, 264, 286, 305, 306]. Various system-initiative actions can be taken by a CIS system to take the initiative, e.g., asking a clarifying question or requesting feedback [246, 250]. Existing work has extensively studied the clarification need prediction task, that is, predicting when to ask a clarifying question in an information-seeking conversation [9, 11, 16, 263, 264, 273]. However, as shown in Figure 2.1, asking a clarifying question is only one of several possible system-initiative actions [10, 29, 286].

Task and motivation. We define *system initiative prediction* (SIP) task, which is to predict whether the CIS system should take the initiative at the next turn in an information-seeking conversation. To the best of our knowledge, no existing studies

This chapter was published as C. Meng, M. Aliannejadi, and M. de Rijke. System initiative prediction for multi-turn conversational information seeking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1807–1817, 2023.

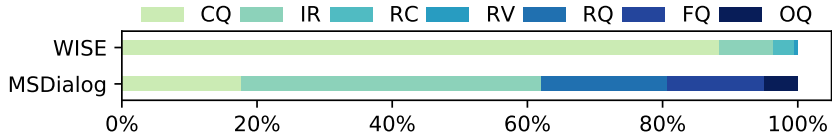


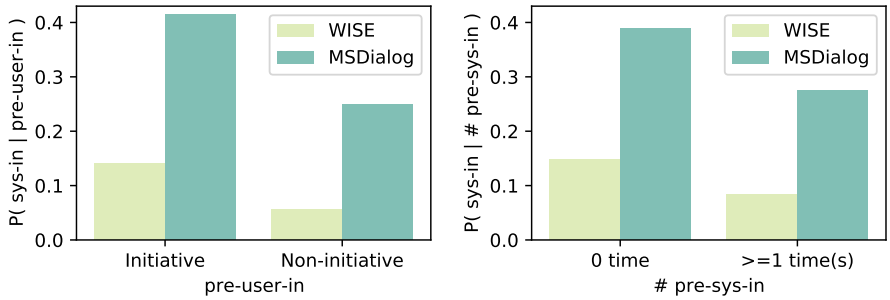
Figure 2.1: Distribution of system-initiative actions in two realistic CIS training datasets, WISE and MSDialog. CQ: clarifying question (called *clarify* in WISE); IR: information request (called *request* in WISE); RV: revise; RC: recommendation (ask users if they would like something); OQ: original question; RQ: repeat question; and FQ: follow up question.

explicitly model this problem. Compared to clarification need prediction, SIP is a high-level decision. SIP has three benefits for CIS systems: (i) SIP can improve the controllability of the overall initiative level of the system to balance utility and user experience [228]. (ii) SIP can enable knowledge sharing among various system-initiative actions; the shared knowledge learned through SIP can be transferred to improve the prediction of a specific system-initiative action by transfer learning, e.g., by fine-tuning a model, pre-trained on SIP, on clarification need prediction. And (iii) SIP can boost the prediction performance of downstream tasks that depend on SIP by narrowing their decision space; e.g., in action prediction, where the system selects an action from a large action space, certain actions, such as *requesting feedback*, are performed only if the SIP result is initiative. One could argue that existing action prediction methods [280] are sufficiently effective for SIP. However, our experiments show that using action prediction methods for SIP leads to *suboptimal* results, but conversely, SIP significantly improves downstream action prediction.

Our empirical analysis of two CIS datasets [200, 207] reveals that a system's initiative-taking decision at the next turn is not isolated but depends on the user's previous initiative-taking decision. Figure 2.2a shows that the system is more likely to take the initiative immediately after the user has taken the initiative in a conversation; thus, capturing the *dependencies between adjacent user–system initiative-taking decisions* is critical for modeling SIP.

A natural way to capture such structural dependencies is to use probabilistic graphical models, such as conditional random fields (CRFs) [128]. We propose to use linear-chain CRFs [128, 236] to model SIP for three reasons: (i) they have been shown to be effective in capturing dependencies between adjacent output decisions [236]; (ii) linear-chain CRFs for SIP can guarantee the best initiative-taking decision at the next turn by decoding the optimal sequence of initiative-taking decisions in context ($1:T-1$ in Figure 2.3a) and the next turn (T in Figure 2.3a), instead of outputting the decision at the next turn independently [128, 236]; and (iii) due to CRFs' graphical nature, they have been shown to exhibit better interpretability and transparency than other methods [89, 125], such as emergent large language models (LLMs) [49, 245, 297].

Challenges. When adopting linear-chain CRFs to the SIP task we face two challenges: (i) They cannot be directly applied to SIP because we face an *input-incomplete* sequence labeling problem. Linear-chain CRFs are designed for sequence labeling problems that have a one-to-one correspondence between input observations and output decisions. As



(a) (b)

Figure 2.2: The probability of system initiative-taking (sys-in) conditioned on the user's initiative-taking decision at the preceding turn (pre-user-in) and the number of times the system has taken the initiative (# pre-sys-in) on the WISE and MSDialog training sets.

shown in Figure 2.3a, to output initiative-taking decisions in context and at the next turn, linear-chain CRFs need to be given a complete input sequence of utterances in context and at the next turn. However, given the nature of SIP, as shown in Figure 2.3b, the system utterance at the next turn is unobservable, leading to an *input-incomplete sequence labeling* problem. And (ii) linear-chain CRFs do not explicitly model *multi–turn features*. Our empirical analysis shows that an initiative-taking decision depends on multi-turn features. We define a multi-turn feature as a variable that varies across turns. Consider, e.g., *the number of times the system has taken the initiative*; Figure 2.2b shows that a system is much less likely to take the initiative once again if it has already taken the initiative before. But linear-chain CRFs do not consider this feature as it is beyond the dependency between adjacent initiative decisions.

Approach. To address the challenges, we cast SIP as an *input-incomplete* sequence labeling problem and propose a *multi-turn system initiative predictor* (MuSIc). We propose (i) *prior-posterior inter-utterance encoders* to adapt linear-chain CRFs to the *input-incomplete* sequence labeling problem and eliminate the need to be given the unobservable system utterance, and (ii) a *multi-turn feature-aware conditional random field* (CRF) layer to explicitly capture *the impact of multi-turn features on an initiative–taking decision* by conditioning the dependencies between adjacent initiative-taking decisions on multi-turn features. MuSIc can use an arbitrary number of multi-turn features; we consider three essential ones: (i) role transition direction, (ii) the number of times the system has taken the initiative, and (iii) the distance to the last system initiative turn.

Experiments. We annotate the initiative-taking decision at each turn on two multi-turn CIS datasets, WISE [207] and MSDialog [199]. Experiments on both datasets show that MuSIc achieves state-of-the-art performance on SIP, outperforming strong clarification need prediction, action prediction, and LLM-based (LLaMA [245]) baselines (see Section 2.6.1). We get two more insights: (i) LLMs do not show promising performance on SIP where scaling up LLMs is not an effective way to solve SIP; and (ii) probabilistic graphical modeling is still competitive and effective for this task and

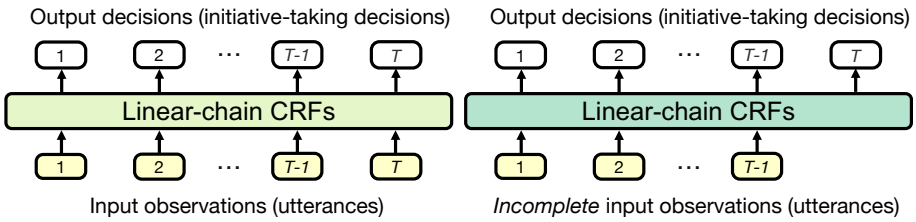


Figure 2.3: A comparison between a sequence labeling problem (a) and an *input-incomplete* sequence labeling problem (b). $1:T$ denote turn numbers and T is the next system turn.

it should not be ignored in the era of LLMs. Furthermore, a visual analysis indicates that the transition matrices learned through the MuSIc exhibit meaningful transition patterns and explicitly show how MuSIc models the dependencies, showing great interpretability and transparency (see Section 2.6.2). We explore how SIP can enhance two key downstream tasks (see Section 2.6.4): clarification need prediction and action prediction. For clarification need prediction, we propose a *SIP-to-clarification transfer learning* approach, which fine-tunes a model (e.g., MuSIc) pre-trained on SIP on the clarification need prediction task. We found that by using this approach, MuSIc achieves the state-of-the-art clarification need prediction performance on ClariQ [9, 11], indicating that the knowledge shared among various system-initiative actions learned through SIP can be used to improve the prediction of a specific system-initiative action. For action prediction, we construct a *SIP-aware action prediction* framework where action prediction is fed with SIP results returned by MuSIc. The action prediction performance is significantly improved, indicating the effectiveness of SIP in benefiting downstream tasks.

Contributions. Our main contributions in this chapter are as follows:

- We introduce the task of *system initiative prediction* (SIP) for CIS, which has not been explicitly modeled in prior work.
- We propose a *multi-turn system-initiative predictor* (MuSIc), which formalizes SIP as an *input-incomplete* sequence labeling problem and jointly considers *dependencies between adjacent user–system initiative-taking decisions* and *the impact of multi-turn features on an initiative-taking decision*.
- We conduct experiments on two multi-turn CIS datasets, showing state-of-the-art performance of MuSIc on SIP.
- We propose a SIP-to-clarification transfer learning approach, which fine-tunes a model, pre-trained on SIP, on the clarification need prediction task. By using this approach, MuSIc achieves state-of-the-art performance on clarification need prediction, setting a new benchmark on the ClariQ dataset.
- We propose a SIP-aware action prediction framework, where downstream action prediction depends on SIP outcomes. By using MuSIc-predicted SIP results, this

hierarchical framework significantly enhances the performance of action prediction.

2.2 Related Work

This chapter builds on two strands of research: conversational information seeking (CIS) and conditional random fields (CRFs).

2.2.1 Conversational information seeking

We focus on modeling mixed-initiative CIS systems [59, 92, 171–173, 286].

Mixed initiative. Mixed initiative is a key aspect in CIS [203]: the user and system can both take the initiative at different times in a conversation. Mixed-initiative CIS systems can ask clarifying questions [7, 11, 218, 284], elicit user preferences [204, 219], ask for feedback [246, 250], initiate a conversation [258] and so on. Existing work focuses on when a CIS system should take the initiative [27] and response generation/selection given a decided system-initiative action [7, 43, 218, 265, 284]. We focus on the former. In this direction, Avula et al. [27] run a user study to investigate it. The user study into when to take the initiative [27] has been explored. This chapter differs as we explicitly model when to take the initiative. Besides, much work has studied the prediction of when to perform a specific system-initiative action, asking a clarifying question (a.k.a. clarification need prediction) [9, 11, 16, 263, 264, 273].

Clarification need prediction. Zou et al. [305, 306] show that asking a clarifying question is not always necessary, and inappropriate requests for clarification can hurt user experience. Xu et al. [273] propose a binary classification model to identify whether clarification is needed given the conversational context. Aliannejadi et al. [9, 11] fine-tune pre-trained language models fed with user queries to return a clarification need score. Wang and Ai [263, 264] propose a binary classification model that further takes into account clarifying question and answer candidates returned by retrieval models. Arabzadeh et al. [16] utilize the coherency of items retrieved for the user query: the more coherent the retrieved items are, the less ambiguous the query is, and the need for clarification decreases.

This chapter differs from these studies, as SIP covers a broader range of system-initiative actions, while these studies are limited to one initiative type (i.e., asking a clarifying question). As we have seen in Figure 2.1, real-world CIS datasets include diverse system-initiative actions, which are neglected in the studies listed above.

System action prediction. Radlinski and Craswell [203] define a system action space and emphasize the need for system action prediction in CIS, i.e., a CIS system should predict an appropriate action from an action space at the right time. Azzopardi et al. [28] define a more detailed taxonomy of user/system actions in CIS. Ren et al. [207] propose a model that can predict one action per turn. Schneider et al. [217] conduct user study to reveal action flow patterns in CIS. Ghosh et al. [95] first identify the user action used in the previous user utterance and then use that to benefit the system action prediction. In this chapter, we are concerned with the more challenging multi-action system action prediction task, i.e., the system performs multiple actions concurrently per turn [291]. Beyond CIS, multi-action system action prediction has been well

studied for task-oriented conversations, where it is typically formulated as a multi-label classification [107, 138, 269] or sequence generation problem [117, 138, 225, 260]. Ye et al. [280] propose a sequence generation-based method, called Co-Gen, achieving leading performance in terms of response generation and action prediction.

This chapter differs from action prediction because SIP is a higher-level decision on which the action prediction depends.

2.2.2 Linear-chain conditional random fields

Linear-chain CRFs are discriminative probabilistic graphical models for sequence labeling problems that assign output decisions to all of the observations in a sequence jointly [128]. The output decisions are arranged in a sequence/linear chain where adjacent output decisions are dependent according to the first-order Markov assumption, enabling linear-chain CRFs to effectively capture dependencies between adjacent output decisions [236]. We focus on neural linear-chain CRFs [112, 113], where parameters can be trained end-to-end. They have been widely used for sequence labeling tasks, e.g., POS tagging [112], named entity recognition [112, 129] and dialogue act recognition [46, 68, 126, 205, 221].

None of the work listed above can be directly applied to SIP due to the *input-incomplete* sequence labeling problem. Another line of research captures the dependencies between adjacent output decisions by dynamically generating transition matrices [100, 113, 231, 235]. MuSIc differs as it explicitly incorporates multi-turn features into the adjacent dependencies. While some work [42, 221] injects features (e.g., emotion shifts) into the adjacent dependencies for sequence labeling, MuSIc is for *input-incomplete* sequence labeling and considers CIS-specific features that have not been studied yet.

2.3 Task Definition

Consider an information-seeking conversation $X = (x_1, x_2, \ldots, x_{|X|-1}, x_{|X|})$ with a sequence of $|X|$ utterances, where x is an utterance uttered by either a user or system. The conversation X comes with a sequence of ground-truth initiative-taking decisions $Y = (y_1, y_2, \ldots, y_{|X|-1}, y_{|X|})$, i.e., each utterance x in the conversation has a corresponding initiative-taking decision $y \in \{\textit{Initiative}, \textit{Non-initiative}\}$. Given the context $X_{1:T-1} = (x_1, x_2, \ldots, x_{T-1})$, where $T-1$ is a user turn, the *system initiative prediction* (SIP) task is to predict the system's initiative-taking decision y_T at the next turn T. We formulate SIP as an *input-incomplete* sequence labeling problem: we model the conditional probability $P(Y_{1:T} \mid X_{1:T-1})$ of the sequence of initiative-taking decisions in the context $Y_{1:T-1}$ and at the next turn y_T given the sequence $X_{1:T-1}$ of utterances in the context. Only the system's initiative-taking decision y_T at the next turn T is used for evaluation.

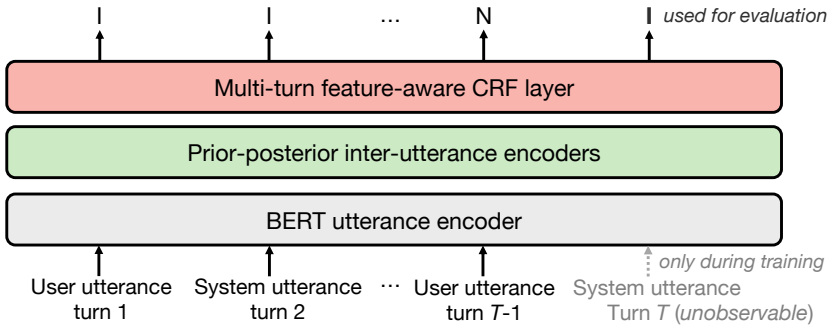


Figure 2.4: Overview of MuSIc. Its target is to predict the optimal sequence of initiative-taking decisions in the context $1:T-1$ and at the next turn T given the utterances over turns $1:T-1$. I/N at the top denotes *Initiative*/*Non-initiative*.

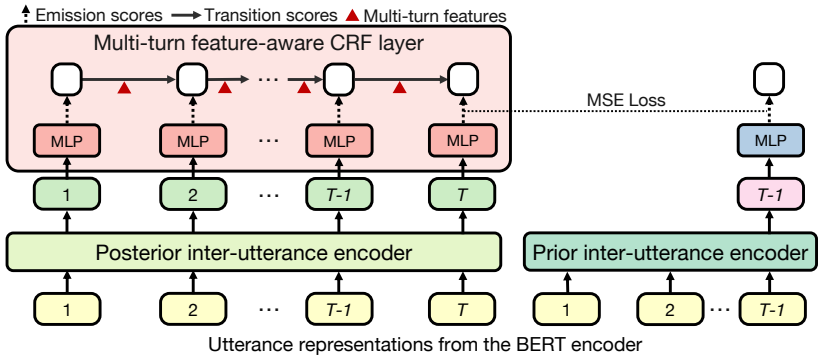
2.4 Method

2.4.1 Limitations of linear-chain conditional random fields

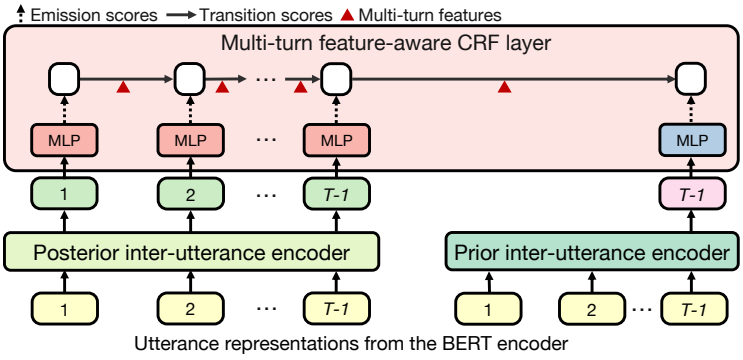
Linear-chain CRFs predict a sequence of output decisions based on emission and transition scores (see [112, 129] for details), and have two main limitations when applied "as is" to SIP: (i) They model $P(Y_{1:T} \mid X_{1:T})$ to output the sequence $Y_{1:T}$: they use the sequence $X_{1:T}$ of utterances in the context and at the next turn to calculate emissions scores over $\{$*Initiative*, *Non-initiative*$\}$ over turns $1:T$; there is a one-to-one correspondence between $X_{1:T}$ and emission scores over turns $1:T$. However, x_T, the utterance at the next turn, is unobservable for SIP (see Figure 2.3b), leading to the absence of the emission scores at turn T. (ii) They use a transition matrix that contains transition scores from one initiative-taking decision to itself (e.g., *Initiative* to *Initiative*) or the other (e.g., *Initiative* to *Non-initiative*) to capture dependencies between adjacent initiative-taking decisions. An initiative-taking decision y_{t+1} is also impacted by a multi-turn feature $s_{t:t+1}$ that changes across turns, e.g., *the number of times the system has taken the initiative* (see Figure 2.2b). However, the transition matrix is unique and shared across all turns; thus, the transition scores cannot be adjusted across turns to capture the impact of a multi-turn feature $s_{t:t+1}$ effectively.

2.4.2 Overview of MuSIc

We propose MuSIc for SIP, which consists of three parts: (i) a *BERT utterance encoder*, (ii) *prior-posterior inter-utterance encoders*, and (iii) a *multi-turn feature-aware CRF layer*. See Figure 2.4. The *BERT utterance encoder* is used to encode each utterance into a latent representation. *Prior-posterior inter-utterance encoders* enable MuSIc to model the *input-incomplete* sequence labeling by approximating the absent emission scores at turn T. We model $P(Y_{1:T} \mid X_{1:T})$ during training (see Figure 2.5a) as we can access the unobservable system utterance x_T at the next turn T; we pass $X_{1:T}$ through the BERT encoder and a posterior inter-utterance encoder to calculate emission scores over turns $1:T$; we define them as posterior emission scores. Similarly, we



(a) Training



(b) Inference

Figure 2.5: *Prior-posterior inter-utterance encoders* and *multi-turn feature-aware CRF layer* during (a) training and (b) inference. The system utterance at the next turn T can be accessed by the posterior inter-utterance encoder *only* during training.

pass $X_{1:T-1}$ through BERT and a prior inter-utterance encoder; we use the output of the prior inter-utterance encoder to calculate prior emission scores that are forced to approximate the posterior emission scores at T via an MSE loss. During inference (see Figure 2.5b), we model $P(Y_{1:T} \mid X_{1:T-1})$ and regard the approximate (prior) emission scores as the absent emission scores at turn T, eliminating the need to be given the unobservable system utterance x_T. The *multi-turn feature-aware CRF layer* incorporates three multi-turn features and conditions transition scores (dependencies) between adjacent initiative-taking decisions on multi-turn features. We extend the single transition matrix in linear-chain CRFs to multiple ones, corresponding to different multi-turn features. For a pair of adjacent initiative-taking decisions between turn t and $t+1$, we adjust the transition score between them by selecting the transition matrix corresponding to the multi-turn features from turn t to $t+1$.

2.4.3 BERT utterance encoding

We use a BERT encoder [71] to encode an utterance x_t ($t = 1, \ldots, T$ during training, $t = 1, \ldots, T-1$ during inference) into an utterance representation $\mathbf{H}^{x_t} \in \mathbb{R}^{|x_t| \times d}$, after which an average pooling operation [40] is used to get a condensed representation $\mathbf{h}^{x_t} \in \mathbb{R}^{1 \times d}$, where $|x_t|$ and d denote the number of tokens in x_t and the hidden size, respectively.

2.4.4 Prior-posterior inter-utterance encoding

We have a *prior inter-utterance encoder* that takes as input $\{\mathbf{h}^{x_t}\}_{t=1}^{T-1}$, returning prior utterance representations $\{\mathbf{h}_{pri}^{x_t}\}_{t=1}^{T-1}$, as shown in Figure 2.5. Also, we have a *posterior inter-utterance encoder* that takes as input $\{\mathbf{h}^{x_t}\}_{t=1}^{T}$ during training ($\{\mathbf{h}^{x_t}\}_{t=1}^{T-1}$ during inference), outputting posterior utterance representations $\{\mathbf{h}_{pos}^{x_t}\}_{t=1}^{T}$ during training ($\{\mathbf{h}_{pos}^{x_t}\}_{t=1}^{T-1}$ during inference).[1]

2.4.5 Multi-turn feature-aware conditional random field layer

During training, we feed the unobservable system utterance x_T to MuSIc and model the conditional probability $P(Y_{1:T} \mid X_{1:T})$ of the sequence $Y_{1:T}$ of initiative-taking decisions in the context and at the next turn given the sequence $X_{1:T}$ of utterances in the context and at the next turn.

We consider three multi-turn features $\mathbf{S} = \{s_{t:t+1}^r, s_{t:t+1}^n, s_{t:t+1}^d\}_{t=1}^{T-1}$ as additional input:

(1) $s_{t:t+1}^r$ represents the *role transition direction* from turn t to $t+1$, i.e., $s_{t:t+1}^r = u2s/s2u$ means that the role transition is from the user to the system/the system to the user from turn t to $t+1$.

(2) Given $s_{t:t+1}^r = u2s$ ("user to system"),[2] $s_{t:t+1}^n$ represents *the number of times the system takes the initiative* before the next system turn at $t+1$. Table 2.1 shows that the average number of system initiative utterances in a conversation in training sets is less than 1. To make full use of the sparse training data, we only consider the cases $s_{t:t+1}^n = 0$ and > 0, which means that the system has not taken the initiative and has taken the initiative once or more before the next system turn at $t+1$, respectively.

(3) Given $s_{t:t+1}^r = u2s$ (again, "user to system") and $s_{t:t+1}^n > 0$, $s_{t:t+1}^d$ represents *the distance to the last system initiative turn* from the next system turn at $t+1$. Similarly, to make full use of the sparse data, we only consider $s_{t:t+1}^d = 2$ and > 2,[3] which means that the distance to the last system initiative turn from the next system turn at $t+1$ is 2 and more than 2 turns, respectively.

[1] We implement inter-utterance encoders by BiLSTMs, which got better performance than Transformers in our preliminary experiments.

[2] We consider other features only given "user to system" for simplicity;"user to system" is more critical as the role transition from turn $T-1$ to T is always from "user to system".

[3] We also experimented with more fine-grained cases, such as $s_{t:t+1}^n = 1, 2, 3, 4$ and $s_{t:t+1}^d = 4, 6, 8$, but no further improvements were obtained.

After considering the three multi-turn features, MuSIc models:

$$P(Y_{1:T} \mid X_{1:T}, \mathbf{S}) = \frac{\exp(\psi(X_{1:T}, Y_{1:T}, \mathbf{S}))}{\sum_{\tilde{Y}_{1:T}} \exp(\psi(X_{1:T}, \tilde{Y}_{1:T}, \mathbf{S}))},$$
$$\psi(Y_{1:T}, X_{1:T}, \mathbf{S}) = \sum_{t=1}^{T} e(y_t, X_{1:T}) + \sum_{t=1}^{T-1} g(y_t, y_{t+1}, s^r_{t:t+1}, s^n_{t:t+1}, s^d_{t:t+1}), \tag{2.1}$$

where $\tilde{Y}_{1:T}$ denotes one of all possible sequences of initiative-taking decisions, $e(y_t, X_{1:T})$ is the emission scoring function to calculate the posterior emission scores based on $X_{1:T}$, and $g(y_t, y_{t+1}, s^r_{t:t+1}, s^n_{t:t+1}, s^d_{t:t+1})$ is the transition score function to calculate the transition scores conditioned on multi-turn features $\mathbf{S}$.

Computing emission scores. $e(y_t, X_{1:T})$ calculates the posterior emission scores $\{\mathbf{e}^{x_t}_{pos}\}^T_{t=1}$ based on the posterior utterance representations $\{\mathbf{h}^{x_t}_{pos}\}^T_{t=1}$; see Figure 2.5a. The calculation at each turn is modeled as:

$$e(y_t, X_{1:T}) = \mathbf{e}^{x_t, y_t}_{pos} \in \mathbb{R}^{1\times 1}$$
$$\mathbf{e}^{x_t}_{pos} = \mathrm{MLP}(\mathbf{h}^{x_t}_{pos}) \in \mathbb{R}^{1\times 2}, \tag{2.2}$$

where $t = 1, 2, \ldots, T$, $\mathbf{e}^{x_t}_{pos} \in \mathbb{R}^{1\times 2}$ are posterior emission scores over {*Initiative*, *Non-initiative*}, and $\mathrm{MLP}(\cdot)$ denotes a multilayer perceptron (MLP). In parallel, we calculate the prior emission scores $\mathbf{e}^{x_{T-1}}_{pri} \in \mathbb{R}^{1\times 2}$ based on the last output (at turn $T-1$) of the prior inter-utterance encoder $\mathbf{h}^{x_{T-1}}_{pri}$ (see Figure 2.5a):

$$\mathbf{e}^{x_{T-1}}_{pri} = \mathrm{MLP}(\mathbf{h}^{x_{T-1}}_{pri}) \in \mathbb{R}^{1\times 2}. \tag{2.3}$$

The prior emission scores $\mathbf{e}^{x_{T-1}}_{pri} \in \mathbb{R}^{1\times 2}$ would learn to approximate the posterior emission scores $\mathbf{e}^{x_T}_{pos} \in \mathbb{R}^{1\times 2}$ at turn T (see Figure 2.5a and Equation 2.8). The parameters of the MLP in Equation 2.2 and Equation 2.3 are not shared.

Computing transition scores. Linear-chain CRFs do not condition a transition score on any multi-turn features:

$$g(y_t, y_{t+1}) = G_{y_t, y_{t+1}} \in \mathbb{R}^{1\times 1}, \tag{2.4}$$

where $G \in \mathbb{R}^{2\times 2}$ is a transition matrix shared across all turns, and $G_{y_t, y_{t+1}}$ is the transition score from the decision y_t to y_{t+1}. Our transition scoring function, denoted as $g(y_t, y_{t+1}, s^r_{t:t+1}, s^n_{t:t+1}, s^d_{t:t+1})$, does condition the computation of the transition scores between adjacent initiative-taking decisions on the multi-turn features $s^r_{t:t+1}$, $s^n_{t:t+1}$, and $s^d_{t:t+1}$. We define separate transition matrices corresponding to different combinations of multi-turn features. For a pair of adjacent initiative-taking decisions between turn t and $t+1$, we select the transition matrix corresponding to the multi-turn features from turn t to $t+1$. If the transition score is only conditioned on the multi-turn feature *role transition direction* $s^r_{t:t+1}$, it is calculated as:

$$g(y_t, y_{t+1}, s^r_{t:t+1}) = (1 - I(s^r_{t:t+1})) \cdot \mathbf{G}^{s2u}_{y_t, y_{t+1}} + I(s^r_{t:t+1}) \cdot \mathbf{G}^{u2s}_{y_t, y_{t+1}}, \tag{2.5}$$

where $I(s^r_{t:t+1})$ is an indicator function that equals 1 if $s^r_{t:t+1} = u2s$ and 0 otherwise, and $\mathbf{G}^{s2u} \in \mathbb{R}^{2\times2}$ and $\mathbf{G}^{u2s} \in \mathbb{R}^{2\times2}$ are transition matrices corresponding to "from system to user" and "from user to system," respectively.

Given $s^r_{t:t+1} = u2s$, if the transition score is further conditioned on the feature $s^n_{t:t+1}$, *the number of times the system takes the initiative* before the next system turn at $t+1$, it is calculated as:

$$\begin{aligned} g(y_t, y_{t+1}, s^r_{t:t+1}, s^n_{t:t+1})) = (1 - I(s^r_{t:t+1})) \cdot \mathbf{G}^{s2u}_{y_t,y_{t+1}} + \\ I(s^r_{t:t+1}) \cdot [(1 - I(s^n_{t:t+1})) \cdot \mathbf{G}^{u2s,n=0}_{y_t,y_{t+1}} + I(s^n_{t:t+1}) \cdot \mathbf{G}^{u2s,n>0}_{y_t,y_{t+1}}], \end{aligned} \tag{2.6}$$

where $I(s^n_{t:t+1})$ is an indicator function that equals 1 if $s^n_{t:t+1} > 0$ and 0 otherwise, and $\mathbf{G}^{u2s,n=0} \in \mathbb{R}^{2\times2}$ and $\mathbf{G}^{u2s,n>0} \in \mathbb{R}^{2\times2}$ are transition matrices corresponding to "the system has not take the initiative" and "the system has taken the initiative once or more" before the next system turn at $t+1$, respectively.

Given $s^r_{t:t+1} = u2s$ and $s^n_{t:t+1} > 0$, if the transition score is further conditioned on the feature $s^d_{t:t+1}$, *the distance to the last system's initiative turn* from the next system turn at $t+1$, it is calculated as:

$$\begin{aligned} g(y_t, y_{t+1}, s^r_{t:t+1}, s^n_{t:t+1}, s^d_{t:t+1})) = (1 - I(s^r_{t:t+1})) \cdot \mathbf{G}^{s2u}_{y_t,y_{t+1}} + \\ I(s^r_{t:t+1}) \cdot \{(1 - I(s^n_{t:t+1})) \cdot \mathbf{G}^{u2s,n=0}_{y_t,y_{t+1}} + \\ I(s^n_{t:t+1}) \cdot [(1 - I(s^d_{t:t+1})) \cdot \mathbf{G}^{u2s,n>0,d=2}_{y_t,y_{t+1}} + \\ I(s^d_{t:t+1}) \cdot \mathbf{G}^{u2s,n>0,d>2}_{y_t,y_{t+1}}]\}, \end{aligned} \tag{2.7}$$

where $I(s^d_{t:t+1})$ is an indicator function that equals 1 if $s^d_{t:t+1} > 2$ and 0 otherwise, and $\mathbf{G}^{u2s,n>0,d=2} \in \mathbb{R}^{2\times2}$ and $\mathbf{G}^{u2s,n>0,d>2} \in \mathbb{R}^{2\times2}$ are transition matrices for "the distance to the last system's initiative turn is 2 turns" and "the distance to the last system's initiative turn is more than 2 turns" from the next system turn at $t+1$, respectively.

Training objectives. Our final loss function is defined as $\mathcal{L} = \mathcal{L}_{crf} + \mathcal{L}_{mse}$. We not only minimize the negative log-likelihood of the sequence $Y_{1:T}$ of ground-truth initiative-taking decisions in the context and at the next turn, but also force $\mathbf{e}^{x_{T-1}}_{pri}$ to learn to approximate $\mathbf{e}^{x_T}_{pos}$ via an MSE loss (see Figure 2.5a):

$$\begin{aligned} \mathcal{L}_{crf} &= -\log P(Y_{1:T} \mid X_{1:T}, \mathbf{S}) \\ \mathcal{L}_{mse} &= -(\mathbf{e}^{x_{T-1}}_{pri} - \mathbf{e}^{x_T}_{pos})^2. \end{aligned} \tag{2.8}$$

Inference phase. MuSIc models the conditional probability $P(\tilde{Y}_{1:T} \mid X_{1:T-1}, \mathbf{S})$ of a possible sequence $\tilde{Y}_{1:T}$ of initiative-taking decisions in the context $(1:T-1)$ and at the next turn T only given the sequence $X_{1:T-1}$ of utterances in the context (see

Figure 2.5b):

$$P(\tilde{Y}_{1:T} \mid X_{1:T-1}, \mathbf{S}) = \frac{\exp(\psi(X_{1:T-1}, \tilde{Y}_{1:T}, \mathbf{S}))}{\sum_{\tilde{Y}_{1:T}} \exp(\psi(X_{1:T-1}, \tilde{Y}_{1:T}, \mathbf{S}))},$$
$$\psi(\tilde{Y}_{1:T}, X_{1:T-1}, \mathbf{S}) = \sum_{t=1}^{T} e(\tilde{y}_t, X_{1:T-1}) + \sum_{t=1}^{T-1} g(\tilde{y}_t, \tilde{y}_{t+1}, s^r_{t:t+1}, s^n_{t:t+1}, s^d_{t:t+1}), \tag{2.9}$$

where $e(\tilde{y}_t, X_{1:T-1}) = \mathbf{e}_{pri}^{x_{T-1}, \tilde{y}_T}$ if $t = T$ and $\mathbf{e}_{pos}^{x_t, \tilde{y}_t}$ otherwise (see Figure 2.5b). The optimal sequence $Y^*_{1:T}$ of initiative-taking decisions in context and at the next turn is decoded by the Viterbi algorithm [254]:

$$Y^*_{1:T} = \arg\max_{\tilde{Y}_{1:T}} P(\tilde{Y}_{1:T} \mid X_{1:T-1}, \mathbf{S}). \tag{2.10}$$

2.5 Experimental Setup

2.5.1 Research questions

This chapter expands on the thesis-level research question **RQ1** by examining the following chapter-specific research questions:

RQ1.1 To what extent does MuSIc improve performance on the SIP task compared to state-the-art baselines?

RQ1.2 What is the effect of multi-turn features on the performance of MuSIc?

RQ1.3 To what extent does knowledge shared among various system-initiative actions learned through SIP benefit the clarification need prediction task?

RQ1.4 To what extent does the SIP task benefit the downstream action prediction task?

2.5.2 Datasets

We consider two multi-turn CIS datasets with annotations of actions for utterances, WISE [207] and MSDialog [199, 200, 278]. Based on the action annotations, we annotate the initiative-taking decision for each utterance. WISE is collected through crowdsourcing; it consists of mixed-initiative conversations between two workers playing the role of user and system. All utterances are annotated with actions. We use the data split from [207]. MSDialog consists of mixed-initiative conversations between users who ask for technical help and expert users or staff (i.e., system) who help to solve problems. This dataset has two versions: the complete set and a labeled subset. Each utterance in the labeled subset is annotated with actions; we use the data split of the labeled subset from [200].

Table 2.1: Statistics of the WISE and MSDialog datasets after preprocessing; conv. is short for "conversation."

	WISE			MSDialog		
	train	valid	test	train	valid	test
# conversations	705	200	1,000	1,760	220	219
# utterances	12,184	3,811	18,828	6,305	752	747
# system utterances	5,949	1,868	9,246	2,938	352	354
# system initiatives	691	324	1,457	1,085	131	143
Max. # turns/conversation	38	38	42	10	10	10
Avg. # turns/conversation	17.28	19.06	18.83	3.58	3.42	3.41
Max. # actions/system turn	3	2	3	6	6	7
Avg. # actions/system turn	1.02	1.02	1.02	1.67	1.77	1.80
Avg. # system initiatives/conv.	0.98	1.62	1.46	0.62	0.60	0.65
Avg. # clarifying questions/conv.	0.87	1.23	1.17	0.15	0.18	0.15

2.5.3 Pre-processing

Following [251, 263, 264], we merge consecutive utterances from either the user or system into one utterance by concatenation; their corresponding actions are merged by a union operation too. See Table 2.1 for the statistics of the datasets. The average numbers of turns in both datasets are less than the numbers in the original papers [200, 207] due to the merging operation.

2.5.4 Annotation of initiative-taking decision labels

For both datasets, we derive the initiative annotations by mapping the manual annotations of actions to initiative or non-initiative labels. An utterance is annotated as *initiative* if it is annotated with any of the actions showing initiative[4] and *non-initiative* otherwise.

2.5.5 Baselines

We compare MuSIc with recently proposed LLM-based baselines, and three other groups of state-of-the-art baselines for the SIP task: (i) clarification need prediction, (ii) system action prediction, and (iii) linear-chain CRF-based methods.

Regarding LLM-based baselines, we consider **LLaMA-7B/13B/33B/65B** [245] using in-context learning [35, 74] as the LLM-based baselines. Mao et al. [166] prompt LLMs for conversational query rewriting and we adapt their designed prompt to SIP. We prepend the SIP task instruction at the beginning of the prompt, followed by two groups of demonstrations: (i) a few complete conversations randomly sampled from the training set, and (ii) utterances in the context $X_{1:T-1}$ prior to the next turn T. Ground-truth

[4]The WISE dataset has different taxonomies for user and system actions; system actions showing initiative have been shown in Figure 2.1; user actions showing initiative are *reveal*, *request*, and *revise*. MSDialog has the same taxonomy for user and system actions; actions showing initiative have been shown in Figure 2.1.

system-initiative decisions are prepended to the corresponding system utterances in the demonstrations. Given the prompt, LLaMA generates the system-initiative decision at the next system turn T. WISE is a Chinese language dataset; however, the original LLaMA has a limited ability to encode and decode Chinese text [57]. Cui et al. [57] release Chinese-LLaMA-Plus-7B and -13B at the time of writing. These LLaMA variants use the extended Chinese vocabulary and are further trained on Chinese data. We report the performance of both [57] on WISE.

We train and test two clarification need prediction models on SIP:

- **CtxPred (BERT)** uses a BERT encoder to encode the context and predict whether to take the initiative at the next turn [9, 11, 273].
- **Risk-aware Conversational Search agent with Q-learning (RCSQ)** is fed with the context, clarifying question and answer candidates returned by retrievers, and is trained with a user simulator by reinforcement learning [263, 264]. To adapt it to SIP,[5] we replace the clarifying question and answer candidates with initiative and non-initiative system utterance candidates retrieved by bi-encoders;[6] we also replace Q-learning with supervised learning using the annotations of initiative-taking decisions. We follow the original implementation for the rest.

We also compare MuSIc with the state-of-art system action prediction method **Co-Gen** [280]. Co-Gen generates actions and responses concurrently — the two generators share a common latent space. We consider two variants of Co-Gen:[7]

- **Co-Gen (action prediction)** is trained with action and response generation; the model outputs actions based on which we derive initiative-taking decisions using our action-initiative mapping.
- **Co-Gen (SIP)** is trained with SIP and response generation; the action generator in the original paper directly learns SIP to output the initiative-taking decision at the next turn.

Linear-chain CRF-based methods cannot be directly applied to SIP as they need to be given the unobservable utterance at the next turn. Based on the same BERT utterance encoder and prior-posterior inter-utterance encoders as in MuSIc, we implement the following:

- **VanillaCRF** only uses a unique transition matrix (see Equation 2.4).
- **VanillaCRF+features** feeding the three multi-turn features into the prior-posterior inter-utterance encoders by encoding the multi-turn features as one-hot vectors at each turn and concatenating the vectors with the BERT utterance representation.

[5] We use the code from the author: `https://github.com/zhenduow/conversationalQA`

[6] We implement the bi-encoders based on BERT, as MuSIc and most of the baselines use BERT.

[7] We use the code released by the author and adapt Co-Gen to SIP by making three changes: (i) we replace the GRU encoder with a BERT encoder like MuSIc has; (ii) Co-Gen requires a state vector (belief state and database records) that does not exist in CIS, so we replace the state vector with one-hot vectors encoding the current multi-turn features; and (iii) we remove reinforcement learning in Co-Gen as the rewards (task completion) do not exist in both CIS datasets.

- **DynamicCRF** uses adjacent input observations x_t, x_{t+1} to generate a dynamic transition matrix $G^{x_t,x_{t+1}}$ to model the dependency between the corresponding output decisions y_t, y_{t+1} [100, 113, 231, 235]. x_T is unseen so G^{x_{T-1},x_T} cannot be computed. Like the calculation of the prior/posterior emissions scores in MuSIc, we use the output of the prior inter-utterance encoder $\mathbf{h}_{pri}^{x_{T-1}}$ to generate a prior transition matrix $G^{x_{T-1}}$ for the output decisions y_{T-1}, y_T; $G^{x_{T-1}}$ approximates a posterior matrix G^{x_{T-1},x_T} generated by the output of the posterior encoder $\mathbf{h}_{pos}^{x_{T-1}}, \mathbf{h}_{pos}^{x_T}$ via an MSE loss.

2.5.6 Evaluation metrics

Because SIP is a binary classification problem, we use macro-averaged F1, precision, recall, and accuracy.

2.5.7 Implementation details

For all models except LLaMA, we use BERT encoders (BERT-base) on all datasets, set the hidden size to 768, batch whole conversations instead of individual turns, set the overall learning rate to 0.00002, use the Adam optimizer [124], and pick the best checkpoint in terms of F1 on the validation set.[8] For all CRF-based methods, our preliminary experiments showed that higher learning rates for transition matrices lead to better performance; we set the learning rate of transition matrices to 0.001. For LLaMA with all sizes, we randomly sample 2 complete conversations from the training set of WISE/MSDialog as demonstrations since other numbers lead to degraded performance. Note that all methods need to predict initiative-taking decisions for all system turns in all conversations in a dataset.

2.6 Results and Analysis

2.6.1 Performance comparison

To answer **RQ1.1**, we present the performance of MuSIc alongside all baseline methods on the WISE and MSDialog datasets in Tables 2.2 and 2.3, respectively. We have five observations.

First, LLaMA-7B/13B gets the worst result on WISE; on MSDialog, LLaMA-13B outperforms CtxPred (BERT), and is comparable to VanillaCRF and DynamicCRF, showing the effectiveness of LLMs. However, LLaMA with a larger parameter size even performs worse in most cases, e.g., 7B vs. 13B on WISE and 33B vs. 65B on MSDialog. This problem is also known as *inverse scaling* [168]. McKenzie et al. [168] identify four potential causes of it and highlight that there's still much to uncover in understanding it. Further investigation of this problem on SIP is left for future work.

Second, MuSIc and the linear-chain CRF-based methods outperform CtxPred (BERT). In terms of F1, VanillaCRF outperforms CtxPred (BERT) by 0.59% and

[8] We found that F1 can better show the ability of a model to deal with the class imbalance problem according to experimental results on the WISE and MSDialog validation sets.

Table 2.2: Performance comparison of SIP on the WISE dataset. Significant improvements over the best baseline results are marked with * (t-test, $p < 0.05$). The significance test is only performed on accuracy because it gives a score for each individual example, while other metrics evaluate the performance over all examples. At the time of writing, only LLaMA-7B and LLaMA-13B have Chinese versions available.

Methods	F1	Precision	Recall	Accuracy
LLaMA-7B	46.96	46.69	47.57	75.45
LLaMA-13B	26.91	55.01	54.28	26.96
CtxPred (BERT)	68.47	69.66	67.52	84.16
RCSQ	70.11	71.57	68.96	85.07
Co-Gen (action prediction)	67.65	69.89	66.14	84.40
Co-Gen (SIP)	69.47	71.37	68.09	85.01
VanillaCRF	69.06	71.38	67.46	85.04
DynamicCRF	69.21	71.25	67.75	84.97
VanillaCRF+features	69.32	71.85	67.61	85.25
MuSIc	**71.40**	**73.53**	**69.84**	**85.98***

2.14% on WISE and MSDialog, respectively. The gains indicate that it is beneficial for SIP to capture dependencies between adjacent initiative-taking decisions.

Third, both MuSIc and VanillaCRF+features outperform VanillaCRF and DynamicCRF, indicating that it is beneficial for SIP to take into account the impact of multi-turn features on an initiative-taking decision. Also, in terms of F1, MuSIc outperforms VanillaCRF+features by more than 3% on both datasets, underlining the importance of introducing such impact in the CRF layer.

Fourth, Co-Gen (action prediction) performs poorly, indicating that SIP cannot be effectively inferred from the predicted system actions. This could be due to the large action space, making the model prone to action prediction errors, which would propagate to SIP. It also implies the potential of SIP to reduce the decision space of action prediction, which we discuss in response to **RQ1.4**. Co-Gen (SIP) outperforms Co-Gen (action prediction), suggesting that sharing a common latent space between SIP and response generation is beneficial, however, MuSIc does not use that information.

Fifth, MuSIc outperforms RCSQ, which uses system initiative and non-initiative utterance candidates returned by retrieval models, whereas MuSIc does not have access to such information. MuSIc outperforms RCSQ in terms of F1 by 2.51% and 2.89% on WISE and MSDialog, respectively, confirming the effectiveness of MuSIc.

2.6.2 Visualisation of transition matrices

We show MuSIc's transition matrices $\mathbf{G}^{s2u}$, $\mathbf{G}^{u2s,n=0}$, $\mathbf{G}^{u2s,n>0,d=2}$ and $\mathbf{G}^{u2s,n>0,d>2}$ on WISE and MSDialog in Figure 2.6. We see different patterns in each transition matrix, indicating that different transition patterns are associated with different cases: (i) $\mathbf{G}^{u2s,n=0}$ shows that the user's initiative tends to transition to the system's initiative

Table 2.3: Performance comparison of SIP on the MSDialog dataset. Significant improvements over the best baseline results are marked with * (t-test, $p < 0.05$). The significance test is only performed on accuracy because it gives a score for each individual example, while other metrics evaluate the performance over all examples.

Methods	F1	Precision	Recall	Accuracy
LLaMA-7B	60.22	60.40	60.13	62.15
LLaMA-13B	62.54	62.73	63.21	62.99
LLaMA-33B	58.11	58.24	58.53	58.76
LLaMA-65B	55.30	62.33	60.44	55.93
CtxPred (BERT)	60.17	60.25	60.12	61.86
RCSQ	63.68	63.86	64.38	64.12
Co-Gen (action prediction)	53.76	55.23	54.35	58.47
Co-Gen (SIP)	63.13	63.62	62.97	65.25
VanillaCRF	62.31	63.24	62.17	64.97
DynamicCRF	62.01	61.95	62.20	62.99
VanillaCRF+features	63.29	64.19	63.10	65.82
MuSIc	**65.37**	**65.79**	**65.19**	**67.23***

when the system has not taken the initiative before. This corresponds to cases where the system tends to take the initiative for the first time to ask a clarifying question after the user has asked a question. (ii) $\mathbf{G}^{u2s,n>0,d=2}$ shows that the user's initiative tends to transition to the system's non-initiative if the system has taken the initiative at the last system turn. In other words, the system is less likely to take the initiative in two consecutive system turns if the user takes the initiative in the middle. (iii) According to $\mathbf{G}^{u2s,n>0,d>2}$, we see that compared to $\mathbf{G}^{u2s,n>0,d=2}$, if the system has not taken the initiative at the last system turn, the possibility of system initiative increases, especially when the user takes the initiative (on MSDialog). This corresponds to cases where the system takes the initiative once again to ask for feedback after answering a question from the user. The complexities of the patterns described above indicate that MuSIc effectively captures the impact of multi-turn features on an initiative-taking decision.

2.6.3 Effect of different multi-turn features

To answer **RQ1.2**, we evaluate MuSIc with multi-turn features on WISE and MSDialog. We consider four settings: (i) (r, n, d) is our final model considering all features (Equation 2.7); (ii) (r, n) does not consider *the distance to the last system's initiative turn* (Equation 2.6); (iii) (r) does not consider *the number of times the system has taken the initiative* (Equation 2.5); (iv) – does not consider any feature, degrading to VanillaCRF (Equation 2.4). See Table 2.4. All proposed multi-turn features contribute to the success of MuSIc as removing any of them decreases performance. On WISE, the MuSIc performance shows the biggest drop (0.92%) in terms of F1 score after removing role transition direction ((r) vs. –). On MSDialog, MuSIc's F1 score shows the biggest

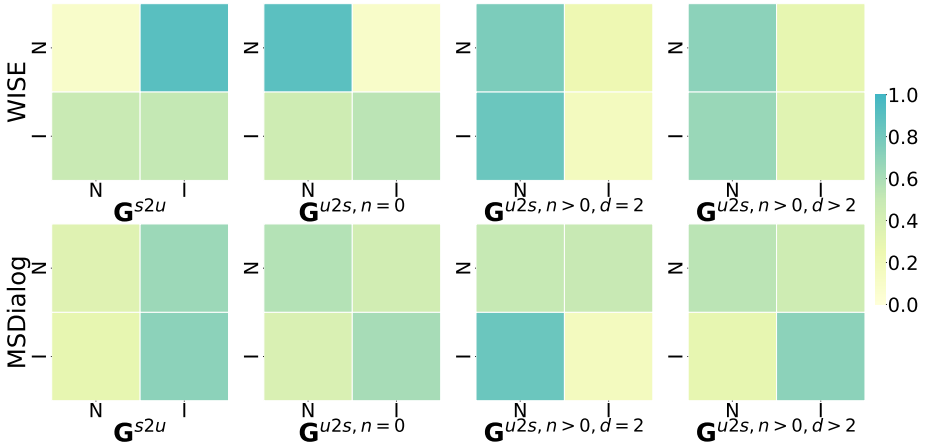


Figure 2.6: MuSIc's transition matrices learned on WISE and MSDialog. N and I denote *non-initiative* and *initiative*, respectively. $\mathbf{G}^{s2u}$ corresponds to "from system to user." $\mathbf{G}^{u2s,n=0}$, $\mathbf{G}^{u2s,n>0,d=2}$ and $\mathbf{G}^{u2s,n>0,d>2}$ correspond to "from user to system." $\mathbf{G}^{u2s,n=0}$ further corresponds to "the system has not taken the initiative before." $\mathbf{G}^{u2s,n>0,d=2}$ and $\mathbf{G}^{u2s,n>0,d>2}$ all correspond to "the system has taken the initiative once or more." The former further corresponds to "the distance to the last system initiative turn is 2 turns from the next system turn", while the latter corresponds to "the distance to the last system's initiative turn is more than 2 turns from the next system turn." See Section 2.4.5 for more information about each transition matrix. Transition scores are normalized across columns. Darker colors indicate higher scores.

drop (1.96%) after removing the number of times the system has taken the initiative ((r, n) vs. (r)).

2.6.4 Benefits of system initiative prediction on downstream tasks

We have demonstrated the effectiveness of MuSIc on SIP. Next, we illustrate two applications of SIP.

Improving clarification need prediction via transfer learning. To answer **RQ1.3**, we investigate the benefits of SIP to clarification need prediction (CNP) [9, 11, 16, 263, 264, 273]. Specifically, we explore how and to what extent knowledge shared among system-initiative actions learned through SIP on a dataset (MSDialog) can be reused to improve clarification need prediction on the single-turn ClariQ dataset [9, 11]. To achieve this, we propose a *SIP-to-clarification transfer learning* approach, which fine-tunes a model pre-trained on SIP on clarification need prediction. To evaluate the effectiveness of this strategy, we adopt MuSIc and the two strong clarification need prediction baselines CtxPred (BERT) [9, 11, 273] and RCSQ [263, 264] in two settings: (i) a clarification-only setting (CNP, ClariQ), where we only train models on the ClariQ training dataset, and (ii) a SIP-to-clarification transfer learning setting (SIP, MS. $\rightarrow$ CNP, ClariQ), where we first get the best checkpoints pre-trained on SIP on the MSDialog training set and then fine-tune them on the ClariQ training dataset. We also introduce

Table 2.4: Effect of multi-turn features in MuSIc. Notation for features explained in Section 2.6.3. * means (r, n, d) is significantly better than (–).

features	WISE (%)				MSDialog (%)			
	F1	Prec.	Recall	Acc.	F1	Prec.	Recall	Acc.
r, n, d	**71.40**	**73.53**	**69.84**	**85.98***	**65.37**	**65.79**	**65.19**	**67.23***
r, n	70.71	73.00	69.08	85.75	64.80	64.96	64.70	66.38
r	69.98	72.03	68.49	85.31	62.84	63.10	62.72	64.69
–	69.06	71.38	67.46	85.04	62.31	63.24	62.17	64.97

Table 2.5: Performance on clarification need prediction on ClariQ. (CNP, ClariQ) indicates models in the clarification-only setting, where we only train the models on the ClariQ training dataset; (SIP, MS. → CNP, ClariQ) indicates models in the SIP to-clarification transfer learning setting, where we further fine-tune the best checkpoints, pre-trained on SIP, on the ClariQ training dataset; MuSIc (CNP, MS. → CNP, ClariQ), pre-trained on the SIP examples only containing clarifying questions on the MSDialog training dataset. Significant improvements over the best baseline results are marked with * (t-test, $p < 0.05$).

Method	ClariQ (%)			
	F1	Prec.	Recall	Acc.
MiniLm-ANC	54.38	54.12	54.95	77.05
CtxPred (CNP, ClariQ)	50.59	50.66	50.59	78.69
RCSQ (CNP, ClariQ)	58.19	58.73	57.78	81.97
MuSIc (CNP, ClariQ)	61.26	64.64	59.67	85.25
CtxPred (SIP, MS. → CNP, ClariQ)	56.84	56.84	56.84	80.33
RCSQ (SIP, MS. → CNP, ClariQ)	61.26	64.64	59.67	85.25
MuSIc (CNP, MS. → CNP, ClariQ)	63.03	69.74	60.61	86.89
MuSIc (SIP, MS. → CNP, ClariQ)	**65.03**	**78.16**	**61.56**	**88.52***

MiniLm-ANC [16], an unsupervised learning method for clarification need prediction. We follow [16] to binarize the graded clarification need scores ranging from 1 (no need for clarification) to 4 (clarification is necessary) on ClariQ. Unlike [16], where scores are split in the middle, we only regard score 1 as not asking a clarifying question because the author of ClariQ states that clarification is still needed for scores 2 and 3 but not as much as score 4.[9] We present the results in Table 2.5.

MuSIc outperforms strong baselines on the single-turn ClariQ dataset in the supervised setting; it outperforms MiniLm-ANC and RCSQ (CNP, ClariQ) that use retrieved documents by 6.88% and 3.07% in terms of F1 score, respectively. Transferring knowledge from SIP to clarification need prediction (i.e., SIP-to-clarification transfer learning) benefits both MuSIc and the baseline models: performance increases with knowledge shared among system-initiative actions acquired from SIP. MuSIc (SIP, MS. → CNP,

[9] https://github.com/aliannejadi/ClariQ

Table 2.6: Performance on the downstream task. Methods used: mlc (multi-label classification), sg (sequence generation), and Co-Gen (state-of-the-art action prediction). + MuSIc: inject the initiative-taking decision predicted by MuSIc; + oracle: inject ground-truth initiative-taking decisions. Significant improvements over results of methods without using SIP results are marked with * (t-test, $p < 0.05$).

Method	WISE (%)				MSDialog (%)			
	F1	Prec.	Recall	Acc.	F1	Prec.	Recall	Acc.
mlc	21.59	25.80	20.24	48.78	18.23	20.41	18.06	48.83
+ MuSIc	**23.05**	**25.87**	**22.71**	**51.98***	**19.61**	**24.00**	**18.53**	**50.11***
+ oracle	24.78	27.53	24.82	54.69	21.77	29.08	19.84	56.51
sg	21.92	22.77	23.01	54.28	19.36	21.65	19.31	45.87
+ MuSIc	**23.28**	**25.92**	**24.07**	**56.68***	**21.12**	**22.94**	**21.00**	**49.87***
+ oracle	29.40	29.29	32.09	61.50	27.88	31.43	26.61	53.71
Co-Gen	24.17	24.14	25.77	55.02	21.34	22.98	20.95	48.94
+ MuSIc	**26.26**	**28.95**	**26.86**	**58.54***	**23.38**	**24.39**	**23.08**	**51.76***
+ oracle	30.49	31.49	32.23	62.32	28.37	29.14	28.27	57.47

ClariQ) shows an increase (3.77%) in terms of F1 compared to MuSIc (CNP, ClariQ), significantly exceeding all baselines in the transfer learning setting and achieving state-of-the-art performance on ClariQ.

Because the MSDialog training set contains system utterances of clarifying questions, pre-training on SIP on the MSDialog dataset already includes the pre-training of clarification need prediction. Is the improvement of transfer learning because the model learns knowledge shared among various system-initiative actions on the SIP task or because the model is just augmented with more training examples of clarification need prediction on MSDialog? In order to determine this, we introduce MuSIc (CNP, MS. $\rightarrow$ CNP, ClariQ), which is only pre-trained on clarification need prediction on the MSDialog training dataset, i.e., pre-trained on the partial SIP training examples containing clarifying questions. The performance of MuSIc (SIP, MS. $\rightarrow$ CNP, ClariQ) shows an increase (2%) in terms of F1 score compared to the performance of MuSIc (CNP, MS. $\rightarrow$ CNP, ClariQ), confirming that shared knowledge of various system-initiative actions learned through SIP benefits the model.

Improving downstream action prediction. To answer **RQ1.4**, we propose a SIP-aware action prediction framework where action prediction is fed with the initiative-taking decision predicted by MuSIc. In our scenario, the system can take multiple actions per turn. Multi-action system action prediction is typically modeled as multi-label classification [107, 138, 269] or sequence generation [117, 138, 225, 260]. We adopt two typical models for both types and a state-of-art system action prediction method, Co-Gen [280]: (i) following [107, 138, 269], we construct a multi-label classification model by using a BERT encoder to encode the context and feeding the [CLS] token to an MLP followed by sigmoid activation function to perform binary classification for each action; (ii) following [117, 138, 225, 260], we construct a sequence generation model

by using BERT to encode the context and feeding the [CLS] token to a GRU decoder to sequentially decode actions step by step; and (iii) Co-Gen is a sequence generation model, and we use Co-Gen (action prediction) (see Section 2.5.5) to generate actions. To inject initiative-taking decisions into these models, we first embed an initiative-taking decision (annotated during training and predicted by MuSIc during inference) to a 768-dimensional vector. For the models under (i) and (ii) we concatenate the vector with the [CLS] token and feed the concatenation to an MLP/GRU decoder. For Co-Gen, we concatenate the vector with the context representation (see [280]).

For evaluation, we adopt the same metrics as the previous sections except for accuracy. Here, accuracy is measured by the Hamming score (a.k.a. the intersection over the union) [98] that is widely used in multi-label classification evaluation [200]. Table 2.6 shows the results. The performance of three action prediction models fed with the initiative-taking decision predicted by MuSIc (+ MuSIc) is significantly improved compared to models without using SIP results. We think that this is because SIP, when effective, can reduce the action space of the downstream action prediction models. However, the downstream action prediction model cannot solve the SIP task (see Section 2.6.1). It shows that action prediction cannot replace SIP, reiterating the effectiveness of SIP in benefiting downstream tasks.

2.6.5 Error analysis

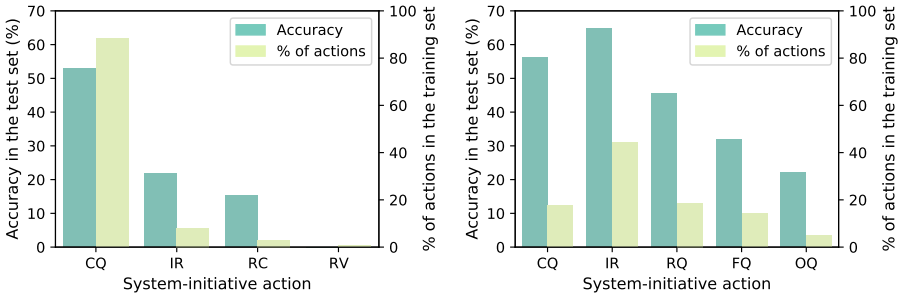
We conduct an error analysis of SIP. We group system initiative utterances in the test sets of WISE and MSDialog according to their annotated system-initiative actions; utterances in each group share the same system-initiative action. See Figure 2.7. MuSIc can still perform well on some system-initiative actions that only take up a limited proportion of the training sets. E.g., on MSDialog, the percentage of CQ is far less than the percentage of IR in the training set, but the performance of MuSIc is comparable in terms of CQ and IR in the test set. SIP enables knowledge sharing among various system-initiative actions, benefiting individual system-initiative actions. For revise (RV), there are only 4 and 3 system utterances of this type in the WISE training and test sets, respectively, numbers that are too small to properly evaluate the performance.

2.7 Conclusions and Future Work

In this chapter, we have introduced the task of *system initiative prediction* (SIP), which is to predict whether a CIS system should take the initiative at the next turn. This chapter examined the following thesis-level research question **RQ1**:

RQ1 How can we effectively model system initiative prediction (SIP), and how does this prediction benefit downstream tasks?

To answer this question, regarding the modeling of SIP, our empirical analysis found that it is natural to utilize probabilistic graphical models for SIP, but we faced two main challenges: solving the *input-incomplete* sequence labeling problem and explicitly modeling multi-turn features. To solve the challenges, we proposed MuSIc, which has (i) *prior-posterior inter-utterance encoders* to adapt CRFs to *input-incomplete* sequence



(a) WISE (b) MSDialog

Figure 2.7: SIP accuracy over utterance groups (utterances in one group share the same system-initiative action) in the test sets and percentages of system-initiative actions in the training sets. CQ: clarifying question (called *clarify* in WISE); IR: information request (called *request* in WISE); RV: revise; RC: recommendation (ask users if they would like something); OQ: original question; RQ: repeat question; and FQ: follow up question.

labeling by eliminating the need to be given the unobservable system utterance at the next turn, and (ii) a *multi-turn feature-aware CRF layer* to jointly consider *dependencies between adjacent user–system initiative-taking decisions* and *the impact of multi-turn features on an initiative-taking decision*. Next, we have explored how SIP can enhance two downstream tasks: clarification need prediction and action prediction. For the former, we have proposed a SIP-to-clarification transfer learning approach, which fine-tunes a model, pre-trained on SIP, on the clarification need prediction task. For the latter, we have introduced a SIP-aware action prediction framework, where downstream action prediction depends on SIP outcomes.

Experiments on two CIS datasets show that MuSIc outperforms various baselines including LLMs and achieves state-of-the-art performance on SIP. We get two additional insights: (i) LLMs do not show promising performance on SIP and just scaling up LLMs is not an effective way to solve SIP; and (ii) probabilistic graphical modeling is still competitive, effective and it should not be ignored in the era of LLMs. A visual analysis shows how the learned transition matrices exhibit MuSIc's interpretability and transparency. Regarding the applications of SIP to downstream tasks, SIP-to-clarification transfer learning significantly improves clarification need prediction performance; this indicates that the knowledge shared among various system-initiative actions learned through SIP can be used to improve the prediction of a specific system-initiative action. By applying this approach, MuSIc achieves state-of-the-art performance in clarification need prediction, setting a new benchmark on the ClariQ dataset. SIP-aware action prediction uses MuSIc-predicted SIP results, leading to significant performance gains in downstream action prediction.

Regarding limitations and future directions of this chapter, MuSIc does not utilize retrieved documents to improve SIP. Recent research into query performance prediction (QPP) on conversational search [174, 176] has shown that QPP can model retrieved documents and has the potential to help a CIS system take appropriate action at the next

turn [174, 176]. We plan to incorporate QPP-based features into our model. Clearly, splitting out SIP as a separate task adds complexity to CIS systems. Pre-training a model on SIP to learn knowledge shared among system-initiative actions and then fine-tuning the model on other tasks does not change the model architecture, but only increases training time without affecting inference time. Our proposed SIP-aware action prediction framework models SIP and action prediction as a two-stage process, which carries additional computational costs at inference time. We plan to improve the efficiency in the future, e.g., by modeling SIP and action prediction jointly in one stage.

In the next chapter, we will consider optimizing ranking strategy planning within the agentic workflow for information access. Specifically, we will explore dynamic per-query re-ranking depth prediction in the context of LLM.

Part II

Ranking Strategy Planning

3

Predicting Dynamic Re-Ranking Depths for LLM-based Re-Ranking

Research on ranking strategy planning has explored a key aspect: dynamic per-query re-ranking depth prediction [58, 135, 262, 285], which has been shown to improve both efficiency [58, 135, 262] and effectiveness [285] in non-large language model (LLM)-based re-ranking. Recently, LLM-based re-rankers [153, 195, 196, 198, 234, 293, 302] have achieved state-of-the-art re-ranking performance but at the cost of high query latency, limiting their practical use. While dynamic per-query re-ranking depth prediction can potentially help balance effectiveness and efficiency in LLM-based re-ranking, little research has explored this direction. Although using dynamic per-query re-ranking depths has been studied in non-LLM-based settings, its impact on efficiency and effectiveness in LLM-based re-ranking has not been systematically analyzed. Moreover, there is no established approach for modeling dynamic per-query re-ranking depth prediction in this context. Ranked list truncation (RLT) has been shown to be effective for dynamic per-query re-ranking depth prediction in non-LLM-based re-ranking [285], making RLT approaches a promising direction to explore in this new setting. This chapter targets the following thesis-level research question:

RQ2 In the context of LLM-based re-ranking, what are the potential benefits of using dynamic per-query re-ranking depths over fixed ones, and to what extent can RLT methods effectively predict dynamic re-ranking depths?

3.1 Introduction

Ranked list truncation (RLT), a.k.a. query cut-off prediction [51, 133], has been studied for over two decades [21, 163] and recently attracted lots of attention in the information retrieval (IR) community [30, 31, 139, 155, 259, 268]. The task of RLT is to determine how many items in a ranked list should be returned such that a user-defined metric is optimized [31]. The user-defined metric typically considers the balance between the utility of search results and the cost of processing search results [259]. RLT is crucial

This chapter was published as C. Meng, N. Arabzadeh, A. Askari, M. Aliannejadi, and M. de Rijke. Ranked list truncation for large language model-based re-ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 141–151, 2024.

Figure 3.1: A schematic diagram of RLT in the "*retrieve-then-re-rank*" setup.

in various IR applications where it is money- or time-consuming to review a returned item [268]. E.g., in patent search [150] or legal search [243, 259], providing a ranked list with an overwhelming number of items is too costly for patent experts or litigation support professionals [21].

A new angle. Existing studies mainly focus on RLT for single-stage retrieval, i.e., optimizing a user-defined metric (e.g., F1) of a retrieved list by truncating it at a certain position. In this chapter, we focus on RLT for *re-ranking*, i.e., RLT in a "*retrieve-then-re-rank*" setup, as shown in Figure 3.1. In this setup, we still truncate a given retrieved list but focus on enhancing trade-offs between effectiveness and efficiency in re-ranking; truncating the retrieved list directly translates to a reduction in re-ranking depth. RLT for re-ranking is important because: (i) RLT can improve re-ranking efficiency by sending variable-length lists of candidates to a re-ranker on a per-query basis; re-rankers are typically computationally expensive [139] and particularly, recently proposed LLM-based re-rankers [153, 195, 196, 198, 234, 293, 302] with billions of parameters lead to a substantial increase in computational overhead [304], making it hard to apply them in practice; applying a fixed re-ranking cut-off to all queries is a common practice in the literature; however, individual queries can be answered effectively by a shorter or a longer list of re-ranking candidates [36], so RLT can avoid unnecessary re-ranking costs by dynamically trimming the retrieved list; and (ii) RLT has the potential to improve re-ranking effectiveness; indeed, feeding a long retrieved list that includes many irrelevant items to a re-ranker instead can result in inferior re-ranking quality than a shorter retrieved list [285].

Despite its importance, limited research has explored the application of RLT methods in the "*retrieve-then-re-rank*" setup [58, 135, 262, 285]. E.g., Zamani et al. [285] only use one RLT method to truncate retrieved lists from BM25 to improve the performance of BERT-based re-ranking [185]. Put differently, there is a lack of systematic and comprehensive studies into the use of RLT methods that have originally been introduced to optimize retrieval, in the context of re-ranking, especially newly emerged LLMs-based re-ranking.

Research goal. In this chapter, we begin with a systematic analysis of the advantages of dynamic per-query re-ranking depths over fixed ones in LLM-based re-ranking, highlighting the importance of using dynamic per-query re-ranking depths in this new context. Next, we examine *to what extent established findings on RLT for retrieval are generalizable to the "retrieve-then-re-rank" setup*. Specifically, we study the following **findings** from the literature on RLT: (i) Supervised RLT methods generally perform better than their unsupervised counterparts (e.g., set a fixed cut-off for all queries) [30, 139, 259, 268]. (ii) Distribution-based supervised RLT methods (i.e., directly predict a distribution among all candidate cut-off points) perform better than their sequential

labeling-based counterpart (i.e., predict whether to truncate at each candidate point) [30, 259, 268]. (iii) Jointly learning RLT with other tasks (e.g., predicting the relevance of each item in the retrieved list) results in better RLT quality [259]. (iv) When truncating a retrieved list returned by a neural-based retriever, incorporating its embeddings improves RLT quality [155].

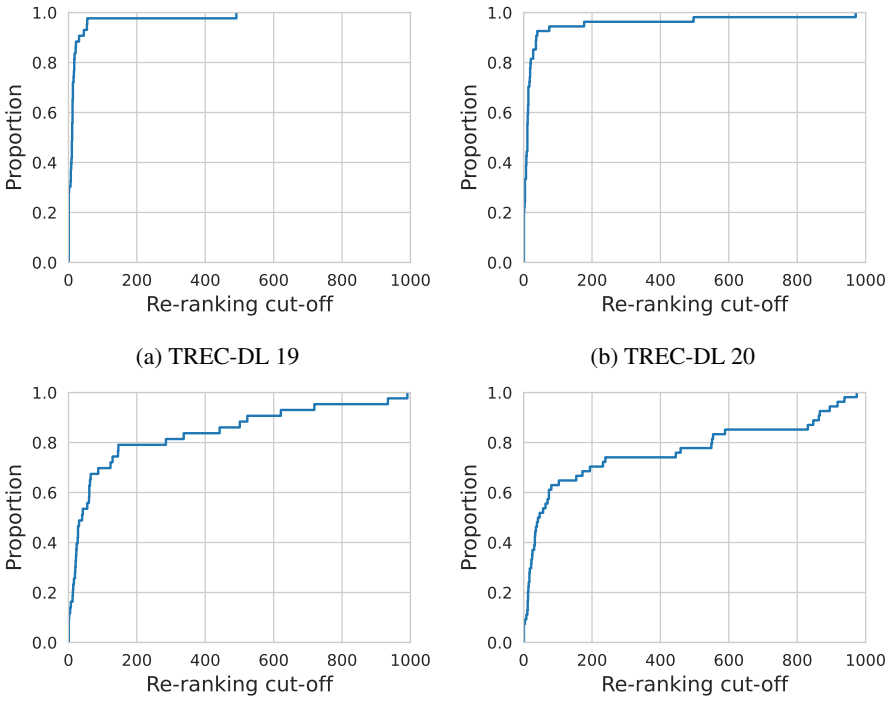
Reproducibility challenge. We highlight the main challenges of applying RLT methods from optimizing retrieval to optimizing re-ranking: (i) the new "retrieve-then-re-rank" setup leads to a new optimization goal for RLT methods, i.e., improving the trade-offs between effectiveness and efficiency in the re-ranking process; more importantly, a specific trade-off can be considered as the optimization goal to meet the requirements of a specific scenario, e.g., effectiveness is more important than efficiency in professional search than web-search; and (ii) the re-ranking setup introduces the *type* of re-ranker as a factor that influences RLT quality; also, it is important to investigate the impact of the interaction between retrievers and re-rankers on RLT; thus, it is important to explore RLT performance under different pipelines of widely-used retrievers, e.g., lexical, leaned sparse [87] and dense [153] retrievers, and different re-rankers, e.g., LLM-based [153] or pre-trained language model-based re-rankers [187].

Scope. We consider the challenges and examine each established finding from the literature on RLT in three settings: (i) we begin by checking if RLT methods optimizing for different trade-offs between effectiveness and efficiency of a state-of-the-art LLM-based re-ranker, RankLLaMA [153], with a lexical first-stage retriever; next, to study the impact of retriever types on RLT methods, we assess RLT methods for the LLM re-ranker with other types of retrievers, i.e., learned sparse (SPLADE++ [87]), and dense (RepLLaMA [153]) retrievers; and finally, to study the impact of the choice of re-rankers on RLT methods, we assess RLT methods for a widely-used pre-trained language model-based re-ranker, monoT5 [187]. We perform all experiments on the TREC 2019 and 2020 deep learning (TREC-DL) tracks [52, 53] and consider 8 RLT methods and pipelines involving 3 retrievers and 2 re-rankers, leading to various configurations.

Lessons. Our systematic analysis of dynamic per-query re-ranking depths in LLM-based re-ranking demonstrates that an effective per-query depth selection can significantly enhance both efficiency and effectiveness (see Section 3.2). Our experiments (see Section 3.5) reveal that findings on RLT do not generalize well to the "*retrieve-then-re-rank*" setup. E.g., we found supervised RLT methods do not show a clear advantage over using a fixed re-ranking depth; potential fixed re-ranking depths are able to closely approximate the effectiveness/efficiency trade-offs achieved by supervised RLT methods. Moreover, we found the choice of retriever has a substantial impact on RLT for re-ranking: with an effective retriever like SPLADE++ or RepLLaMA, a fixed re-ranking depth of 20 can already yield an excellent effectiveness/efficiency trade-off; increasing the fixed depth does not significantly improve effectiveness. An error analysis (see Section 3.5.4) reveals that supervised RLT methods tend to fail to predict when not to carry out re-ranking; moreover, they seem to suffer from a lack of training data.

Contributions. Our main contributions in this chapter are as follows:

- We conduct an empirical analysis to identify the limitations of fixed re-ranking depths and explore the potential advantages of using dynamic re-ranking depths on a



(a) TREC-DL 19

(b) TREC-DL 20

(c) TREC-DL 19

(d) TREC-DL 20

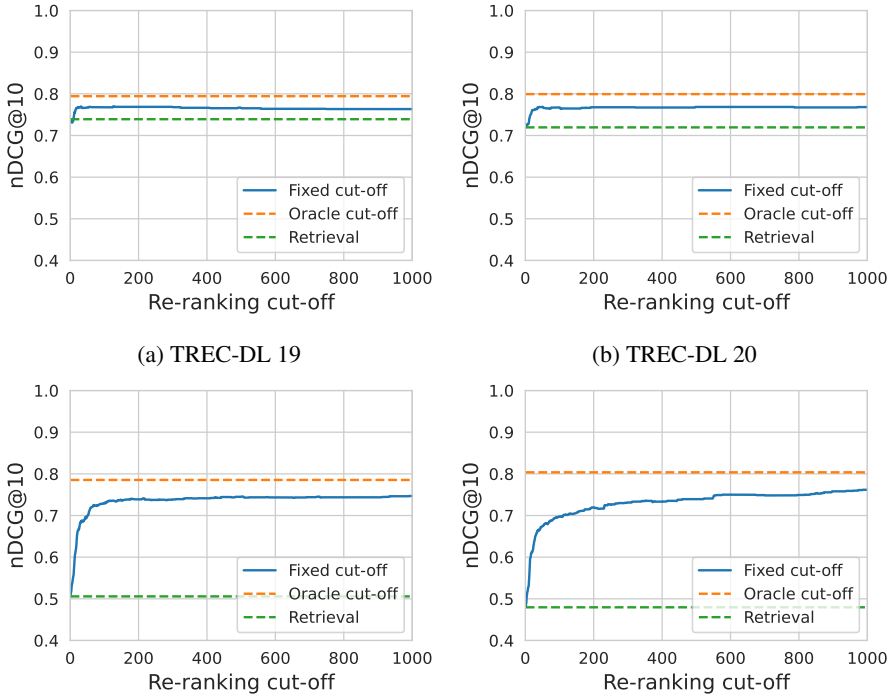
Figure 3.2: Cumulative distribution function of oracle cut-offs for RepLLaMA–RankLLaMA (a, b) and BM25–RankLLaMA (c, d) on TREC-DL 19 and 20. The oracle cut-offs are the minimum re-ranking cut-offs that yield the highest nDCG@10 values.

per-query basis in the context of LLM-based re-ranking.

- We reproduce a comprehensive set of RLT methods in a *"retrieve-then-re-rank"* perspective.
- We conduct an empirical analysis with a state-of-the-art LLM-based re-ranker, revealing that setting fixed re-ranking cut-offs results in unnecessary computational costs and diminishes re-ranking quality.
- We conduct extensive experiments on 2 datasets, 8 RLT methods, and pipelines involving 3 retrievers and 2 re-rankers, allowing a comprehensive understanding of how RLT methods generalize to the new perspective.

3.2 Motivation

In the literature, applying a fixed re-ranking cut-off to all queries is a common practice [152, 153, 195, 196, 198, 234, 293, 302]; however, individual queries may need a



(a) TREC-DL 19

(b) TREC-DL 20

(c) TREC-DL 19

(d) TREC-DL 20

Figure 3.3: nDCG@10 values for RepLLaMA–RankLLaMA (a, b) and BM25–RankLLaMA (c, d) w.r.t. re-ranking cut-offs on TREC-DL 19 and 20.

shorter or a longer list of re-ranking candidates [36]. We conduct an empirical analysis to demonstrate how RLT holds the potential to enhance both the effectiveness and efficiency in re-ranking compared to fixed cut-offs. To do so, we analyze two "retrieve-then-re-rank" pipelines on the TREC-DL 19 and 20 datasets. We use an LLM-based re-ranker (RankLLaMA [153]) in both pipelines, but for first-stage retrieval, we employ a lexical retriever (BM25) in one pipeline and an LLM-based dense retriever (RepLLaMA [153]) in the other.

3.2.1 Query-specific cut-offs improve efficiency

We study an *oracle* setup in which we define the oracle as the minimum re-ranking cut-offs yielding the highest nDCG@10 values. we find that individual queries have different oracle cut-offs with a wide range. Thus, a fixed cut-off either wastes computational resources or compromises re-ranking quality for queries that need a deeper cut-off. Figure 3.2 illustrates the cumulative distribution of oracle cut-offs for both pipelines on both datasets. Interestingly, about 30% of queries do not need re-ranking with RepLLaMA as the retriever, and approximately 5% with BM25; thus, calling expensive re-rankers can be omitted for these queries.

3.2.2 Query-specific cut-offs improve effectiveness

Figure 3.3 illustrates the comparison of re-ranking quality between using oracle and fixed cut-offs. We find that oracle cut-offs always perform *statistically significantly* (paired t-test, $p < 0.05$) better than all fixed cut-offs in terms of nDCG@10. Hence, a deeper re-ranking cut-off does not consistently result in improvement and can even be detrimental to re-ranking quality. Our finding is consistent with Zamani et al. [285]. While one might argue that the re-ranking results with a deeper cut-off might be underestimated because of the limited number of judged items within the top 10 ranks, i.e., judged@10 [196], we find that RankLLaMA's judged@10 values for using a fixed cut-off at 1,000 and oracle cut-offs are similar, e.g., 95.35% vs.96.05% and 97.41% vs. 97.41% when RepLLaMA as the retriever on TREC-DL 19 and 20, respectively.

RLT methods truncate the retrieved list (i.e., trim re-ranking candidates) on a per-query basis, suggesting that effective RLT has the potential to improve re-ranking efficiency and effectiveness.

3.3 Preliminaries and Task Definition

We first revisit the original task definition of RLT and then demonstrate how it extends to the re-ranking setting.

3.3.1 Original definition of ranked list truncation

Given a user query q, a collection C containing $|C|$ items, and a retrieved list $L = [d_1, d_2, \ldots, d_{|L|}]$ with $|L|$ ($|L| \ll |C|$) items induced by a first-stage retriever over C in response to q. An RLT approach f aims to predict a truncation point k that maximizes a target metric that is about the retrieved list L itself [30, 139, 155, 259, 268], formally:

$$k = f([x_1, x_2, \ldots, x_{|L|}]) \in \{1, 2, \ldots, |L|\}, \tag{3.1}$$

where $[x_1, x_2, \ldots, x_{|L|}]$ are item features corresponding one-to-one with the items in the retrieved list $L = [d_1, d_2, \ldots, d_{|L|}]$. Typically, x includes the retrieval score [30] and item statistics [139, 259, 268]. As for the target metric, $F1@k$ and $DCG@k$ have been widely used in prior studies [30, 31, 139, 155, 259, 268]. E.g., $F1@k$ is calculated as:

$$\begin{aligned} F1@k =& \frac{2 \cdot P@k \cdot R@k}{P@k + R@k}, \\ P@k =& \frac{1}{k} \sum_{i=1}^{k} \mathbb{I}(y_i = 1), R@K = \frac{1}{N_L} \sum_{i=1}^{k} \mathbb{I}(y_i = 1), \end{aligned} \tag{3.2}$$

where $y_i \in \{0, 1\}$ is the relevance label for item d_i in the truncated retrieved list, and N_L denotes the number of relevant items in the retrieved list L. Note that the original discounted cumulative gain (DCG) metric [114] is a monotonic metric since its value always increases with the value of k; it cannot evaluate RLT properly because the optimal solution would be to avoid truncation entirely [30]. Therefore, the DCG

metric employed in RLT penalizes non-relevant items, rendering it a non-monotonic metric [30, 31, 155, 259, 268]:

$$DCG@k = \sum_{i=1}^{k} \frac{y_i}{\log_2(i+1)} , \tag{3.3}$$

where $y_i \in \{1, -1\}$; $y_i = -1$ if item d_i is irrelevant to the query.

3.3.2 Ranked list truncation for re-ranking

In the "*retrieve-then-re-rank*" setup, we no longer focus on optimizing the retrieved list L, but we aim to test the capability of the RLT in optimizing the trade-offs between effectiveness and efficiency in re-ranking. As shown in Figure 3.1, the truncated retrieved list $L_{1:k} = [d_1, \ldots, d_k]$, serving as re-ranking candidates, is further forwarded to a re-ranker that returns a re-ranked list $\hat{L}_{1:k}$. We append the partial list $L_{k+1:|L|}$ from the retrieved list L that is not re-ranked to $\hat{L}_{1:k}$.

The target metric should evaluate the ranking quality of the re-ranked list $\hat{L}_{1:k}$ (with $L_{k+1:|L|}$) in terms of an IR evaluation metric (e.g., nDCG@10), and measure the computational cost of re-ranking.

3.4 Reproducibility Methodology

We state our research questions, the experiments designed to address them, and our experimental setup.

3.4.1 Research questions and experimental design

This chapter expands on the thesis-level research question **RQ2** by introducing the following chapter-specific research questions:

RQ2.1 Do RLT methods generalize to the context of LLM-based re-ranking with a *lexical first-stage retriever* when optimized for different effectiveness/efficiency trade-offs?

To address **RQ2.1**, we first quantify the trade-off between re-ranking effectiveness and efficiency, and then optimize RLT methods to model different trade-offs between effectiveness and efficiency, simulating different requirements and scenarios; then, we evaluate their predicted truncation positions in terms of effectiveness and efficiency in LLM-based re-ranking with a lexical retriever.

RQ2.2 Do RLT methods generalize to the context of LLM-based re-ranking with *learned sparse or dense first-stage* retrievers when optimized for the different trade-offs, and how does it compare to the case of a lexical retriever?

For answering **RQ2.2**, we still optimize RLT methods for different trade-offs of the LLM-based re-ranker used in **RQ2.1**, but study their performance given learned sparse or dense retrievers, and compare the results with those of using a lexical retriever.

RQ2.3 Do RLT methods generalize to the context of *pre-trained language model-based re-ranking*, and how does it compare to the case of an LLM-based re-ranker?

We address **RQ2.3** by evaluating the truncation points predicted by RLT methods w.r.t. effectiveness and efficiency in the context of a widely-used pre-trained language model-based re-ranker, and compare the results with those of the LLM-based re-ranker.

3.4.2 Experimental setup

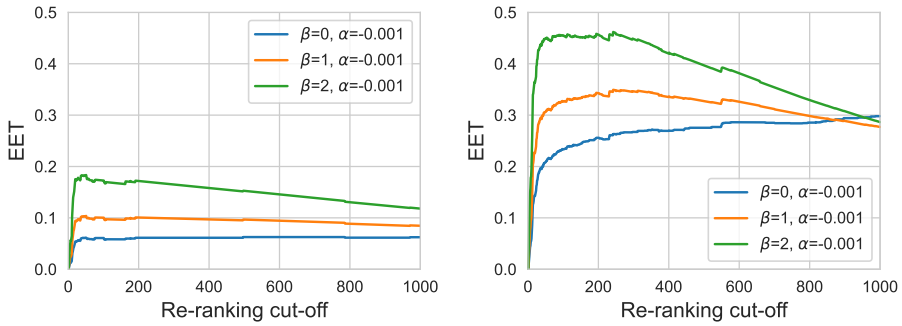
RLT approaches. We reproduce a variety of unsupervised and supervised RLT methods [30, 31, 139, 155, 259, 268].

We consider the following unsupervised RLT methods.

- **Fixed-k** [139] applies a fixed re-ranking cut-off to all queries; we follow common practice and consider cut-offs that are widely used in the literature about re-ranking, namely 10 [152], 20 [152, 195, 196], 100 [75, 153, 195, 196, 198, 234, 293, 302–304], 200 [153], 1,000 [211].
- **Greedy-k** [139] uses the fixed truncation position k that maximizes the target metric value on a training set.
- **Surprise** [31] first calibrates retrieval scores by using generalized Pareto distributions in extreme value theory [192], and truncates a ranked list using a score threshold.

We consider the following supervised RLT methods:

- **BiCut** [139] is a *sequential labeling-based* method; it uses a bidirectional long short-term memory (LSTM) to encode item features over a ranked list, and optimizes the LSTM make a binary prediction (continue or truncate) at each position in a ranked list.
- **Choppy** [30] is a *distribution-based* method, which directly predicts the distribution among all candidate cut-off points, using a transformer encoder [253] to encode item features over a ranked list and predicts the distribution.
- **AttnCut** [268] is also *distribution-based*, encoding item features using a bidirectional LSTM and a transformer encoder.
- **MtCut** [259] is also *distribution-based* and similar to AttnCut, but jointly trains the RLT task with two auxiliary tasks: predicting the relevance of each item in the ranked list and increasing the margin between relevant and irrelevant items. We use the MMoECut variant due to its superior performance.
- **LeCut** [155] is another *distribution-based* and similar to AttnCut, but can only work with a neural-based retriever and incorporates its query–item embeddings as one of the item features. Ma et al. [155] further optimize LeCut with an learning-to-rank (LtR) model jointly. We omit this phase for a fair comparison since other methods are trained without signals from an external LtR model.



(a) RepLLaMA–RankLLaMA (b) BM25–RankLLaMA

Figure 3.4: Average efficiency-effectiveness trade-off (EET) values across TREC-DL 20 queries w.r.t. re-ranking cut-offs. We use nDCG@10 in effectiveness σ. β values 0, 1 and 2 represent prioritizing effectiveness, balancing effectiveness and efficiency, and emphasizing efficiency, respectively.

We also include **Oracle**, which truncates the retrieved list at the earliest position maximizing re-ranking quality per query.

Optimizing effectiveness/efficiency trade-offs. The leading challenge of adapting RLT methods in the context of re-ranking is to optimize RLT methods with a specific trade-off between effectiveness and efficiency. To solve this challenge, we need to score each truncation point (i.e., re-ranking candidate cut-off) under different effectiveness/efficiency trade-offs. To do so, we quantify different trade-offs using the EET metric [261] and then compute EET scores at a specific trade-off for all re-ranking candidate cut-offs.

EET is defined for as the weighted harmonic mean of effectiveness σ and efficiency γ measures:

$$EET = \frac{(1+\beta^2)\cdot(\gamma\cdot\sigma)}{\beta^2\cdot\sigma+\gamma}, \tag{3.4}$$

where β is a hyperparameter to control the relative importance of effectiveness and efficiency, where $\beta = 0$ only considers effectiveness and as it increases more attention is paid to to efficiency. EET requires instantiation of σ and γ based on the specific use case [261]. We follow [261] to instantiate efficiency γ using "exponential decay":

$$\gamma = \exp(\alpha\cdot k), \tag{3.5}$$

where $k \in \{1, 2, \ldots, |L|\}$ is a truncation point (i.e., re-ranking candidate cut-off) in the given retrieved list L, and $\alpha < 0$ is a hyperparameter to control how rapidly the efficiency measure decreases; we set α to -0.001. We instantiate effectiveness σ as the re-ranking gain with a cut-off k, which is quantified by the difference of re-ranking performance with a cut-off k minus the performance without re-ranking; the performance is in terms of an IR evaluation metric (e.g., nDCG@10).

Therefore, we can adjust β in Equation 3.4 and α in Equation 3.5 in EET to quantify different effectiveness/efficiency trade-offs in re-ranking, so as to generate EET value

distributions across all cut-off candidates under different trade-offs. As illustrated in Figure 3.4, we consider three trade-offs between effectiveness and efficiency: β=0 (emphasizing effectiveness), 1 (weighting effectiveness and efficiency equally), and 2 (prioritizing efficiency). With the help of EET value distributions under the three trade-offs, we optimize all distribution-based RLT methods (Choppy [30], AttnCut [268], MtCut [259], LeCut [155]) and Greedy-k for the three trade-offs.

However, the sequential labeling-based RLT method BiCut [139] cannot optimize a EET value distribution. During training, BiCut optimizes the probability of "continue" and "truncation" at each position in a ranked list via the following loss:

$$\mathcal{L} = \sum_{i=1}^{|L|} (\eta \mathbb{I}(y_i = 0) \frac{p_i}{1-r} + (1-\eta) \frac{1-p_i}{r} \mathbb{I}(y_i = 1)), \tag{3.6}$$

where $y_i \in \{0, 1\}$ is the relevance label for an item at a position it, h p_i is the "continue" probability at a position i, and r is the proportion of relevant items in the ranked list; $\eta \in [0, 1]$ is a hyperparameter to control the balance between "continue" and "truncation". We optimize BiCut for different effectiveness/efficiency trade-offs by adjusting η values, e.g., BiCut trained with a high η value tends to truncate a retrieved list earlier, resulting in more efficiency. Specifically, we consider η=0.4 (emphasizing effectiveness), 0.5 (balancing effectiveness and efficiency), and 0.6 (prioritizing efficiency).[1]

Datasets. We experiment with 2 widely-used IR datasets, TREC 2019 and 2020 deep learning (TREC-DL) tracks [52, 53].[2] These datasets offer relevance judgments in multi-graded relevance scales per query. TREC-DL 19 and 20 are built upon MS MARCO V1 passage ranking collection encompassing 8.8 million passages.

Choice of retrievers. Regarding retrievers, we employ three distinct types: a lexical-based retriever BM25 [208], a learned sparse retriever SPLADE++ ("EnsembleDistil") [87] and an LLM-based dense retriever RepLLaMA (7B) [153]. To increase the comparability and reproducibility of this chapter, we obtain retrieval results of BM25 and SPLADE++ using the publicly available resource from Pyserini[3] and get retrieval results of RepLLaMA from Tevatron;[4] each retriever returns 1,000 items per query.

Choice of re-rankers. For **RQ2.1**, **RQ2.2**, we employ a state-of-the-art LLM-based point-wise reranker, RankLLaMA (7B) [153] and use the resource from Tevatron. For **RQ2.3**, we employ a widely-used pre-trained language model-based re-ranker, monoT5 ("monot5-base-msmarco") [187] and use the resource from PyGaggle.[5]

Evaluation metrics. We measure re-ranking effectiveness using nDCG@10, the official evaluation metric in TREC deep learning tracks [52, 53], and a widely employed metric in ranking literature [153, 187, 196]. We follow [304] to evaluate re-ranking efficiency by calculating the average re-ranking cut-off across all test set queries, i.e., the number of the average number of re-ranking inferences per query. This consideration is driven

[1] We also explore η values of 0.3 and 0.7. BiCut trained with the former tends not to truncate the retrieved list at all, while BiCut trained with the latter tends to truncate the entire retrieved list.

[2] We also conducted experiments on Robust04 and draw a similar conclusion as TREC-DL 19 and 20; due to space constraints, we show the result on Robust04 in our repository.

[3] https://github.com/castorini/pyserini

[4] https://github.com/texttron/tevatron

[5] https://github.com/castorini/pygaggle

by the fact that the re-rankers we employ in this chapter are point-wise, and the time spent in point-wise re-ranking is directly proportional to the length of a re-ranking cut-off (i.e., the length of a truncated retrieved list) [158]. We additionally gauge the efficiency by measuring per-query latency; all latency measurements exclude the time to load data and models. We do not consider the latency of first-stage retrieval. Note that RLT methods are lightweight with significantly fewer parameters compared to state-of-the-art re-rankers; the latency introduced by RLT methods can be neglected in the "retrieve-then-re-rank" setup.

Implementation details. We pass the top-1000 retrieved items to all RLT methods per query because 1,000 is typically the deepest re-ranking depth in the literature [153, 186, 211]. Note that Suprise [31] only depends on retrieval scores and uses a score threshold for truncation; the score threshold, set based on Cramer-von-Mises statistic testings [64], is not a tunable hyperparameter; thus, Suprise cannot be tuned for different effectiveness/efficiency trade-offs.

For all supervised RLT methods, we use identical item features to eliminate confounding factors from the input; each item is represented by its retrieval score, length, unique token count, and the cosine similarity between its tf-idf/doc2vec [132] vector and the vectors of its adjacent items. We follow [259, 268] to use gensim[6] for computing tf-idf and doc2vec vectors for each item; The dimension of the tf-idf vectors on the MS MARCO V1 collection is 846,221; we follow [259] to set the dimension of doc2vec vectors as 128. LeCut relies on query-item embeddings from a neural retriever as extra item features; thus, we provide LeCut with the concatenation of query and item embeddings from RepLLaMA. Note that we follow the original publications to set hyperparameters: for BiCut, we set # Bi-LSTM layers to 2 and the LSTM hidden size to 128; we train BiCut via Adam [124] with a learning rate of 1×10^{-4}; for Choppy, we set # transformer layers to 3, # transformer heads to 8, and transformer hidden size to 128; we train Choppy via Adam with a learning rate of 1×10^{-3}; for AttnCut, we set # Bi-LSTM layers to 2, the LSTM hidden size to 128, # transformer heads to 4, and the transformer hidden size to 128; MtCut and LeCut share almost the same hyperparameters with AttnCut; we train AttnCut, MtCut and LeCut via Adam with a learning rate of 3×10^{-5}. We train all supervised RLT methods for 100 epochs using a batch size of 64 on TREC-DL 19, then infer them on TREC-DL 20, and vice versa. All methods are trained/inferred on an NVIDIA A100 GPU (40GB).

3.5 Results and Discussions

3.5.1 Towards large language model-based re-ranking

To answer **RQ2.1**, we evaluate RLT methods (optimized for the three effectiveness/efficiency trade-offs) in the context of an LLM-based re-ranker (RankLLaMA [153]) with a lexical retriever (BM25). We report the results on TREC-DL 19 and 20 in Table 3.1; we also plot the result on TREC-DL 20 in Figure 3.5. We have three observations.

First, compared to unsupervised RLT methods, supervised RLT one only shows an advantage at achieving better re-ranking effectiveness at a less re-ranking cost in the

[6] `https://radimrehurek.com/gensim`

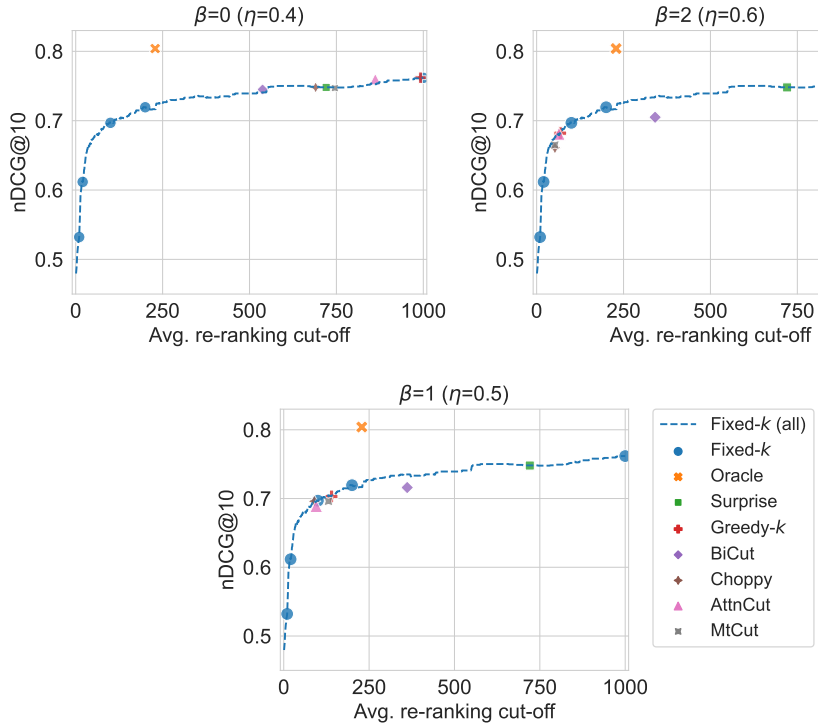


Figure 3.5: A comparison of RLT methods in predicting re-ranking cut-off points for BM25–RankLLaMA on TREC-DL 20. $\beta = 0$ ($\eta = 0.4$), $\beta = 1$ ($\eta = 0.5$), and $\beta = 2$ ($\eta = 0.6$) represent effectiveness emphasis, balance, and efficiency emphasis, respectively.

scenario emphasizing effectiveness; nevertheless, alternative fixed re-ranking depths can deliver results on par with those obtained through supervised methods. (i) In the scenario where effectiveness is prioritized ($\beta = 0$ and $\eta = 0.4$), supervised RLT methods achieve better re-ranking effectiveness while maintaining less re-ranking cost. For instance, Choppy ($\beta = 0$) and AttnCut ($\beta = 0$) show no significant difference from Fixed-k (1,000) in terms of nDCG@10 on TREC-DL 20, but with only 69% and 86% of the re-ranking cost of Fixed-k (1,000), respectively. (ii) In the other scenarios where efficiency received more attention ($\beta = 1/2$ and $\eta = 0.5/0.6$), supervised methods do not show an obvious advantage than unsupervised counterparts. E.g., while AttnCut ($\beta = 2$) and MtCut ($\beta = 2$) manage to achieve nDCG@10 values comparable to Fixed-k (100) using roughly half the re-ranking cost on TREC-DL 20, Greedy-k ($\beta = 2$) attains very similar results as AttnCut and MtCut. (iii) Moreover, as illustrated in Figure 3.5, other potential fixed ranking depths (excluding 10, 20, 100, 200 and 1,000) can yield results comparable to those of supervised methods across all scenarios.

Second, in scenarios balancing efficiency and effectiveness or prioritizing efficiency, distribution-based supervised RLT methods (Choppy, AttnCut, and MtCut) outperform the sequential labeling-based method (BiCut); however, in the scenario emphasizing

Table 3.1: A comparison of RLT methods in predicting re-ranking cut-off points for BM25–RankLLaMA on TREC-DL 19 and 20. Query latency is measured in seconds. The best nDCG@10 value in each column is marked in **bold**, and the second best is underlined. Significant differences with Fixed-k (100), Fixed-k (200) and Fixed-k (1,000) are marked with *, § and †, respectively (paired t-test, $p < 0.05$).

Method	TREC-DL 19			TREC-DL 20		
	Avg. k	nDCG@10	Lat.	Avg. k	nDCG@10	Lat.
w/o re-ranking	-	0.506*§†	-	-	0.480*§†	-
Fixed-k (10)	10	0.557*§†	0.30	10	0.532*§†	0.30
Fixed-k (20)	20	0.651*§†	0.60	20	0.612*§†	0.60
Fixed-k (100)	100	0.730	2.98	100	0.697§†	2.98
Fixed-k (200)	200	0.739	5.95	200	0.719*†	5.96
Fixed-k (1,000)	1,000	**0.747**	29.77	1,000	**0.762***§	29.78
Surprise	712	0.743	21.20	721	0.748*§†	21.46
Greedy-k ($\beta = 0$)	987	0.746	29.39	992	**0.762***§	29.54
BiCut ($\eta = 0.4$)	386	0.719	11.48	538	0.745†	16.01
Choppy ($\beta = 0$)	843	<u>0.744</u>	25.10	690	0.748*§	20.56
AttnCut ($\beta = 0$)	904	**0.747**	26.92	862	<u>0.760</u>*§	25.67
MtCut ($\beta = 0$)	844	0.741	25.12	745	0.747*§†	22.20
Greedy-k ($\beta = 1$)	242	0.737	7.21	140	0.703§†	4.17
BiCut ($\eta = 0.5$)	184	0.729†	5.48	362	0.716†	10.77
Choppy ($\beta = 1$)	141	0.733	4.19	90	0.696§†	2.67
AttnCut ($\beta = 1$)	211	0.720§†	6.29	95	0.689§†	2.83
MtCut ($\beta = 1$)	138	0.714§†	4.12	131	0.696†	3.90
Greedy-k ($\beta = 2$)	242	0.737	7.21	68	0.682§†	2.03
BiCut ($\eta = 0.6$)	131	0.693*§†	3.89	341	0.705†	10.15
Choppy ($\beta = 2$)	119	0.732	3.54	53	0.661*§†	1.57
AttnCut ($\beta = 2$)	64	0.692*§†	1.91	64	0.681§†	1.90
MtCut ($\beta = 2$)	62	0.687*†§	1.85	52	0.665§†	1.56
Oracle	157	0.785*§†	4.67	229	0.804*§†	6.80

effectiveness, BiCut shows a slight advantage. For instance, as shown in Figure 3.5, BiCut incurs lower costs to achieve nDCG@10 comparable to distribution-based methods when effectiveness is emphasized; however, in other scenarios ($\beta = 1/2$ and $\eta = 0.5/0.6$), the point denoting BiCut is below the dashed line denoting potential fixed re-ranking depths, while the points representing other supervised methods are on the line, indicating a worse effectiveness/efficiency balance achieved by BiCut.

Third, the supervised method (MtCut) learning RLT in a multi-task manner does not show a clear advantage over other supervised methods across all three trade-offs.

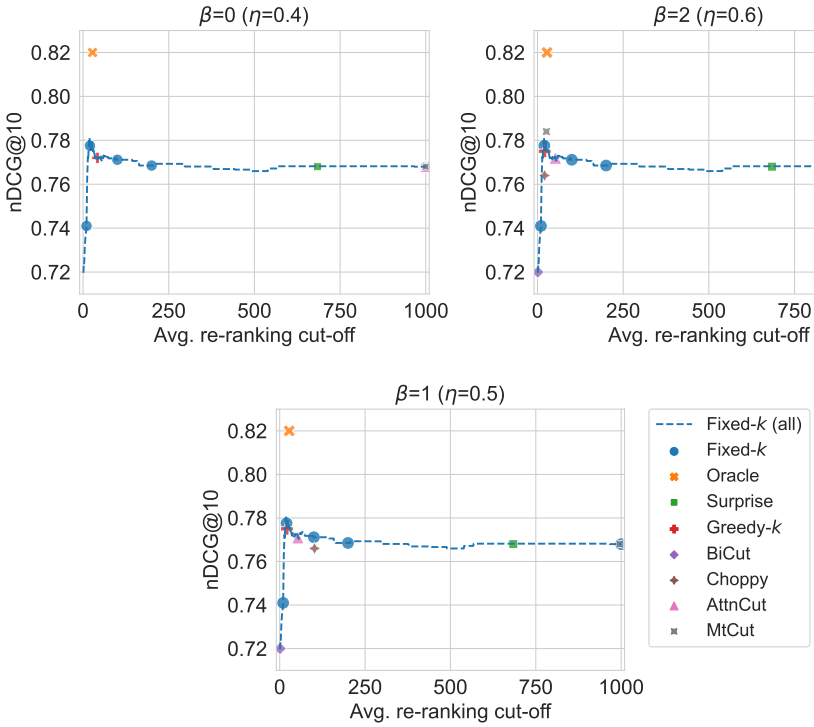


Figure 3.6: A comparison of RLT methods in predicting re-ranking cut-off points for Splade++–RankLLaMA on TREC-DL 20.

3.5.2 The impact of retriever types on ranked list truncation

To answer **RQ2.2**, we assess RLT methods (optimized for the three effectiveness/efficiency trade-offs) in the context of an LLM-based re-ranker (RankLLaMA [153]) with other novel retrievers; we explore learned sparse (SPLADE++ [87]) and dense (RepLLaMA [153]) retrievers. We present the results for SPLADE++–RankLLaMA and RepLLaMA–RankLLaMA on TREC-DL 19 and 20 in Tables 3.2 and 3.3, respectively. we plot both pipelines' results for TREC-DL 20 in Figures 3.6 and 3.7. Note that LeCut is only compatible with pipelines featuring a dense retriever. We have four observations.

First, different from the findings in Section 3.5.1, the unsupervised method Fixed-k (20) consistently achieves the best effectiveness/efficiency trade-off compared to supervised methods for both pipelines across all scenarios. Although some supervised methods achieve higher nDCG@10 values than Fixed-k (20), the nDCG@10 improvement is not statistically significant, e.g., for SPLADE++–RankLLaMA, MtCut ($\beta = 2$) outperform Fixed-k (20) by 0.006 in terms of nDCG@10 at a comparable cost on TREC-DL 20; however, this difference is too marginal. We believe Fixed-k (20) performs well due to the superior retrieval capabilities of SPLADE++ and RepLLaMA. Both tend to retrieve more relevant items at the top ranks, making a shallow fixed re-ranking cutoff both effective and efficient.

Table 3.2: Comparison of RLT methods in predicting re-ranking cut-off points for SPLADE++–RankLLaMA on TREC-DL 19 and 20. Query latency measured in seconds. The best nDCG@10 value in each column is marked in **bold**, and the second best is underlined. Significant differences with Fixed-k (20) are marked with * (paired t-test, $p < 0.05$).

Method	TREC-DL 19			TREC-DL 20		
	Avg. k	nDCG@10	Lat.	Avg. k	nDCG@10	Lat.
w/o re-ranking	-	0.731*	-	-	0.720*	-
Fixed-k (10)	10	0.740*	0.30	10	0.741*	0.30
Fixed-k (20)	20	0.773	0.60	20	0.778	0.60
Fixed-k (100)	100	0.769	2.98	100	0.771	2.98
Fixed-k (200)	200	0.769	5.95	200	0.769	5.96
Fixed-k (1,000)	1,000	0.768	29.77	1,000	0.768	29.79
Surprise	702	0.766	20.90	684	0.768	20.38
Greedy-k ($\beta = 0$)	65	0.770	1.94	42	0.772	1.25
BiCut ($\eta = 0.4$)	1,000	0.768	29.77	1,000	0.768	29.79
Choppy ($\beta = 0$)	904	0.768	26.92	1,000	0.768	29.79
AttnCut ($\beta = 0$)	999	0.768	29.74	999	0.768	29.76
MtCut ($\beta = 0$)	999	0.768	29.74	1,000	0.768	29.79
Greedy-k ($\beta = 1$)	65	0.770	1.94	21	0.775	0.63
BiCut ($\eta = 0.5$)	2	0.731*	0.06	1	0.720*	0.00
Choppy ($\beta = 1$)	68	0.771	2.03	102	0.766	3.03
AttnCut ($\beta = 1$)	46	0.771	1.38	53	0.771	1.58
MtCut ($\beta = 1$)	120	**0.775**	3.56	996	0.768	29.67
Greedy-k ($\beta = 2$)	18	0.769	0.54	21	0.775	0.63
BiCut ($\eta = 0.6$)	2	0.731*	0.06	1	0.720*	0.00
Choppy ($\beta = 2$)	1	0.731*	0.00	21	0.764	0.61
AttnCut ($\beta = 2$)	24	0.758*	0.70	52	0.772	1.55
MtCut ($\beta = 2$)	20	0.756*	0.59	26	**0.784**	0.77
Oracle	17	0.810*	0.52	28	0.820*	0.82

Second, similar to Section 3.5.1, distribution-based supervised methods typically offer a better effectiveness/efficiency trade-off in re-ranking than the sequential labeling-based method (BiCut). E.g., for SPLADE++–RankLLaMA, BiCut predicts depths that are either too shallow (1 or 2) or too deep (1,000); for RepLLaMA–RankLLaMA, certain distribution-based methods (e.g., Choppy) achieve similar nDCG@10 values to BiCut but at a reduced re-ranking cost.

Third, different from Section 3.5.1, the supervised method (MtCut), which learns RLT in a multi-task manner, exhibits a superior effectiveness/efficiency trade-off compared to other methods in a specific case; however, this advantage is not consistently observed. Specifically, as shown in Figure 3.6, in the scenario emphasizing efficiency, MtCut ($\beta = 2$) achieves the highest nDCG@10 value (except for Oracle)

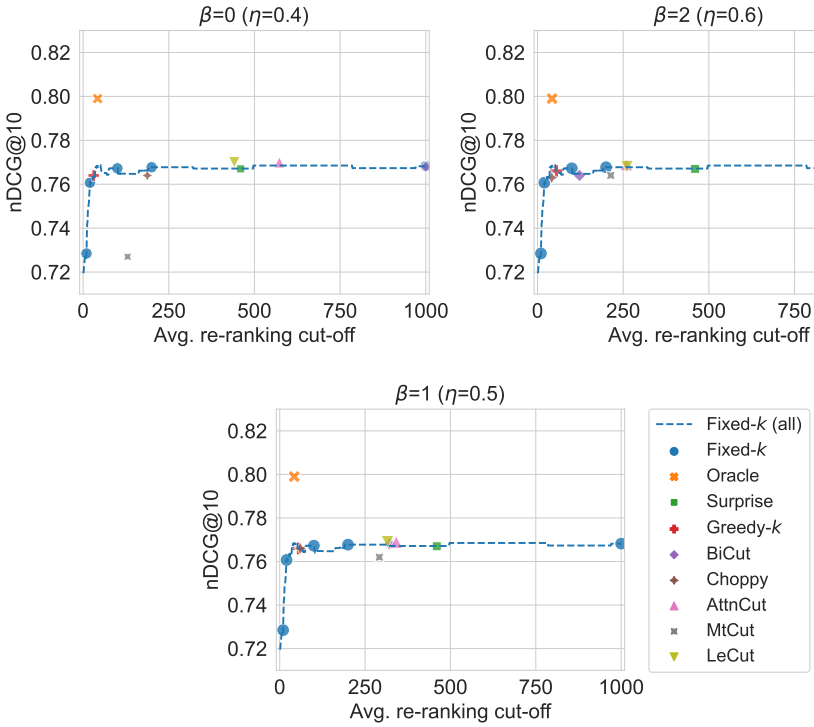


Figure 3.7: A comparison of RLT methods in predicting re-ranking cut-off points for RepLLaMA–RankLLaMA on TREC-DL 20.

for SPLADE++–RankLLaMA on TREC-DL 20 at a notably low cost. Nevertheless, as depicted in Figure 3.7, for RepLLaMA–RankLLaMA, the points representing MtCut consistently fall below the dashed line of Fixed-k, indicating a worse effectiveness/efficiency trade-off compared to other supervised methods.

Fourth, LeCut utilizes query-item embeddings from RepLLaMA to predict re-ranking cut-off points for RepLLaMA–RankLLaMA, leading to marginal improvements over other supervised methods in nDCG@10. These improvements are often too minimal to be significant; moreover, LeCut often attains these marginal improvements at the cost of efficiency. E.g., LeCut ($\beta = 0$) achieves the highest nDCG@10 value of 0.770 on TREC-DL 20, outperforming that of Choppy ($\beta = 0$) by 0.006, yet at more than double the cost of Choppy.

3.5.3 Towards pre-trained language model-based re-ranking

To answer **RQ2.3**, we evaluate RLT methods in the context of a pre-trained language model-based re-ranker (monoT5 [187]) with a lexical retriever (BM25). Note that using monoT5 [187] to re-rank the retrieved list returned by RepLLaMA [153] and Splade++ [87] yields worse results; hence we only consider the pipeline of BM25–monoT5. We report the raw result numbers on TREC-DL 19 and 20 in Table 3.4. We

Table 3.3: Comparison of RLT methods in predicting cut-off points for RepLLaMA–RankLLaMA on TREC-DL 19 and 20. Query latency measured in seconds. The best nDCG@10 value in each column is marked in **bold**, and the second best is underlined. Significant differences with Fixed-k (20) are marked with * (paired t-test, $p < 0.05$).

Method	TREC-DL 19			TREC-DL 20		
	Avg. k	nDCG@10	Lat.	Avg. k	nDCG@10	Lat.
w/o re-ranking	-	0.738*	-	-	0.720*	-
Fixed-k (10)	10	0.742*	0.30	10	0.729*	0.30
Fixed-k (20)	20	0.765	0.60	20	0.761	0.60
Fixed-k (100)	100	**0.769**	2.98	100	0.767	2.99
Fixed-k (200)	200	0.768	5.96	200	0.768	5.97
Fixed-k (1,000)	1,000	0.763	29.81	1,000	0.768	29.86
Surprise	458	0.765	13.66	460	0.767	13.73
Greedy-k ($\beta = 0$)	770	0.763	22.95	31	0.764	0.93
BiCut ($\eta = 0.4$)	1,000	0.763	29.81	1,000	0.768	29.86
Choppy ($\beta = 0$)	77	0.766	2.29	187	0.764	5.60
AttnCut ($\beta = 0$)	929	0.763	27.69	573	**0.770**	17.10
MtCut ($\beta = 0$)	510	0.758	15.19	130	0.727*	3.88
LeCut ($\beta = 0$)	418	0.766	12.45	441	**0.770**	13.17
Greedy-k ($\beta = 1$)	50	0.767	1.49	55	0.766	1.64
BiCut ($\eta = 0.5$)	167	0.766	4.97	323	0.768	9.63
Choppy ($\beta = 1$)	107	0.766	3.18	61	0.766	1.81
AttnCut ($\beta = 1$)	458	0.765	13.67	341	0.769	10.19
MtCut ($\beta = 1$)	583	0.761	17.37	292	0.762	8.71
LeCut ($\beta = 1$)	315	**0.769**	9.40	316	0.769	9.42
Greedy-k ($\beta = 2$)	50	0.767	1.49	55	0.766	1.64
BiCut ($\eta = 0.6$)	154	0.766	4.58	122	0.764	3.65
Choppy ($\beta = 2$)	72	0.766	2.15	41	0.763	1.24
AttnCut ($\beta = 2$)	210	0.767	6.26	259	0.769	7.74
MtCut ($\beta = 2$)	515	0.763	15.34	214	0.764	6.38
LeCut ($\beta = 2$)	214	**0.769**	6.39	261	0.768	7.80
Oracle	23	0.794*	0.70	42	0.799*	1.27

also plot the results on TREC-DL 20 in Figure 3.8.

Generally, the findings obtained with pre-trained language model-based and an LLM-based re-rankers (see Section 3.5.1) are similar. Specifically, (i) compared to unsupervised ones, supervised methods only show an advantage in the scenario emphasizing effectiveness, where supervised methods achieve better re-ranking quality at a lower cost; e.g., BiCut ($\eta = 0.4$) achieves an nDCG@10 value comparable to that of Fixed-k (1,000) while incurring only half the cost on TREC-DL 20; however, similar to Section 3.5.1, potential fixed re-ranking depths (excluding 10, 20, 100, 200, and 1,000) can still yield results that are very similar to those obtained with supervised methods

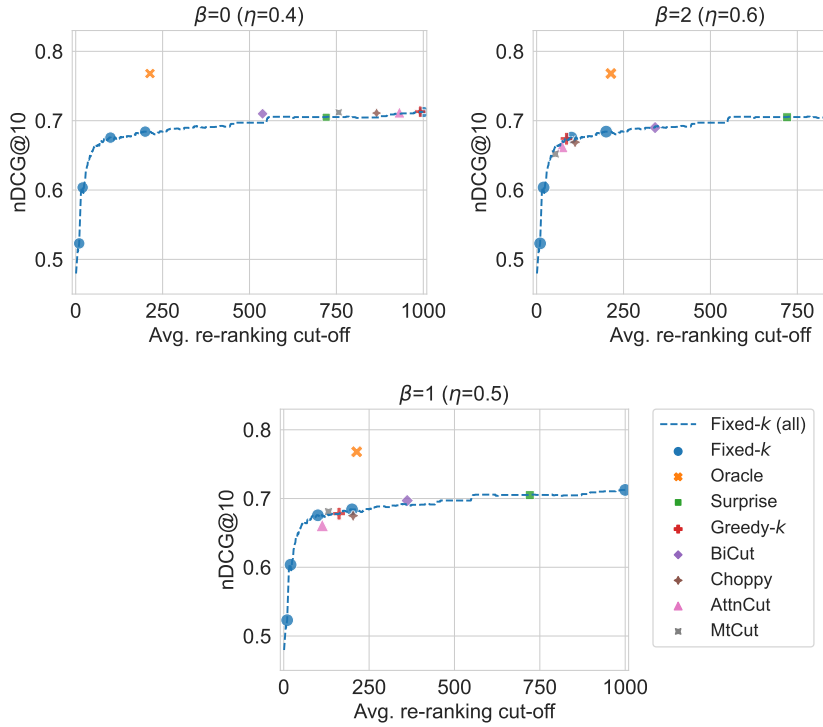


Figure 3.8: A comparison of RLT methods in predicting re-ranking cut-off points for BM25–monoT5 on TREC-DL 20.

(see Figure 3.8); (ii) distribution-based supervised methods perform better in scenarios balancing efficiency and effectiveness or prioritizing efficiency, while the sequential labeling-based method (BiCut) shows a slight advantage in the scenario emphasizing effectiveness; and (iii) the supervised method (MtCut), which learns RLT in a multi-task manner, does not consistently show a clear advantage over other supervised methods across all three trade-offs.

3.5.4 Error analysis

To understand the reasons behind the less-than-ideal performance of supervised RLT methods, we compare distributions of re-ranking cut-off points predicted by Oracle with those predicted by a supervised method. We consider two relatively effective methods,[7] MtCut ($\beta = 2$) and Choopy ($\beta = 2$), for pipelines featuring novel retrievers, SPLADE++ and RepLLaMA on TREC-DL 20; see Figure 3.9. First, both methods fail to predict a re-ranking cut-off of zero. For both pipelines, around 20% of queries do not need re-ranking. Thus, enhancing supervised RLT methods' ability to predict when re-ranking is unnecessary is crucial. Second, both methods perform worse when truncating RepLLaMA's retrieved lists (see Figures 3.9c and 3.9d) compared to

[7]Due to space limitations, we provide error analysis for all methods in our repository.

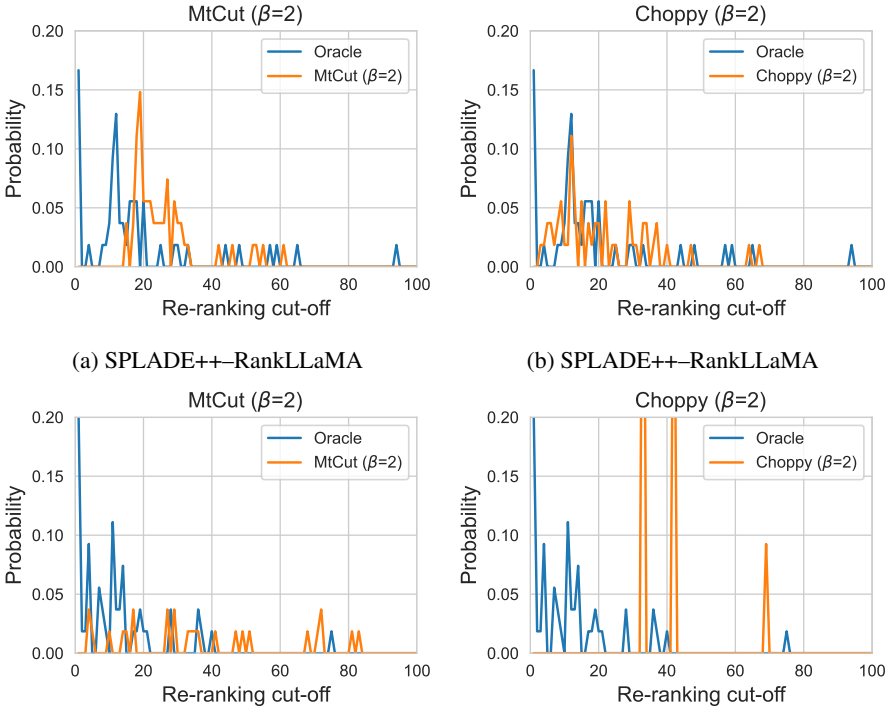
Table 3.4: A comparison of RLT methods in predicting re-ranking cut-off points for BM25–monoT5 on TREC-DL 19 and 20. Query latency measured in seconds. The best nDCG@10 value in each column is marked in **bold**, and the second best is underlined. Significant differences with Fixed-k (100), Fixed-k (200) and Fixed-k (1,000) are marked with *, § and †, respectively (paired t-test, $p < 0.05$).

Method	TREC-DL 19			TREC-DL 20		
	Avg. k	nDCG@10	Lat.	Avg. k	nDCG@10	Lat.
w/o re-ranking	-	0.506*§†	-	-	0.480*§†	-
Fixed-k (10)	10	0.553*§†	0.14	10	0.523*§†	0.14
Fixed-k (20)	20	0.634*§†	0.27	20	0.604*§†	0.27
Fixed-k (100)	100	0.706	1.37	100	0.676†	1.37
Fixed-k (200)	200	0.717	2.73	200	0.684†	2.73
Fixed-k (1,000)	1,000	**0.730**	13.66	1,000	**0.713***§	13.66
Surprise	712	0.721	9.73	721	0.705*§	9.84
Greedy-k ($\beta = 0$)	993	**0.730**	13.57	991	**0.713***§	13.54
BiCut ($\eta = 0.4$)	386	0.702	5.27	538	0.710*§	7.34
Choppy ($\beta = 0$)	784	0.727	10.70	866	0.711*§	11.82
AttnCut ($\beta = 0$)	888	0.729	12.13	931	0.712*§	12.72
MtCut ($\beta = 0$)	791	0.722	10.81	757	0.712*§	10.34
Greedy-k ($\beta = 1$)	112	0.709	1.53	162	0.678†	2.21
BiCut ($\eta = 0.5$)	184	0.711	2.51	362	0.697	4.94
Choppy ($\beta = 1$)	161	0.714	2.20	203	0.675†	2.78
AttnCut ($\beta = 1$)	198	0.701	2.70	113	0.661†	1.54
MtCut ($\beta = 1$)	145	0.690§†	1.97	131	0.681	1.79
Greedy-k ($\beta = 2$)	112	0.709	1.53	86	0.674†	1.17
BiCut ($\eta = 0.6$)	131	0.672*§†	1.79	341	0.690	4.66
Choppy ($\beta = 2$)	117	0.711	1.59	111	0.669†	1.51
AttnCut ($\beta = 2$)	68	0.677*§†	0.92	73	0.663†	1.00
MtCut ($\beta = 2$)	62	0.668*§†	0.85	54	0.652†	0.73
Oracle	181	0.774*§†	2.47	214	0.768*§†	2.92

SPLADE++ (see Figures 3.9a and 3.9b). Especially, Choppy ($\beta = 2$) seems to underfit for RepLLaMA (see 3.9d), suggesting a potential need for more training data.

3.6 Related Work

Our work relates to three strands of research: ranked list truncation (RLT), improving the efficiency of neural re-ranking, and the use of LLMs as re-rankers.



(a) SPLADE++–RankLLaMA

(b) SPLADE++–RankLLaMA

(c) RepLLaMA–RankLLaMA

(d) RepLLaMA–RankLLaMA

Figure 3.9: The distribution of re-ranking cut-off points on TREC-DL 20.

3.6.1 Ranked list truncation

RLT is also known as query cut-off prediction [51, 133]. For a query and a ranked list of documents, the RLT task is to predict the number of items in the ranked list that should be returned, to optimize a user-defined metric [31]. The task can potentially benefit IR applications where it is money- and time-consuming to review a returned item, e.g., in patent search [150] and legal search [243, 259]. Early work is mainly assumption-based (hence non-neural-based). This kind of research focuses on modeling score distributions by fitting prior distributions to them [21, 163], which helps identify the best cut-off. However, prior assumptions on score distributions do not always hold as retrieval settings change [139, 259]; hence we do not study this line of studies in this chapter. Assumption-free methods, on the other hand, learn to predict the truncation position during training and do not rely on a prior assumption. We have already introduced those methods (BiCut, Choppy, AttnCut, MtCut and LeCut) in Section 3.4.2.

Zamani et al. [285] apply the RLT method from [30] to truncate retrieval lists returned by BM25 for BERT-based re-ranking [185], finding that truncating the retrieval result list to avoid including a large number of non-relevant items in the lower ranks, achieves better re-ranking performance than using fixed cut-offs for all queries.

We differ from Zamani et al. [285] as we provide a systematic and comprehensive

study into the use of RLT methods in the context of re-ranking, especially newly emerged LLMs-based re-ranking.

3.6.2 Improving neural re-ranking efficiency

Improving neural re-ranking efficiency has been extensively studied. There are two ideas to improve the efficiency [96]: (i) speed up inference of a neural re-ranker, and (ii) reducing the number of inferences of a neural re-ranker. Approaches to (i) include a simpler re-ranker model [108], distilling knowledge in BERT [71] into a smaller re-ranker [93], pre-computing item representations at indexing time [157], and early-exiting [230, 271]. Early-exiting is to use only a partial model for "easy" item [36], which also has been studied in LtR [37, 38, 149]. Studies into (ii) are more related to this chapter. It includes *multi-stage re-ranking* [167, 186, 262, 294] and *candidate pruning* [58, 135, 186].

Multi-stage re-ranking first exploits faster and less effective re-rankers to discard likely non-relevant items and sends fewer candidate items to more expensive re-rankers in later stages. E.g., Zhang et al. [294] first use a feature-based LtR model to reorder the items returned by BM25 and then send the top-k (applied to all queries) items returned by the faster re-ranker to a BERT re-ranker.

Candidate pruning trims the candidate list in the first (or earlier) stage and then forwards the pruned ranked list to the next stage re-ranking. Wang et al. [262] propose a boosting algorithm for jointly learning pruning and ranker stages. Culpepper et al. [58] use a cascade of binary classifiers based on random forests; each classifier is used to predict whether to truncate the given ranked list at a specific cut-off value. Li et al. [135] propose a score-thresholding method, which makes sure the trimmed candidate list produces re-ranking outcomes that satisfy the user-specified error tolerance of an IR evaluation metric.

We also differ from Asadi and Lin [22] and Tonellotto et al. [244], who investigate improving the efficiency of candidate generation, i.e., first-stage retrieval. Specifically, Tonellotto et al. [244] predict the number of candidate items that should be retrieved by the candidate generation algorithm WAND [34] on a per-query basis. Our focus lies in improving re-ranking efficiency by truncating retrieved lists; in our setup, the retriever always returns a fixed number of items.

This chapter also differs from Ganguly and Yilmaz [90], who propose variable-depth pooling (VDP) to reduce relevance judgment costs in collection construction; VDP uses query performance prediction (QPP) [18–20] to predict variable cut-off depths for ranked lists when building a pool of items for a query. In contrast, we focus on using RLT methods in the re-ranking scenario.

3.6.3 Large language models as re-rankers

There are four paradigms of LLM-based re-ranking: *pointwise* [75, 153, 211, 302], *pairwise* [198], *listwise* [152, 195, 196, 234, 238, 293], and *setwise* [304]. 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. There are two popular ways of computing relevance scores, *special token-based* [33, 152, 153, 302] or

query likelihood-based [75, 211, 303]: *Special token-based* methods either use LLMs' output logits of special tokens [33, 302] to compute relevance scores, or compute them by projecting LLMs' representation of a special token [153]; *query likelihood-based* methods regard as a relevance score the likelihood of generating the user query given an item. Qin et al. [198] argue that outputting calibrated relevance scores for sorting is challenging for LLMs and requires accessing the generation API, and it is unnecessary for LLMs to compute relevance scores because re-ranking requires relative ordering.

The *pairwise* paradigm [198] 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 frame re-ranking as a pure generation task and directly output the reordered ranking list given a query and a ranked list return by first-stage retriever [152, 195, 196, 234, 238, 293]. Compared to pointwise and pairwise counterparts that sort items by multiple inference passes of LLMs, listwise re-rankers have the potential of achieving higher effectiveness by referring to multiple items simultaneously to determine their relative ordering [195, 293].

Given the low efficiency of pairwise (multiple inference passes) and listwise (multiple decoding steps) re-rankers, the *setwise* paradigm [304] 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 study provides an alternative perspective of enhancing effectiveness and efficiency in LLM-based re-ranking via RLT.

3.7 Conclusions and Future Work

In this chapter, we have conducted a systematic analysis of dynamic per-query re-ranking depths, and have reproduced numerous RLT methods in the context of LLM-based re-ranking. Our findings contribute to answering the following thesis-level research question:

RQ2 In the context of LLM-based re-ranking, what are the potential benefits of using dynamic per-query re-ranking depths over fixed ones, and to what extent can RLT methods effectively predict dynamic re-ranking depths?

For the first part of this question, our analysis demonstrates that effective per-query re-ranking depth selection can significantly improve both efficiency and effectiveness. From an efficiency perspective, we observe that when using a highly effective retriever, re-ranking does not improve ranking quality for about 30% of queries. In such cases, setting the re-ranking depth to zero allows the system to bypass costly LLM-based re-ranking, leading to substantial computational savings. From an effectiveness perspective, we found that increasing the re-ranking depth can sometimes degrade ranking quality. This occurs when the re-ranker mistakenly lifts irrelevant documents to higher positions in the ranked list. Our results suggest that dynamically adjusting the re-ranking depth to exclude "false positive" re-ranking candidate documents has the potential to enhance

overall retrieval effectiveness.

For the second part of this research question, our reproducibility study of RLT methods in LLM-based re-ranking has revealed that they do not demonstrate a clear advantage over using a fixed re-ranking depth. Also, we showed that findings on RLT do not generalize well to this new setup. We found that (i) supervised RLT methods do not demonstrate a clear advantage over their unsupervised counterparts; potential fixed re-ranking depths can closely approximate the effectiveness/efficiency trade-off achieved by supervised methods; (ii) distribution-based supervised methods achieve better effectiveness/efficiency trade-offs than their sequential labeling-based counterpart in most cases; the latter attains better re-ranking effectiveness at a lower cost for pipelines using BM25 retrieval; (iii) jointly learning RLT with other tasks [259] does not consistently yield a clear improvement; it only demonstrate a superior re-ranking effectiveness/efficiency trade-off for SPLADE++–RankLLaMA; and (iv) incorporating neural retriever embeddings [155] does not exhibit a clear advantage; it merely yields marginal improvements in re-ranking effectiveness for RepLLaMA–RankLLaMA.

We also learn valuable lessons from our experiments: (i) the type of retriever significantly affects RLT for re-ranking; with an effective retriever like SPLADE++ or RepLLaMA, a fixed re-ranking depth of 20 can already yield an excellent effectiveness/efficiency trade-off; increasing the fixed depth do not significantly improve effectiveness; using a fixed depth of 200 for retrieved lists returned by RepLLaMA, as done by Ma et al. [153], results in unnecessary computational costs; and (ii) the type of re-ranker (LLM or pre-trained LM-based) does not appear to influence the findings.

We identify future directions: (i) the gap between Oracle and RLT methods highlights the necessity of enhancing RLT methods for re-ranking; we plan to solve the potential data scarcity issue highlighted in Section 3.5.4, and explore the use of QPP methods for predicting query-specific re-ranking cut-offs [20]; (ii) we only consider point-wise re-rankers; we plan to explore RLT for pair-wise and list-wise LLM-based re-rankers [195, 196, 198, 293]; and (iii) we plan to explore RLT for re-ranking in conversational search (CS) [182].

In the next chapter, we shift our focus to the ranking result reflection component in agentic workflows for information access, where we explore QPP, a crucial ranking evaluation method in the newly emerging conversational search area.

Part III

Ranking Result Reflection

4

Query Performance Prediction in Conversational Search

Ranking result reflection is a key component of agentic workflows for information access. It assesses ranking quality and, if found inadequate, triggers refinement processes [227, 277]. Accurate ranking quality prediction is critical, as it directly influences decision-making in execution paths. Various approaches have been explored to automate ranking quality assessment. Among them, query performance prediction (QPP) has been a longstanding focus in the information retrieval (IR) community, where effective QPP methods have been used to adjust execution paths and improve ranking quality. Given its potential, this chapter and the next focus on QPP.

As information access systems increasingly operate in multi-turn conversational settings, ranking result reflection must also evolve. While QPP has been extensively studied in traditional ad-hoc search, limited research has studied it in the emerging area of conversational search. Unlike ad-hoc search, conversational search is characterized by context-dependent queries and a strong emphasis on top-ranked results, particularly for mobile interfaces with limited screen space. Given the differences, it is unclear how well existing QPP methods perform in this context. Also, how to adapt QPP to this context presents an open challenge. To address these gaps, this chapter investigates the following thesis-level question:

RQ3 How can QPP methods originally designed for ad-hoc search be effectively adapted to conversational search, and how well do QPP methods for ad-hoc search perform in conversational search?

4.1 Introduction

Query performance prediction (QPP) is an essential task in information retrieval (IR). It is about estimating the retrieval quality of a search system for a given query without relevance judgments [65, 67, 91, 105, 283, 300]. QPP has been long studied in the IR community [56]. Numerous benefits of QPP have been identified, including selecting

This chapter was published as C. Meng, N. Arabzadeh, M. Aliannejadi, and M. de Rijke. Query performance prediction: From ad-hoc to conversational search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2583–2593, 2023.

the most effective ranking algorithm for a query [105, 106, 283] based on the difficulty of the input query.

Recent years have seen remarkable advancements in conversational search (CS) [182], one of the long-standing goals of IR. CS is the task of retrieving relevant documents in response to each user query in a multi-turn conversation [61]. Significant progress has been made across multiple CS subtasks [286], including passage retrieval [61, 62, 115, 282], query rewriting [257, 281], mixed-initiative interactions [7, 284], response generation [171–173], and evaluation [77, 79].

Specifically, passage retrieval has been the main focus of TREC CAsT 2019–2022 [61–63, 115, 190], where modeling long conversational context for retrieval is shown to be challenging [8]. Moreover, research has shown that mixed-initiative interactions can lead to improved user and system performance [7, 305]. QPP holds significant potential to enhance CS in multiple ways. For instance, effective QPP can help a CS system take appropriate action at the next turn, e.g., take the initiative in asking a clarifying question or saying "I cannot answer your question" to the user, instead of giving a low-quality or risky answer when the estimated retrieval quality for the current user query is low [16, 210].

Despite its importance, little research has been done on QPP for CS [174]. We take the first steps in this direction by conducting a comprehensive reproducibility study, where we examine a variety of QPP methods that were originally designed for ad-hoc retrieval in the setting of CS. We aim to characterize the novel challenges of QPP for CS and highlight the unique characteristics of this field, while simultaneously assessing the effectiveness of existing QPP methods in a conversational setting.

Challenges. We highlight three main challenges of QPP applied to CS that distinguish it from the ad-hoc search setting:

(1) a user query in a conversation depends on the conversational context, i.e., it may contain omissions, coreferences, or ambiguities [282]. Many QPP methods rely heavily on the input query [15, 45, 65, 67, 102, 283], yet they are primarily designed for self-contained queries in ad-hoc search. Thus, they lack the necessary capabilities to interpret context-dependent conversational queries effectively.

(2) QPP for CS has to predict the performance of novel retrieval approaches, approaches that are specifically designed for CS; two main groups of CS methods have been proposed to solve the query understanding challenge in CS, i.e., query-rewriting-based retrieval [143, 169, 252, 257, 270, 281] and conversational dense retrieval methods [123, 142, 164, 164, 165, 201, 282]. However, it remains unclear how existing QPP methods can effectively predict retrieval performance for these two approaches.

(3) QPP for CS should focus on estimating the retrieval quality for the top-ranked results rather than for a full-ranked list because CS systems need to return brief responses to adapt to limited-bandwidth interfaces, such as a mobile screen [286].

Research goals. In this chapter, we first explore how to adapt existing QPP methods that heavily depend on input queries to address the challenge of context-dependent query understanding in conversational search. To bridge this gap, we propose using

self-contained query rewrites generated by off-the-shelf query rewriting methods as input to these QPP methods. We use this approach to predict the performance of both query-rewriting-based retrieval and conversational dense retrieval methods (see Sections 4.2.2 and 4.2.3). This approach ensures that these QPP methods continue to process self-contained queries, for which these QPP methods were originally designed in ad-hoc search. Building on this adaptation, we then conduct a systematic reproducibility study to examine whether established findings on QPP for ad-hoc search still hold in conversational search.

Specifically, we study the following findings from the literature on QPP for ad-hoc search: (i) supervised QPP methods outperform unsupervised QPP methods [15, 45, 65, 67, 102, 283]; (ii) list-wise supervised QPP methods outperform their point-wise counterparts [45, 67]; and (iii) retrieval score-based unsupervised QPP methods perform poorly in estimating the retrieval quality of neural-based retrievers [66, 102]. By examining each of these QPP-for-ad-hoc-search findings listed above in the setting of CS, we aim to characterize the problem of QPP applied to CS, with novel findings and directions for future research as additional outcomes.

Experiments. We conduct experiments on three widely-used CS datasets: CAsT-19 [62, 115], CAsT-20 [61], and OR-QuAC [201]. Our results demonstrate that feeding machine-generated query rewrites into QPP methods is an effective strategy for adapting them to predict the performance of both query-rewriting-based retrieval methods (see Section 4.4.1) and conversational dense retrieval methods (see Section 4.4.2).

Furthermore, predicting an IR evaluation metric with a shallow cut-off proves more challenging than predicting one with a deep cut-off (see Section 4.4.3).

Regarding our reproducibility study findings in the CS setting, we observe that: (i) supervised QPP methods distinctly outperform unsupervised counterparts only when a large amount of training data is available; unsupervised QPP methods show strong performance in a few-shot setting and when predicting the retrieval quality for deeper ranked lists; (ii) point-wise supervised QPP methods outperform their list-wise counterparts in most cases; however, list-wise QPP methods show a slight advantage in a few-shot setting and when predicting the retrieval quality for deeper ranked lists; and (iii) retrieval score-based unsupervised QPP methods show high effectiveness in estimating the retrieval quality of a conversational dense retrieval method, ConvDR, either for top ranks or deeper ranked lists.

Contributions. Our main contributions in this chapter are as follows:

- We propose an approach to adapt existing QPP methods for conversational search by using machine-generated self-contained query rewrites.

- We conduct a systematic reproducibility study to assess the effectiveness of existing QPP methods in conversational search.

- Our extensive experiments demonstrate that using query rewrites as input to QPP methods is an effective adaptation strategy. Additionally, we uncover important findings and insights into QPP for CS.

4.2 Preliminaries and Task Definition

4.2.1 From ad-hoc search to conversational search

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

$$\phi = f(q, D^k_{q;M}, D) \in \mathbb{R}\,, \tag{4.1}$$

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

Next, we define the task of QPP for CS. The CS task is to find relevant items for each query in a multi-turn conversation $Q = \{q_t\}^n_{t=1}$ [61], where n is the number of turns in a conversation. Unlike traditional ad-hoc search, the query q_t at turn t may contain omissions, coreferences, or ambiguities, making it hard for ad-hoc search methods to capture the underlying information need of the query q_t [282].

Two main groups of CS methods have been proposed to solve the query understanding challenge in CS, i.e., query rewriting-based retrieval [143, 169, 252, 257, 281] and conversational dense retrieval methods [142, 164, 282].

4.2.2 Towards query rewriting-based retrieval methods

In this section, we describe our approach to modeling QPP for predicting the performance of query rewriting-based retrieval methods. These methods first rewrite the query q_t into a self-contained query q'_t with the conversational history $Q_{1:t-1} = q_1, q_2, \ldots, q_{t-1}$, and then reuse ad-hoc search methods using the rewritten query q'_t as input. When estimating the retrieval quality of this group of CS methods, we define QPP for CS as:

$$\phi_t = f(q'_t, D^k_{q'_t;M}, D) \in \mathbb{R}\,, \tag{4.2}$$

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

4.2.3 Towards conversational dense retrieval methods

In this section, we present our approach to modeling QPP for predicting the performance of conversational dense retrieval methods. These methods train a query encoder to encode the current query q_t and the conversation history $Q_{1:t-1}$ into a contextualized query embedding that is used to represent the information need of the current query in a latent space [164, 282]. However, existing QPP methods do not have such a special module to understand the noisy raw utterances $Q_{1:t}$; directly feeding the raw utterances $Q_{1:t}$ into QPP methods may confuse them. Thus, when estimating the retrieval quality

of a conversational dense retrieval method, we still feed a query rewrite q'_t instead of the raw utterances $Q_{1:t}$ into QPP methods, formally:

$$\phi_t = f(q'_t, D^k_{Q_{1:t};M}, D) \in \mathbb{R}\,, \tag{4.3}$$

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

4.3 Reproducibility Methodology

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

4.3.1 Research questions

This chapter refines the thesis research question **RQ3** into the following research questions at the chapter level.

RQ3.1 Does the performance of QPP methods for ad-hoc search generalize to CS when estimating the retrieval quality of different query rewriting-based retrieval methods?

RQ3.2 Does the performance of QPP methods for ad-hoc search generalize to CS when estimating the retrieval quality of a conversational dense retrieval method? Is the QPP effectiveness influenced by the choice of query rewrites?

RQ3.3 What is the performance difference between QPP methods when predicting the retrieval quality for top-ranked items vs. for longer-ranked lists?

4.3.2 Experimental design

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

E1 To address **RQ3.1**, we estimate the retrieval quality of BM25 with three query rewriting methods, namely, T5, QuReTeC, and perfect rewriting (human-rewritten) [61, 62, 115]. Note that QPP methods and BM25 always share the same query rewrites.

E2 To address **RQ3.2**, we study the performance of QPP methods for a conversational dense retrieval method, ConvDR [282], on all three datasets. As ConvDR directly models the raw conversation context, no query rewriting step is required. However, no existing QPP methods can model raw conversations. Hence, we study the effect of feeding different query rewrites into QPP methods when predicting the performance of ConvDR.

Table 4.1: Actual retrieval quality of the CS methods used in this chapter in terms of nDCG@3.

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

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

4.3.3 Experimental setup

QPP methods. We analyze a variety of unsupervised/supervised QPP methods. For unsupervised ones, we consider clarity-based and score-based methods because they have been widely used in the literature. We consider more score-based ones since they have shown great effectiveness [39]. We consider one clarity-based method:

- Clarity [56] quantifies the degree of ambiguity of a query w.r.t. a collection of documents. Specifically, it measures the KL divergence between a relevance model [131] induced from top-ranked documents and a language model induced from the collection:

$$Clarity(q, D^k_{q;M}, D) = \sum_{w \in V} P(w|D^k_{q;M}) \log \frac{P(w|D^k_{q;M})}{P(w|D)}, \tag{4.4}$$

  where w and V denote a term and the entire vocabulary of the collection, respectively. The conjecture is that the larger the KL divergence is, the better the retrieval quality is.

We consider five score-based QPP methods:

- Weighted information gain (WIG) [300] measures the divergence of retrieval scores of top-ranked documents from those of the entire corpus: the higher the divergence is, the better the retrieval quality is [224, 239, 283]. WIG is formulated as:

$$WIG(q, D^k_{q;M}, D) = \frac{1}{k} \sum_{d \in D^k_{q;M}} \frac{1}{\sqrt{|q|}} (Score(q; d) - Score(q; D)), \tag{4.5}$$

  where $Score(q; d)$ and $Score(q; D)$ are the retrieval scores of document d and the entire collection D, respectively; $|q|$ is q's length.

- Normalized query commitment (NQC) [224] measures the standard deviation of retrieval scores of top-ranked documents; the standard deviation is normalized by the

retrieval score of the entire collection D. The higher the standard deviation is, the better the retrieval quality is assumed to be. NQC is modeled as:

$$NQC(q, D^k_{q;M}, D) = \frac{1}{Score(q;D)} \sqrt{\frac{1}{k} \sum_{d \in D^k_{q;M}} (Score(q;d) - \mu)^2}, \quad (4.6)$$

where μ is the mean retrieval score of the top-ranked documents.

- σ_{max} [191] is based on the standard deviation of retrieval scores of ranked documents but finds the most suitable ranked list size k for each query. The intuition is that most of the retrieved documents in a ranked list obtain a low retrieval score; considering such non-relevant documents would hurt QPP effectiveness. σ_{max} computes the standard deviation at each point in the ranked list and selects the maximum standard deviation so as to reduce the impact of the documents with a low retrieval score.
- n($\sigma_{x\%}$) [60], similar to σ_{max}, also uses a dynamic number of documents to calculate the standard deviation for each query, but only considers the documents whose retrieval scores are at least $x\%$ of the top retrieval score. The calculated standard deviation is normalized by query length.
- Score magnitude and variance (SMV) [239] argues that WIG and NQC mainly consider the magnitude and the variance of retrieval scores, respectively. SMV takes both aspects into consideration:

$$SMV(q, D^k_{q;M}, D) = \frac{\frac{1}{k} \sum_{d \in D^k_{q;M}} (Score(q;d) |\ln \frac{Score(q;d)}{\mu}|)}{Score(q;D)}, \quad (4.7)$$

where $Score(q;d)$ denotes score magnitude while $|\ln \frac{Score(q;d)}{\mu}|)$ represents score variance.

Recent studies show that BERT-based supervised QPP methods [15, 45, 67, 102] outperform other neural-based supervised QPP methods, such as NeuralQPP [283] and Deep-QPP [65]. Thus, we consider three competitive BERT-based supervised QPP methods:

- NQA-QPP [102] is the first supervised QPP method based on BERT. It feeds the standard deviation of retrieval scores, BERT representations for the given query and query-document pairs into a feed-forward neural network for estimating the retrieval quality.
- BERT-QPP [15] feeds the given query and the top-ranked document into BERT, followed by a linear layer for estimating the retrieval quality. We use the cross-encoder version of BERT-QPP as it outperforms the bi-encoder version.
- qppBERT-PL [67] is a listwise-document method. It splits the top-ranked documents into chunks and then uses BERT to encode all query-document pairs in each chunk; a sequence of query-document BERT representations in a chunk is fed into an LSTM and linear layers to predict the number of relevant documents in the chunk. A weighted average of the number of relevant documents across all chunks is calculated as the retrieval quality.

Table 4.2: Data statistics of CAsT-19, CAsT-20 and OR-QuAC.

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

We do not include BERT-groupwise-QPP [45]. It is another list-wise supervised QPP method, which uses cross-query information but it cannot be directly applied in a CS setting, as it would access the future next turn query q_{t+1} when estimating the difficulty of the current query q_t during inference, which is unrealistic in CS.

Query rewriting methods. We adopt the following query rewriting techniques/data in the passage retrieval and QPP process: (i) T5 rewriter[1] is fine-tuned on CANARD [76] query rewriting dataset; (ii) QuReTeC [257] is a BERT-based term expansion query rewriting method. We use the checkpoint released by the author;[2] and (iii) Human is the human-generated oracle query rewriting model obtained from the ground-truth data annotations.

CS methods to be evaluated for retrieval quality. We estimate the retrieval quality of two groups of CS methods: query rewriting-based retrieval and conversational dense retrieval methods. For the former, we consider: (i) T5+BM25 rewrites queries using the T5 rewriter and ranks documents using BM25[3]; (ii) QuReTeC+BM25 [257] performs query resolution using QuReTeC, followed by BM25 retrieval; and (iii) Human+BM25 uses the ground-truth query rewrites to rank documents using BM25. For the latter, we consider ConvDR [282] and use the code released by the author.[4] All CS methods return the top-1000 documents per query.

Datasets. We consider three CS datasets: (i) CAsT-19 [62, 115] is constructed manually to mimic a realistic conversation on a specific topic; in this dataset, a later query turn often depends on its previous queries; (ii) CAsT-20 [61] is more realistic and complex because the information needs of queries are derived from commercial search logs and queries can refer to previous system responses; and (iii) OR-QuAC [201] is a large-scale synthetic CS dataset built on a conversational QA dataset, QuAC [48]; there is usually only one annotated relevant item for each query in this dataset. All three datasets provide self-contained queries rewritten by humans for all raw queries. Table 4.2 lists details of the datasets.

Evaluation. A common method for evaluating QPP performance is to assess the correlation between the actual and predicted performance of a query set. We use Pearson's ρ linear coefficient and Kendall's τ ranking correlation for QPP evaluation,

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

because typically both are the most commonly used correlation metrics [91, 283]. We report the correlation based on the major metrics adopted by TREC CAsT [61, 62, 115], namely, nDCG@3 for high ranks and nDCG@100 for deeper ranked lists. As mentioned above, we also adopt Recall@100 to investigate the performance of QPP when evaluating first-stage CS retrievers.

Implementation details. We implement all QPP methods using Pytorch.[5] For unsupervised QPP methods, we use hyperparameters that have been shown to be effective by previous studies. Following [300], k is set to 5 for WIG. As suggested by [224, 239], k is set to 100 for NQC and SMV; following [239], we use the average retrieval score of the top-1000 documents as the corpus score $Score(q; D)$. Following [60], we set x to 50 for n($\sigma_{x\%}$). σ_{max} does not use any hyperparameters. Following [224], we use the Clarity variant that uses the sum-normalized retrieval scores (from BM25 or ConvDR in our setting) for weighing documents when constructing a relevance model [131]; our preliminary experiments showed that this variant performed better than the original Clarity that uses query-likelihood scores to weight documents; we induce the relevance model using the top 100 documents and clip the relevance model at the top-100 terms cutoff [223].

For all supervised QPP methods, we use bert-base-uncased,[6] a fixed learning rate (0.00002), and the Adam optimizer [124]. All methods are trained and inferred on an NVIDIA RTX A6000 GPU. Following [164, 282], all training on CAsT-19 or CAsT-20 uses five-fold cross-validation; we use the data split from [282] and train all supervised QPP methods for 5 epochs. For training on OR-QuAC, we train all QPP methods for 1 epoch on the training set of OR-QuAC; we feed QPP methods with human-rewritten queries and train them to estimate the retrieval quality of BM25 with human-rewritten queries. To address the data scarcity on CAsT-19 and CAsT-20, we consider a *warm-up* setting where we first pre-train supervised QPP methods on the training set of OR-QuAC for one epoch, followed by the five-fold cross-validation training for 5 epochs on CAsT.

4.4 Results and Discussions

Our experiments revolve around three main findings from the literature on QPP for ad-hoc search: (i) supervised QPP methods outperform unsupervised QPP methods [15, 45, 65, 67, 102, 283]; (ii) list-wise supervised QPP methods outperform their point-wise counterparts [45, 67]; and (iii) retrieval score-based unsupervised QPP methods perform poorly in estimating the retrieval quality of neural-based retrievers [66, 102]. We study whether the findings listed above continue to hold for QPP methods in CS.

4.4.1 Assessing query rewriting-based retrieval

Overall performance

To answer **RQ3.1**, we examine the results of Experiment **E1**, where we run QPP methods estimating the retrieval quality of BM25 with three query rewriting methods (T5+BM25,

[5] https://pytorch.org/

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

QuReTeC+BM25, and Human+BM25). For all supervised QPP methods on CAsT, we further consider their variants that are first pre-trained on the training set of OR-QuAC for one epoch before five-fold cross-validation training on CAsT. See Table 4.3. Note that QPP methods and BM25 always share the same query rewrites. Overall, feeding T5/QuReTeC query rewrites into QPP methods to estimate the retrieval quality of BM25 is effective, compared to the case of feeding perfect self-contained queries rewritten by humans. We have two specific observations.

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

Second, in most cases, point-wise supervised QPP methods such as NQA-QPP and BERTQPP outperform the list-wise supervised method qppBERT-PL. Without considering warming up, qppBERT-PL has a slight advantage over its point-wise counterparts. E.g., qppBERT-PL achieves a better performance in predicting the performance of QuReTeC+BM25, Human+BM25 on CAsT-19, and T5+BM25, QuReTeC+BM25 on CAsT-20. qppBERT-PL's list-wise training scheme learns from interactions between a query and all documents in a ranked list, providing the model with more training signals and better use of limited training data.

Turn-wise QPP effectiveness

We study the QPP effectiveness on each turn of conversation on CAsT-19; we report the turn-wise effectiveness of 2 unsupervised (WIG, NQC) and 2 supervised methods (NQA-QPP with warm-up, BERT-QPP with warm-up) when they assess BM25 with T5-based and human-written query rewrites. The results are presented in Figure 4.1.

Table 4.3: QPP quality in predicting the retrieval quality (nDCG@3) of BM25 fed with T5-, QuReTeC-, and human-written query rewrites. QPP methods take as input the same type of query rewrites as those used by BM25. QPP is evaluated by Pearson's ρ and Kendall's τ. *Warm-up* indicates one epoch of pre-training on the OR-QuAC training set. All coefficients are statistically significant (t-test, $p < 0.05$) except the ones in *italics*. **Bold** and underlined denote the best and second-best values per column.

		Assessing BM25 fed with different query rewrites					
		T5		QuReTeC		Human	
	QPP methods	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
CAsT-19	Clarity	0.321	0.234	0.327	0.211	0.359	0.231
	WIG	0.436	0.232	0.354	0.250	0.409	0.293
	NQC	0.348	0.246	0.286	0.190	0.334	0.234
	σ_{max}	0.442	0.354	0.351	0.251	0.410	**0.312**
	n($\sigma_{x\%}$)	0.430	0.332	0.348	0.259	0.407	0.307
	SMV	0.344	0.250	0.289	0.188	0.326	0.230
	NQA-QPP	0.188	*0.047*	*-0.016*	*0.010*	0.152	*0.069*
	BERTQPP	0.440	0.307	0.352	0.272	0.270	0.188
	qppBERT-PL	0.414	0.296	0.392	0.298	0.292	0.196
	NQA-QPP (warm-up)	**0.538**	**0.357**	**0.420**	**0.301**	0.331	0.230
	BERTQPP (warm-up)	0.526	**0.357**	0.369	0.264	**0.418**	0.282
	qppBERT-PL (warm-up)	0.317	0.218	0.330	0.232	0.297	0.190
CAsT-20	Clarity	0.258	0.191	*0.099*	*0.061*	*0.127*	*0.089*
	WIG	0.248	**0.251**	0.245	0.163	0.307	0.222
	NQC	0.150	0.235	0.198	0.189	0.286	**0.266**
	σ_{max}	0.179	0.221	0.207	0.168	0.241	0.199
	n($\sigma_{x\%}$)	0.178	0.225	0.182	0.133	0.213	0.167
	SMV	0.139	0.219	0.189	0.163	0.264	0.260
	NQA-QPP	*0.001*	*0.067*	*-0.064*	*-0.082*	*0.086*	*-0.011*
	BERTQPP	*0.042*	*-0.009*	0.172	0.145	0.194	0.110
	qppBERT-PL	*0.131*	0.125	0.175	0.150	*0.043*	*0.015*
	NQA-QPP (warm-up)	**0.274**	0.170	0.190	0.149	0.231	0.155
	BERTQPP (warm-up)	0.207	0.171	**0.403**	**0.301**	**0.336**	0.227
	qppBERT-PL (warm-up)	0.228	0.213	0.317	0.268	*0.094*	*0.095*
OR-QuAC	Clarity	0.090	0.085	0.110	0.103	0.076	0.069
	WIG	0.247	0.235	0.290	0.270	0.257	0.241
	NQC	0.251	0.274	0.290	0.311	0.276	0.291
	σ_{max}	0.317	0.279	0.367	0.316	0.412	0.367
	n($\sigma_{x\%}$)	0.181	0.172	0.229	0.209	0.245	0.193
	SMV	0.204	0.239	0.239	0.273	0.194	0.232
	NQA-QPP	**0.781**	**0.566**	**0.792**	**0.591**	**0.809**	**0.621**
	BERTQPP	0.678	0.434	0.692	0.476	0.725	0.527
	qppBERT-PL	0.594	0.507	0.617	0.526	0.618	0.525

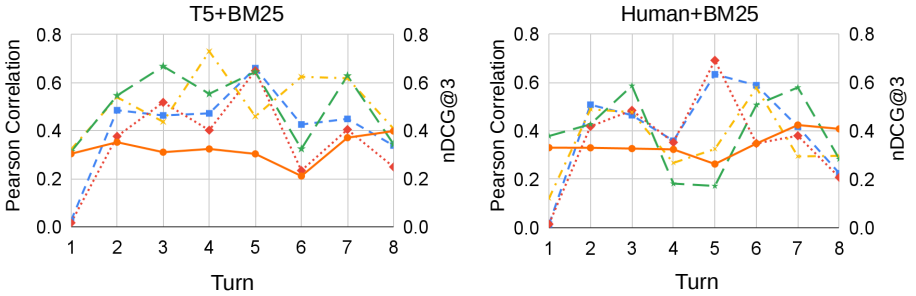


Figure 4.1: QPP effectiveness in predicting the retrieval quality of BM25 with T5-generated and human-written query rewrites at each turn of conversations in CAsT-19. Pearson's r correlation between the actual nDCG@3 scores of the queries with the same turn number and their estimated retrieval quality is calculated per turn.

We also introduce the turn-wise actual retrieval quality in terms of nDCG@3 in each subfigure. As illustrated in both subfigures, all QPP methods exhibit lower performance at the first turn and at the deeper turn 8. There is a correlation between actual retrieval quality and QPP effectiveness: BERT-QPP effectiveness always drops as the actual retrieval quality drops; in contrast, in the case of T5+BM25, NQA-QPP performs better as the actual retrieval quality drops at turn 6; in the case of human+BM25, WIG and NQC show better performance as the actual retrieval quality drops at turn 5.

4.4.2 Assessing conversational dense retrieval

Overall performance

To answer **RQ3.2**, we examine the results of **E2**. We apply QPP methods fed with three types of query rewrites to estimate the retrieval quality of the conversational dense retrieval method ConvDR. See Table 4.4. Note that the results of NQC, σ_{max} and SMV are invariant to different types of query rewrites because they only depend on retrieval scores; Clarity is also invariant to query rewrites because we use the Clarity variant from [224]; see Section 4.3.3 for more information about implementation details. We have four main observations.

First, retrieval score-based methods NQC/WIG show high effectiveness in estimating the retrieval quality of ConvDR, achieving the best performance in most cases on CAsT-19 and CAsT-20. Compared to Table 4.3, the performance of NQC/WIG is even better than their effectiveness in assessing BM25. It contradicts the previous findings [66, 102]: Datta et al. [66] found that the retrieval scores from neural-based retrievers, such as ColBERT [120], are restricted within a shorter range compared to lexical-based retrievers, which may limit the performance of score-based unsupervised QPP methods. We speculate that there are two reasons. First, the effectiveness of score-based methods depends on the retrieval score distribution of a specific retriever, regardless of whether they assess a lexical-based or a neural-based retriever. Figure 4.2 illustrates the retrieval score distributions of ConvDR and BM25 with three kinds of query rewrites in the three datasets. The retrieval score distribution of ConvDR displays

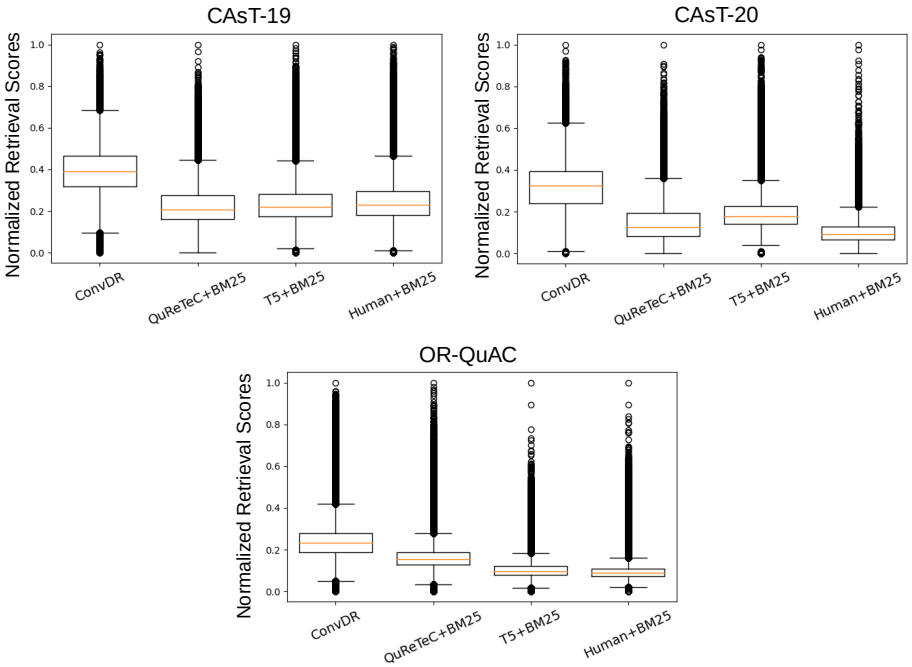


Figure 4.2: Distributions of retrieval scores for ConvDR and BM25 with three different rewriters on the three datasets. For the sake of comparison, we normalize the retrieval scores of a pipeline for all queries in a dataset by min-max normalization.

a higher variance. A higher standard deviation indicates that the score ranges vary more, and so the top-ranked documents are more distinguishable from the rest. Thus, ConvDR has a higher potential to be predicted more accurately using score-based QPP methods. Second, as discussed in Section 4.4.1, score-based QPP methods do not depend on the input queries and tend to be less impacted by the query understanding challenge in CS. Thus, score-based unsupervised methods show more effectiveness when assessing ConvDR compared to other supervised methods.

Second, supervised QPP methods tend to exhibit better performance when fed with human-written query rewrites, especially on CAsT-20, where query rewriting is much harder than CAsT-19. It highlights the importance of query rewriting quality.

Third, similar to our results for **RQ3.1**, supervised QPP methods distinctly outperform all unsupervised QPP methods on the OR-QuAC dataset where a large amount of training data is available. NQA-QPP remains the state-of-the-art method on OR-QuAC.

Fourth, as with the results for **RQ3.1**, point-wise supervised methods outperform qppBERT-PL in most cases (on CAsT-20 and OR-QuAC). On CAsT-19, qppBERT-PL trained using five-fold cross-validation outperforms its point-wise counterparts warming up from OR-QuAC, showing its potential in a few-shot setting.

Table 4.4: QPP quality in predicting the retrieval quality, in terms of nDCG@3, of ConvDR. QPP methods take three types of query rewrites as input: T5-based, QuReTeC-based, and human-written. QPP quality is measured by Pearson's ρ and Kendall's τ correlation coefficients. *Warm-up* means the QPP method is first pre-trained on the training set of OR-QuAC for one epoch. All coefficients are statistically significant (t-test, $p < 0.05$) except the ones in *italics*. The best value in each column is marked in **bold**, and the second best is underlined.

		Assessing ConvDR with different QPP inputs					
		T5		QuReTeC		Human	
	QPP methods	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
CAsT-19	Clarity	0.257	0.176	0.257	0.176	0.257	0.176
	WIG	0.387	0.274	0.388	0.266	0.412	0.285
	NQC	**0.431**	**0.307**	**0.431**	**0.307**	**0.431**	**0.307**
	σ_{max}	0.378	0.267	0.378	0.267	0.378	0.267
	n($\sigma_{x\%}$)	0.187	0.175	0.181	0.170	0.216	0.196
	SMV	0.386	0.285	0.386	0.285	0.386	0.285
	NQA-QPP	*0.121*	*0.075*	*0.118*	*0.073*	0.150	0.109
	BERTQPP	0.167	0.107	0.220	0.145	0.298	0.193
	qppBERT-PL	0.344	0.225	0.316	0.197	0.276	0.178
	NQA-QPP (warm-up)	0.187	0.128	0.161	0.107	0.287	0.191
	BERTQPP (warm-up)	0.282	0.187	0.234	0.157	0.371	0.251
	qppBERT-PL (warm-up)	0.212	0.151	0.167	0.117	0.172	0.115
CAsT-20	Clarity	*0.126*	*0.088*	*0.126*	*0.088*	*0.126*	*0.088*
	WIG	**0.377**	**0.277**	**0.377**	**0.263**	0.384	0.264
	NQC	0.339	0.261	0.339	0.261	0.339	0.261
	σ_{max}	0.282	0.219	0.282	0.219	0.282	0.219
	n($\sigma_{x\%}$)	0.199	0.168	0.197	0.156	0.201	0.156
	SMV	0.275	0.216	0.275	0.216	0.275	0.216
	NQA-QPP	*-0.037*	*-0.037*	*-0.081*	*-0.063*	*0.059*	*0.023*
	BERTQPP	0.223	0.157	0.216	0.146	**0.404**	**0.281**
	qppBERT-PL	0.185	0.144	*0.029*	*0.023*	0.251	0.171
	NQA-QPP (warm-up)	0.315	0.218	0.240	0.178	0.374	0.267
	BERTQPP (warm-up)	0.253	0.183	0.320	0.236	0.349	0.244
	qppBERT-PL (warm-up)	0.218	0.164	0.140	0.115	0.348	0.268
OR-QuAC	Clarity	-0.050	-0.029	-0.050	-0.029	-0.050	-0.029
	WIG	0.137	0.107	0.116	0.088	0.140	0.111
	NQC	0.227	0.163	0.227	0.163	0.227	0.163
	σ_{max}	0.442	0.339	0.442	0.339	0.442	0.339
	n($\sigma_{x\%}$)	-0.032	*-0.003*	-0.073	-0.035	*-0.022*	*0.008*
	SMV	0.098	0.076	0.098	0.076	0.098	0.076
	NQA-QPP	**0.615**	**0.479**	**0.639**	**0.499**	**0.600**	**0.470**
	BERTQPP	0.481	0.417	0.505	0.435	0.481	0.408
	qppBERT-PL	0.391	0.250	0.424	0.294	0.437	0.306

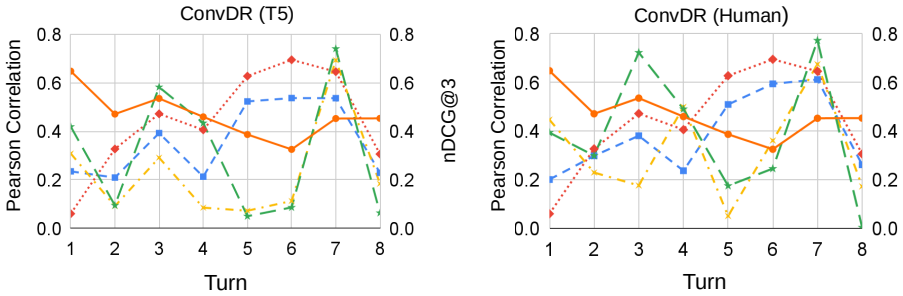


Figure 4.3: QPP effectiveness in predicting the retrieval quality of ConvDR when QPP methods are given T5-generated and human-written query rewrites as input at each turn of conversations in CAsT-19. Pearson's r correlation between the actual nDCG@3 scores of the queries with the same turn number and their estimated retrieval quality is calculated per turn.

Turn-wise QPP effectiveness

Similar to Section 4.4.1, here we report the turn-wise effectiveness of the same QPP methods when they are fed with T5-based and human-written query rewrites to assess ConvDR. See Figure 4.3. As shown in both subfigures, the effectiveness of the score-based unsupervised methods (WIG/NQC) first exhibits lower performance at the first turn, and then shows an upward trend as conversations go on. In contrast, in the middle of conversations, the supervised QPP methods are more sensitive to the actual retrieval quality; their effectiveness drops sharply as the actual retrieval quality drops. Especially, NQA-QPP/BERT-QPP effectiveness shows a more dramatic drop from turn 4 to 6 when they are fed with T5-based query rewrites, compared to when they are fed with human-written ones. It shows the importance of improving query rewriting quality again. Interestingly, there is a sharp drop from turn 7 to 8 for all QPP methods, showing the QPP difficulty at deeper turns.

4.4.3 Top ranks vs. deeper ranked lists

To answer **RQ3.3**, we report the results of **E3** in Tables 4.5 and 4.6, i.e., predicting retrieval quality, in terms of nDCG@3, nDCG@100, and Recall@100, of T5+BM25 and ConvDR. Due to space limitations, for supervised QPP methods, we only show them in the warm-up setting. Since qppBERT-PL works better without warm-up, we consider it both with and without a warm-up round. We have three main observations.

First, all QPP methods generally perform better when predicting the retrieval quality for deeper-ranked lists, i.e., estimating the retrieval quality for top ranks is harder than for deeper-ranked lists. The estimated performance by various QPP methods achieves a higher correlation with the actual nDCG@100/Recall@100 values in comparison with the nDCG@3 values, which is in line with [283], that found predicting NDCG@20 to be harder than AP@1000.

Second, unsupervised QPP methods get a higher correlation with nDCG@100 and Recall@100 on CAsT-19 and CAsT-20, showing high effectiveness in estimating the

Table 4.5: QPP quality in predicting the retrieval quality, in terms of nDCG@3, nDCG@100 and Recall@100, of BM25 fed with T5-based query rewrites. QPP quality is measured by Pearson's ρ and Kendall's τ correlation coefficients. *Warm-up* means the QPP method is first pre-trained on the training set of OR-QuAC for one epoch. All coefficients are statistically significant (t-test, $p < 0.05$) except the ones in *italics*. The best value in each column is marked in **bold**, and the second best is underlined.

		Assessing T5 + BM25					
		nDCG@3		nDCG@100		Recall@100	
	QPP methods	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
CAsT-19	Clarity	0.321	0.234	0.326	0.257	0.214	0.187
	WIG	0.436	0.232	**0.608**	0.429	**0.579**	0.426
	NQC	0.348	0.246	0.548	0.397	0.545	0.444
	σ_{max}	0.442	0.354	0.574	**0.433**	0.494	0.399
	n($\sigma_{x\%}$)	0.430	0.332	0.569	0.406	0.505	0.365
	SMV	0.344	0.250	0.548	0.417	0.541	**0.466**
	NQA-QPP (warm-up)	**0.538**	**0.357**	0.542	0.392	0.537	0.377
	BERTQPP (warm-up)	0.526	**0.357**	0.532	0.391	0.463	0.325
	qppBERT-PL (warm-up)	0.317	0.218	0.412	0.279	0.363	0.263
	qppBERT-PL	0.414	0.296	0.509	0.358	0.452	0.312
CAsT-20	Clarity	0.258	0.191	0.452	0.343	0.467	0.332
	WIG	0.248	**0.251**	**0.494**	**0.453**	**0.478**	**0.438**
	NQC	0.150	0.235	0.363	0.399	0.320	0.380
	σ_{max}	0.179	0.221	0.339	0.372	0.339	0.382
	n($\sigma_{x\%}$)	0.178	0.225	0.413	0.422	0.420	0.410
	SMV	0.139	0.219	0.362	0.400	0.333	0.387
	NQA-QPP (warm-up)	**0.274**	0.170	0.471	0.362	0.466	0.370
	BERTQPP (warm-up)	0.207	0.171	0.404	0.301	0.364	0.246
	qppBERT-PL (warm-up)	0.228	0.213	0.367	0.305	0.312	0.287
	qppBERT-PL	*0.131*	0.125	0.310	0.251	0.363	0.275
OR-QuAC	Clarity	0.090	0.085	0.197	0.196	0.362	0.312
	WIG	0.247	0.235	0.376	0.370	0.482	0.450
	NQC	0.251	0.274	0.356	0.409	0.414	0.461
	σ_{max}	0.317	0.279	0.418	0.393	0.438	0.437
	n($\sigma_{x\%}$)	0.181	0.172	0.295	0.302	0.415	0.401
	SMV	0.204	0.239	0.311	0.383	0.396	0.456
	NQA-QPP	**0.781**	**0.566**	**0.783**	**0.602**	**0.603**	**0.486**
	BERTQPP	0.678	0.434	0.767	0.551	0.589	0.484
	qppBERT-PL	0.594	0.507	0.655	0.552	0.451	0.440

retrieval quality of deeper ranked lists. On OR-QuAC, where training data is ample, supervised QPP methods still keep the lead in terms of all metrics, in line with the results shown in Table 4.3 and Table 4.4.

Table 4.6: QPP quality in predicting the retrieval quality, in terms of nDCG@3, nDCG@100 and Recall@100, of ConvDR. All QPP methods take T5-generated query rewrites as input. QPP quality is measured by Pearson's ρ and Kendall's τ correlation coefficients. *Warm-up* means the QPP method is first pre-trained on the training set of OR-QuAC for one epoch. All coefficients are statistically significant (t-test, $p < 0.05$) except the ones in *italics*. The best value in each column is marked in **bold**, and the second best is underlined.

		Assessing ConvDR (QPP using T5 query rewrites)					
		nDCG@3		nDCG@100		Recall@100	
	QPP methods	P-ρ	K-τ	P-ρ	K-τ	P-ρ	K-τ
CAsT-19	Clarity	0.257	0.176	0.342	0.227	0.335	0.216
	WIG	0.387	0.274	0.542	0.398	0.451	0.347
	NQC	**0.431**	**0.307**	**0.647**	**0.481**	0.557	0.421
	σ_{max}	0.378	0.267	0.637	0.456	**0.591**	**0.441**
	n($\sigma_{x\%}$)	0.187	0.175	0.358	0.292	0.362	0.288
	SMV	0.386	0.285	0.619	0.471	0.556	0.423
	NQA-QPP (warm-up)	0.187	0.128	0.401	0.275	0.364	0.263
	BERTQPP (warm-up)	0.282	0.187	0.378	0.249	0.261	0.194
	qppBERT-PL (warm-up)	0.212	0.151	0.354	0.233	0.345	0.249
	qppBERT-PL	0.344	0.225	0.461	0.310	0.455	0.327
CAsT-20	Clarity	*0.126*	*0.088*	0.270	0.195	0.264	0.178
	WIG	**0.377**	**0.277**	**0.549**	0.389	**0.465**	0.320
	NQC	0.339	0.261	0.544	**0.404**	0.463	**0.357**
	σ_{max}	0.282	0.219	0.496	0.364	0.440	0.328
	n($\sigma_{x\%}$)	0.199	0.168	0.409	0.309	0.397	0.285
	SMV	0.275	0.216	0.503	0.380	0.454	0.352
	NQA-QPP (warm-up)	0.315	0.218	0.310	0.237	0.324	0.223
	BERTQPP (warm-up)	0.253	0.183	0.349	0.242	0.221	0.133
	qppBERT-PL (warm-up)	0.218	0.164	0.378	0.272	0.313	0.229
	qppBERT-PL	0.185	0.144	0.301	0.217	0.263	0.196
OR-QuAC	Clarity	-0.050	-0.029	-0.029	-0.015	0.053	0.057
	WIG	0.137	0.107	0.195	0.130	0.298	0.261
	NQC	0.227	0.163	0.302	0.194	0.402	0.333
	σ_{max}	0.442	0.339	0.490	0.359	0.434	0.370
	n($\sigma_{x\%}$)	-0.032	*-0.003*	*-0.001*	*0.010*	0.102	0.106
	SMV	0.098	0.076	0.170	0.109	0.313	0.277
	NQA-QPP	**0.615**	**0.479**	**0.644**	**0.475**	0.446	0.323
	BERTQPP	0.481	0.417	0.595	0.453	0.447	0.313
	qppBERT-PL	0.391	0.250	0.449	0.277	**0.455**	**0.383**

Third, in some cases, list-wise supervised methods outperform their point-wise counterparts when estimating the retrieval quality in terms of deeper ranked lists. E.g., qppBERT-PL without warm-up outperforms other point-wise methods (NQA-QPP and

BERTQPP with warm-up) on CAsT-19 when assessing ConvDR in terms of nDCG@100 and Recall@100. Also, qppBERT-PL achieves the best performance when predicting the performance of ConvDR in terms of Recall@100 on OR-QuAC. The gains indicate that modeling a list of retrieved items has the potential of benefiting the retrieval quality estimation for deeper-ranked lists.

4.5 Related Work

This chapter builds on two strands of research: query performance prediction and conversational search.

4.5.1 Query performance prediction

The query performance prediction (QPP) task is to estimate the retrieval quality of a search system in response to a user query without relevance judgments [39, 105]. QPP methods have shown a high correlation with the retrieval quality in the context of ad-hoc retrieval. They can help to obtain better-performing retrieval pipelines in different ways, including query routing [215]. Moreover, query difficulty signals have been used to provide direct feedback to users, allowing them to reformulate queries or seek alternative information sources if the results are expected to be poor.

Typically, QPP methods can be classified into pre- and post-retrieval methods [39]. Pre-retrieval methods estimate query performance based on the query and corpus statistics before retrieval takes place. Post-retrieval methods use additional information from the ranked list to predict query performance after retrieval. In this chapter, we focus on post-retrieval QPP methods because they have shown superior performance compared to pre-retrieval methods in most cases. Post-retrieval QPP methods include both supervised and unsupervised methods.

Traditional QPP methods have mostly relied on an unsupervised approach where query term frequency and corpus statistics are used as indicators for query performance [103–106, 223, 224, 300]. More recent studies model QPP by deep learning-based models. These studies have shown that supervised methods for QPP are more effective than unsupervised QPP approaches in an ad-hoc retrieval setting. These supervised methods require a significant amount of data and training instances, such as the MS MARCO dataset [32], to perform QPP effectively [15, 67, 102, 283]. To the best of our knowledge, QPP has mostly been limited to ad-hoc retrieval tasks. Hashemi et al. [102] explore the ability of QPP methods to predict performance for non-factoid question answering. Studies of the performance of QPP methods in CS, have been limited.

4.5.2 Conversational search

Conversational search (CS) is the task of retrieving relevant passages in response to user queries in a multi-turn conversation [61, 62, 115]. A unique challenge in CS is that a user query in a conversation is context-dependent, i.e., it may contain omissions, coreferences, or ambiguities, making it challenging for ad-hoc search methods to capture the underlying information need [203]. Recovering the underlying information

need from the conversational history is crucial [164]. To address this challenge, there are two main groups of CS methods, namely, *query-rewriting-based retrieval* and *conversational dense retrieval*. Query-rewriting-based retrieval methods first rewrite a query that is part of a conversation into a self-contained query and then feed it to an ad-hoc retriever [143, 169, 252, 257, 270, 281]. Query rewriting can be conducted by either term expansion or sequence generation. The former adds terms from the conversational history to the current query, e.g., by designing rules [169] or training a binary term classifier [252], while the latter directly generates the reformulated queries using pre-trained generative language models, e.g., GPT-2 [281] and T5 [143].

Conversational dense retrieval methods train a query encoder to encode the current query and the conversational history into a contextualized query embedding; the contextualized query embedding is expected to implicitly represent the information need of the current query in a latent space [123, 142, 164, 165, 201, 282]. Lin et al. [142] train the query encoder by optimizing a ranking loss over a large number of pseudo-relevance judgments. Yu et al. [282] train the query encoder to mimic the embeddings of human-written queries output by the query encoder of the ad-hoc dense retriever ANCE [272]. Mao et al. [164] train the query encoder to denoise noisy turns in the conversation history by contrastive learning.

Little research has been done into QPP for CS. Arabzadeh et al. [16], Roitman et al. [210] explore QPP in single-turn CS, where they use QPP to help a CS system take the next appropriate action given a user query. Specifically, they use QPP to assess the retrieved answer quality to determine whether the system should return the answer to the user. Al-Thani et al. [6], Lin et al. [143] use QPP to improve the retrieval quality of a CS system. Lin et al. [143] use a QPP method to determine whether the current query should be expanded with keywords from the previous turns. Al-Thani et al. [6] use QPP methods to select the better query rewrite from different ones. Meng et al. [174] investigate the performance of pre-retrieval QPP methods when they estimate the retrieval quality of BM25 fed with T5-generated query rewrites. Also, Meng et al. [174] propose to incorporate query rewriting quality to improve QPP effectiveness. Additionally, Vlachou and Macdonald [255] explore QPP in the context of conversational fashion recommendation, which differs from CS.

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

4.6 Conclusions and Future Work

In this chapter, we addressed the following thesis-level research question:

RQ3 How can QPP methods originally designed for ad-hoc search be effectively adapted to conversational search, and how well do QPP methods for ad-hoc search perform in conversational search?

To answer the first part of this question, we have proposed adapting existing QPP methods that heavily rely on input queries to address the challenge of context-dependent query understanding in conversational search. We achieve this adaptation by feeding

machine-generated self-contained query rewrites into QPP methods. This ensures that QPP methods remain aligned with their original design, continuing to process self-contained queries.

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

Our extensive experiments reveal that using query rewrites produced by T5 [143] or QuReTeC [257] as input to QPP methods leads to great QPP quality in predicting the performance of both query-rewriting-based retrieval and conversational dense retrieval methods. Our reproducibility study demonstrates that we found that the three findings on QPP for ad-hoc search do not generalize to CS very well. Specifically, we found (i) supervised QPP methods distinctly outperform their unsupervised counterparts only when a large amount of training data is available, while unsupervised QPP methods show strong performance when being in a few-shot setting and predicting the retrieval quality for deeper ranked lists; (ii) point-wise supervised QPP methods outperform their list-wise counterparts in most cases; however, list-wise QPP methods are more data-efficient, show a slight advantage in predicting the retrieval quality for deeper ranked lists; and (iii) retrieval score-based unsupervised QPP methods show high effectiveness in estimating the retrieval quality of a conversational dense retrieval method, ConvDR, either for top ranks or deeper ranked lists. One possible explanation is that these methods rely on the retrieval score distribution of a retriever. ConvDR's scores span a wider range and exhibit greater variance compared to BM25, potentially making it easier for these methods to differentiate between high- and low-quality rankings. Another explanation is that these methods bypass the challenges of query understanding in conversational search by operating directly on retrieval scores, rather than modeling context-dependent queries.

Our experimental results also identify next directions for modeling of QPP in CS. First, we observed that, in general, human-written rewrites (i.e., ground-truth rewrites) leads to to higher QPP quality than machine-generated ones. This shows the important role of query rewriting quality, highlighting the need for improved query rewriting techniques. It also shows the need to develop a mechanism of conversational context understanding for QPP methods to directly understand raw historical utterances. Second, we found that QPP methods tend to struggle more when predicting an evaluation metric with a shallow cut-off compared to deeper ones. This underscores the need to improve QPP methods for better prediction accuracy at shallow ranks. Third, we reveal that the data sparsity problem in CS severely reduces the performance of supervised QPP methods. Thus, designing QPP methods using few-shot learning techniques is one possible way.

We point to two limitations of this chapter, namely, (i) we only consider estimating the retrieval quality of one conversational dense retrieval method, and (ii) we only use correlation metrics to evaluate the performance of QPP methods. In future work, we plan to (i) consider more conversational dense retrieval methods such as CQE [142] as

well as other dense retrieval methods for CS, such as T5-based rewriter+ANCE [272], and (ii) introduce QPP-specific evaluation metrics, such as scaled Absolute Ranking Error (sARE) and scaled Mean Absolute Ranking Error (sMARE) [78, 80].

In the next chapter, we continue to focus on QPP as a key methodology for ranking result reflection within agentic workflows for information access. However, we shift our perspective to examine how using large language models (LLMs) can enhance QPP accuracy.

5

Query Performance Prediction with Large Language Models

In line with the previous chapter, this chapter continues to focus on query performance prediction (QPP), a fundamental and long-standing methodology for ranking result reflection within agentic workflows for information access. This chapter focuses specifically on enhancing QPP accuracy. Recent advances in large language models (LLMs) have led to state-of-the-art performance across various information retrieval (IR) and natural language processing (NLP) tasks [178], yet their potential for improving QPP accuracy remains unexplored. There are many possible ways to model QPP with LLMs, yet identifying an effective approach is an open and challenging problem. To bridge this gap, this chapter investigates the following thesis-level research question:

RQ4 How can LLMs be used to effectively enhance QPP accuracy?

5.1 Introduction

Query performance prediction (QPP), a.k.a. query difficulty prediction, has attracted the attention of the information retrieval (IR) community throughout the years [18–20, 39]. QPP aims to estimate the retrieval quality of a search system for a query without using human-labeled relevance judgments [78]. Effective QPP benefits various downstream applications [82], e.g., query variant selection [72, 216, 241], selective query expansion [13], IR system configuration selection [70, 244], enriching query features for learning-to-rank [160] , and query-specific pool depth prediction [90] to reduce human relevance judgment costs.

QPP methods can be applied in various domains and scenarios [82, 178]. 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 [77, 84] for conversational search [3, 146] and on recall for legal search [243]. However, existing QPP approaches typically predict only a single real-valued score that indicates the retrieval quality for a query [91] and do not require the predicted score to approximate a specific IR evaluation

This chapter is about to appear as C. Meng, N. Arabzadeh, A. Askari, M. Aliannejadi, and M. de Rijke. Query performance prediction using relevance judgments generated by large language models. *ACM Transactions on Information Systems (TOIS)*, to appear.

measure [17, 224, 226, 239, 300].

These properties results in two key limitations: (i) While predicted performance scores have been shown to correlate with some IR evaluation metrics [67, 91], 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 [101, 119]. Although some studies train regression-based QPP models to predict a specific IR evaluation measure [15, 45, 65, 102, 121], 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 **re**levance 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 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 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 [81, 97, 151, 156, 237, 242, 248] have shown the potential effectiveness of using LLMs to generate relevance judgments. Notably, prior research has shown that LLMs can even achieve accuracy comparable to human labelers [242]. 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., 44, 81, 151, 242, 247, 249, 289]; 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 [195, 196, 293]. Although MacAvaney and Soldaini [156] prompt small-scale open-source language models (e.g., Flan-T5 [50] 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 chapter, 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 [122, 212, 247, 289] 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 [198]. Also, incorporating in-context learning examples in few-shot prompting leads to high inference costs [144].

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) [69] 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 [52–55] 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 [272] and TAS-B [109], in terms of RR@10, a precision-oriented IR metric, and nDCG@10, an IR metric considering recall. See Sections 5.6.1 and 5.6.2.

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 5.7.1.

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 trade-off 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) [81]. See Section 5.7.2.

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 [153], 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 5.7.3.

To demonstrate the generalizability of QPP-GenRE to a new domain, we conduct experiments of applying QPP-GenRE to conversational search in a zero-shot manner. Specifically, we evaluate QPP-GenRE and baselines when predicting the performance of a conversational dense retriever [282] on the CAsT-19 [62, 115] and 20 [61] datasets. We found that QPP-GenRE consistently outperforms all baselines on both datasets, highlighting its good capability to generalize to a new domain. See Section 5.7.4.

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 5.7.5.

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 [234]. 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 5.7.6.

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 [84], 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 [234]). 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 [177]. 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 [234] 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 work 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 [86]. 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.

Contributions. Our main contributions in this chapter 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. QPP-GenRE can effectively harness the strong relevance prediction capabilities of LLMs, potentially leading to improved QPP accuracy.
- 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.

5.2 Related Work

This chapter is relevant to four strands of research: query performance prediction (QPP) (Section 5.2.1), zero/few-shot prompting and parameter-efficient fine-tuning (PEFT) for LLMs (Section 5.2.2), LLMs for generating relevance judgments (Section 5.2.3), and LLMs for re-ranking (Section 5.2.4).

5.2.1 Query performance prediction

Query performance prediction (QPP) has attracted lots of attention in the IR and NLP community and has been widely studied in ad-hoc search [67, 85, 86, 226], conversational search [83, 84], question answering [102, 213], and image retrieval [193]. This chapter focuses on QPP for ad-hoc search.

Typically, QPP methods are divided into two categories: pre- and post-retrieval methods [39]. 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 [39]. This chapter focuses on post-retrieval QPP methods.

A large number of unsupervised and supervised post-retrieval QPP methods have been proposed [39] for predicting the performance of lexical rankers, such as query likelihood [127] and BM25 [208]. Unsupervised QPP methods can be classified into clarity-based [56], robustness-based [26, 299, 300], coherence-based [14, 73], and score-based [60, 191, 224, 239, 300]. More recently, a set of supervised QPP methods have been proposed [15, 45, 65, 66, 102, 121, 283]. NeuralQPP [283] and DeepQPP [65] are optimized from scratch. NQA-QPP [102] and BERT-QPP [15] fine-tune BERT [71] to improve QPP effectiveness. Further, Datta et al. [67] propose qppBERT-PL, which considers list-wise-document information, while Chen et al. [45] propose BERT-groupwise-QPP that considers both cross-query and cross-document information. Khodabakhsh and Bagheri [121] 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 [86, 102], motivating several new unsupervised post-retrieval QPP methods designed for neural rankers. Datta et al. [66] propose a weighted relative information gain-based model (WRIG), 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. [287] assess a neural re-ranker by measuring the entropy of scores returned by it; Faggioli et al. [85] propose neural-ranker-specific ways of calculating regularization terms used by unsupervised post-retrieval QPP methods; Vlachou and MacDonald [256] propose an unsupervised coherence-based QPP method that employs neural embedding representations to assess dense retrievers; and Singh et al. [226] 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 [194]) 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 [15, 45, 65, 102, 121], 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 [67], 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) [181]. 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. [229] 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. [181] explore applying QPP EEwRJ [181] 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 [181]. QPP-GenRE does not face these limitations.

5.2.2 Zero/few-shot prompting and parameter-efficient fine-tuning for large language models

While fine-tuning pre-trained language models has given rise to many state-of-the-art results [71], fully fine-tuning LLMs for a specific task on consumer-level hardware is typically infeasible [301] 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, a.k.a. in-context learning (ICL) [35, 74], and parameter-efficient fine-tuning (PEFT) [69, 111, 144].

There is limited success in only prompting open-source LLMs for certain tasks [198]. 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 [198, 292], such as Flan-T5 [50], Flan-UL2 [240]. However, Sun et al. [232] 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 [74] 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 [144] 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 [94, 145, 295], has been shown to achieve comparable performance to full-model fine-tuning [69, 148]; LoRA adds learnable low-rank adapters to each network layer of an LLM [111] while all original parameters of the LLM are frozen. QLoRA [69] 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. This chapter explores the possibility of training open-source LLMs with QLoRA to generate relevance judgments.

5.2.3 Large language models for generating relevance judgments

Automatically generating relevance judgments is a long-standing goal in IR that has been studied for multiple decades [161, 162, 188, 189, 206, 229]. Recent studies have demonstrated promising results of using LLMs for the automatic generation of relevance judgments [81, 242]. In this chapter 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 [276]. 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. [81] use zero- and few-shot prompting to instruct GPT-3.5 to predict the relevance of an item to a query. Thomas et al. [242] instruct GPT-4 by zero-shot prompting, and add to the prompt a detailed query description and consider chain-of-thought [266]. Ma et al. [151] instruct GPT-3.5 to generate relevance judgments for a domain-specific scenario, i.e., legal case retrieval [154]; they use prompts specifically designed for this scenario. More recently, Upadhyay et al. [249] prompt GPT-4o in a zero-shot manner. Besides using commercial LLMs, only few studies [122, 212, 247, 289] explore prompting open-source LLMs to generate relevance judgments. E.g., Khramtsova et al. [122], Upadhyay et al. [247] and Salemi and Zamani [212] prompt Flan-T5 [50], Vicuña-7B [298] and Mistral [118], respectively, in either zero-shot or few-shot manners. MacAvaney and Soldaini [156] focus on a special scenario where a relevant item for a given query is already known and use Flan-T5 [50] 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 [81, 156, 242, 247], (ii) ranker selection [122], (iii) item selection and retrieval quality evaluation in retrieval-augmented generation (RAG) [212, 289] and (iv) retriever fine-tuning [151]. Concerning (i), recent studies [2, 81, 156, 242, 247] 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-labelled relevance judgments. As to (ii), given a pool of dense retrievers, Khramtsova et al. [122] 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. [289] 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. [289] observe that items selected via relevance prediction resulted in sub-optimal text generation quality. For retrieval quality evaluation, Salemi and Zamani [212] generate relevance judgments for retrieved candidate items and aggregate those judgments into a score. However, Salemi and Zamani [212] 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. [151] 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.

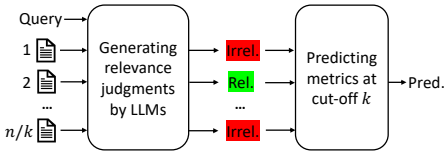
This chapter differs from the studies mentioned above: (i) we explore the possibility of *fine-tuning* open-source LLMs for generating relevance judgments; unlike MacAvaney and Soldaini [156], we focus on a more practical scenario wherein no relevant item is known in advance for each query; and (ii) 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.

5.2.4 Large language models for re-ranking

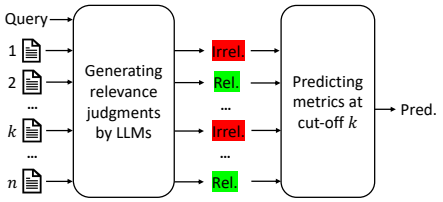
Recent studies on using LLMs for re-ranking have witnessed remarkable progress [23, 33, 75, 110, 152, 152, 153, 177, 195, 196, 211, 234, 238, 293, 302–304]. There are four paradigms of LLM-based re-ranking: pointwise, pairwise, listwise, and setwise [304]. Given a query, pointwise re-rankers produce a relevance score for each item independently, and the final ranking is formed by sorting items by relevance score [75, 153, 211, 302]. The pairwise paradigm [198] 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 [152, 195, 196, 234, 238, 293] 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 [152, 195, 196, 234, 238, 293]. Given the low efficiency of pairwise (multiple inference passes) and listwise (multiple decoding steps) re-rankers, the setwise paradigm [304] 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.

This chapter 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 chapter 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,



(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.

Figure 5.1: The framework of QPP-GenRE.

RankLLaMA [153], 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 chapter.

5.3 Task Definition

In this chapter, we focus on post-retrieval QPP [39]. 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:

$$p = \psi(q, L, C) \in \mathbb{R}, \tag{5.1}$$

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.

5.4 Method

5.4.1 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 5.1. 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's 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, covering precision-oriented metrics and metrics that consider recall.

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

Figure 5.2: Prompt used by LLMs for automatic generation of relevance judgments.

5.4.2 Predicting relevance judgments with large language models

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:

$$\hat{r}_i = \text{LLM}(\text{prompt}(q, d_i)), \tag{5.2}$$

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 5.2. We follow the design proposed by Faggioli et al. [81] 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. [234], 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" 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.

Parameter-efficient fine-tuning (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 [69]. 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:

$$\mathcal{L}(\theta_{LoRA}) = -\frac{1}{M} \sum_{i=1}^{M} \log P(r_i \mid \text{prompt}(q, d_i)), \tag{5.3}$$

where θ_{LoRA} stands for learnable low-rank adapters added to the LLM; M is the number of training examples. See Section 5.5.7 for more details.

5.4.3 Predicting precision-oriented metrics

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 5.1a. Note that in this case, $n = k$. The following is an example to compute RR@k:

$$RR@k = 1/\min_i\{\hat{r}_i > 0\}, \tag{5.4}$$

where $0 < i \leqslant k$. For instance, as illustrated in Figure 5.1a, 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.

5.4.4 An approximation strategy to predict metrics 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. [88], Lu et al. [147], Moffat [184] define normalized discounted cumulative gain (nDCG) [114] 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 [52–55]. Thus, here we show an example of predicting nDCG@k [114], formally:

$$nDCG@k = DCG@k/IDCG@k, \tag{5.5}$$

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

$$DCG@k = \hat{r}_1 + \sum_{i=2}^{k} \hat{r}_i/\log_2 i. \tag{5.6}$$

$IDCG@k$ 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 $\hat{\mathcal{R}}_{iL_{1:n}} = [\hat{ir}_1, \ldots, \hat{ir}_i, \ldots, \hat{ir}_n]$

[1] In this chapter, we employ nDCG@10 and believe that nDCG@10 is a metric considering recall: Figure 5.4 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.

in descending order of predicted relevance; then, we compute $IDCG@k$ based on $\hat{\mathcal{R}}_{iL_{1:n}}$, namely:

$$IDCG@k = \hat{ir}_1 + \sum_{i=2}^{k} \hat{ir}_i / \log_2 i. \tag{5.7}$$

5.5 Experimental Setup

5.5.1 Research questions

This chapter expands on the thesis-level research question **RQ4** by introducing the following chapter-specific research questions.

RQ4.1 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?

RQ4.2 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?

RQ4.3 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?

RQ4.4 To what extent do fine-tuning and the choice of LLMs affect the quality of generated relevance judgments and QPP?

5.5.2 Datasets

We experiment with 4 widely-used IR datasets from the TREC 2019–2022 deep learning (TREC-DL) tracks [52–55]. 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.

5.5.3 Retrieval approaches

We consider BM25 [208] as a lexical ranker; we also consider ANCE [272] and TAS-B [109] as neural-based dense retrievers. To increase the comparability and reproducibility of our chapter, we get the retrieval results of both rankers using the publicly available resource from Pyserini [140].[3] We get BM25's retrieval result with top-1000

[3] `https://github.com/castorini/pyserini`

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-1000 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.[4]

5.5.4 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 [56] computes the KL divergence between language models [131] induced from the top-k items in a ranked list and the corpus.
- Weighted information gain (WIG) [300] 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) [224] 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} [191] 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($\sigma_{x\%}$) [60] 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) [239] considers both the magnitude of retrieval scores (WIG) and their variance (NQC).
- UEF(NQC) [223] uses a pseudo-effective reference list to improve QPP quality; we follow [15, 17, 67] to use NQC as a base predictor.
- RLS(NQC) [209] generates and selects both pseudo-effective and pseudo-ineffective reference lists; we use NQC as a base predictor because Roitman [209] show that RLS works better with NQC.
- QPP-PRP [226] measures the degree to which a pairwise neural re-ranker (DuoT5 [194]) agrees with the ranked list for the query.

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

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:

Figure 5.3: Prompt used by QPP-LLM.

- Dense-QPP [17] 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 [17] 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 [15, 45, 67, 102] perform better than their neural-based counterparts, we only consider BERT-based supervised QPP approaches:

- NQA-QPP [102] 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 [15] 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 [67] 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 [121] 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 [145]. 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 5.3. We consider two variants:

- QPP-LLM (few-shot) uses in-context learning (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.

5.5.5 Evaluation

We follow established best practices [39, 56, 67, 103, 283] 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.

5.5.6 Target information retrieval evaluation measures

As for target IR metrics, we consider the two primary official IR metrics used in TREC DL 19–22 [52–55], RR@10 (precision-oriented) and nDCG@10 (considering recall); recent QPP studies [17, 85, 121] consider either or both of these metrics as their target metrics. Following [67], 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.

5.5.7 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 $\{5, 10, 15, 20, 25, 50, 100, 300, 500, 1000\}$. 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 [15, 102, 121] 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 [15, 102, 121], 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 [109, 272], 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

Table 5.1: 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 and 20. * indicates statistically significant correlation coefficients (p-value < 0.05). † indicates 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). The best value in each column is marked in **bold**. n denotes QPP-GenRE's judgment depth in a ranked list.

	Ranker: BM25			
	TREC-DL 19		TREC-DL 20	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.135	0.028	0.050	0.021
WIG	0.113	0.164	0.286*	0.218*
NQC	0.194	0.117	0.152	0.191
σ_{max}	0.195	0.164	0.200	0.211*
n($\sigma_{x\%}$)	0.144	0.181	0.187	0.123
SMV	0.141	0.097	0.126	0.193
UEF(NQC)	0.235	0.256*	0.270*	0.211*
RLS(NQC)	0.272	0.122	0.290*	0.193
QPP-PRP	0.292	0.189	0.163	0.184
NQA-QPP	0.181	0.122	0.062	0.069
BERTQPP	0.281	0.136	0.237	0.155
qppBERT-PL	0.145	0.138	0.166	0.152
M-QPPF	0.317*	0.208	0.335*	0.273*
QPP-LLM (few-shot)	0.008	0.003	-0.081	-0.129
QPP-LLM (fine-tuned)	0.171	0.158	0.228	0.206
QPP-GenRE ($n = 10$)	**0.538**$^{\dagger *}$	**0.486**$^{\dagger *}$	**0.560**$^{\dagger *}$	**0.475**$^{\dagger *}$

the consistency of the chapter, we train all supervised QPP methods (including QPP-GenRE) on the development set of MS MARCO V1 (6980 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,[5] a constant learning rate (0.00002), and the Adam optimizer [124].

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 2 demonstrations works best. For QPP-LLM (fine-tuned), we fine-tune LLaMA-7B using PEFT as QPP-GenRE fine-tunes LLMs.

[5] `https://github.com/huggingface/transformers`

Table 5.2: Continued from Table 5.1 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 21 and 22. n denotes QPP-GenRE's judgment depth in a ranked list.

	Ranker: BM25			
	TREC-DL 21		TREC-DL 22	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.183	0.161	0.253*	0.099
WIG	0.237	0.206*	0.029	0.082
NQC	0.227	0.195	0.223	0.048
σ_{max}	0.278*	0.174	0.038	0.048
n($\sigma_{x\%}$)	0.127	0.140	0.169	0.113
SMV	0.240	0.189	0.227*	0.094
UEF(NQC)	0.231	0.111	0.216	0.065
RLS(NQC)	0.234	0.195	0.224	0.095
QPP-PRP	-0.080	-0.017	0.122	0.091
NQA-QPP	0.161	0.163	0.224	0.177*
BERTQPP	0.206	0.134	0.148	0.122
qppBERT-PL	0.339*	0.244*	0.131	0.206*
M-QPPF	0.282*	0.209*	0.161	0.187*
QPP-LLM (few-shot)	-0.053	-0.053	-0.241	-0.155
QPP-LLM (fine-tuned)	0.030	0.099	-0.038	0.009
QPP-GenRE ($n = 10$)	**0.524**$^{\dagger *}$	**0.435**$^{\dagger *}$	**0.350**$^{\dagger *}$	**0.262**$^{\dagger *}$

We equip QPP-GenRE with an LLM for judging relevance. We use a recent PEFT method, 4-bit QLoRA [69], 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 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 (40GB).

One might argue why we choose QLoRA fine-tuning instead of distilling an oracle model into a smaller model. [234]. 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 chapter 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 5.5, a 1-billion-parameter

Table 5.3: Correlation coefficients (Pearson's ρ and Kendall's τ) between actual retrieval quality, in terms of RR@10, of ANCE and performance predicted by QPP-GenRE/baselines, on TREC-DL 19 and 20. * indicates statistically significant correlation coefficients (p-value < 0.05). † indicates 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). The best value in each column is marked in **bold**. n denotes QPP-GenRE's judgment depth in a ranked list.

	Ranker: ANCE			
	TREC-DL 19		TREC-DL 20	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	-0.078	-0.012	-0.074	-0.048
WIG	0.313*	0.228	0.059	0.048
NQC	0.350*	0.200	0.145	0.112
σ_{max}	0.384*	0.287*	0.171	0.118
n($\sigma_{x\%}$)	0.200	0.176	-0.008	0.022
SMV	0.352*	0.256*	0.182	0.161
UEF(NQC)	0.340*	0.260*	0.131	0.108
RLS(NQC)	0.359*	0.273*	0.178	0.139
QPP-PRP	0.259	0.246	0.100	-0.008
Dense-QPP	0.452*	0.280*	0.209	0.139
NQA-QPP	-0.026	-0.009	-0.059	-0.080
BERTQPP	0.330*	0.214	0.046	-0.012
qppBERT-PL	0.092	0.025	-0.224	-0.218
M-QPPF	0.292	0.200	0.068	0.038
QPP-LLM (few-shot)	-0.008	0.005	-0.226	-0.207
QPP-LLM (fine-tuned)	-0.073	0.011	-0.022	0.069
QPP-GenRE ($n = 10$)	**0.567**$^{\dagger *}$	**0.440**$^{\dagger *}$	**0.293**$^{\dagger *}$	**0.257**$^{\dagger *}$

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. 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.

Table 5.4: Continued from Table 5.3. Correlation coefficients (Pearson's ρ and Kendall's τ) between actual retrieval quality, in terms of RR@10, of TAS-B and performance predicted by QPP-GenRE/baselines, on TREC-DL 19 and 20.

	Ranker: TAS-B			
	TREC-DL 19		TREC-DL 20	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	-0.212	-0.148	0.148	0.133
WIG	-0.066	-0.125	0.024	0.020
NQC	0.248	0.213	0.260	0.194
σ_{max}	0.015	0.021	0.312*	0.245*
n($\sigma_{x\%}$)	-0.030	-0.079	0.080	0.086
SMV	0.249	0.205	0.263	0.198
UEF(NQC)	0.260	0.228	0.281*	0.213
RLS(NQC)	0.257	0.217	0.283*	0.217*
QPP-PRP	0.155	0.113	0.203	0.116
Dense-QPP	0.251	0.213	0.146	0.012
NQA-QPP	0.172	0.144	-0.058	-0.075
BERTQPP	0.202	0.194	0.077	0.037
qppBERT-PL	0.276	0.269	0.004	-0.002
M-QPPF	0.277	0.236	0.103	0.022
QPP-LLM (few-shot)	-0.080	0.002	0.054	-0.024
QPP-LLM (fine-tuned)	0.155	0.113	0.043	-0.020
QPP-GenRE ($n = 10$)	**0.538**$^{\dagger *}$	**0.481**$^{\dagger *}$	**0.356**$^{\dagger *}$	**0.289**$^{\dagger *}$

5.6 Results

5.6.1 Predicting a precision-oriented evaluation measure

To answer **RQ4.1**, 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. Tables 5.1 and 5.2 present the results of predicting BM25 retrieval performance on TREC-DL 19–20 and TREC-DL 21–22, respectively. Tables 5.3 and 5.4 report the results of predicting the retrieval performance of ANCE and TAS-B, respectively, on TREC-DL 2019 and 2020. Note that Dense-QPP is unable to predict the performance of BM25. 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 [226], 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

Table 5.5: 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 and 20. n denotes QPP-GenRE's judgment depth in a ranked list. * indicates statistically significant correlation coefficients (p-value < 0.05). † indicates 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 best value in each column is marked in **bold**.

	Ranker: BM25			
	TREC-DL 19		TREC-DL 20	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.091	0.056	0.358*	0.250*
WIG	0.520*	0.331*	0.615*	0.423*
NQC	0.468*	0.300*	0.508*	0.401*
σ_{max}	0.478*	0.327*	0.529*	0.440*
n($\sigma_{x\%}$)	0.532*	0.311*	0.622*	0.443*
SMV	0.376*	0.271*	0.463*	0.383*
UEF(NQC)	0.499*	0.322*	0.517*	0.356*
RLS(NQC)	0.469*	0.169	0.522*	0.376*
QPP-PRP	0.321	0.181	0.189	0.157
NQA-QPP	0.210	0.147	0.244	0.210*
BERTQPP	0.458*	0.207	0.426*	0.300*
qppBERT-PL	0.171	0.175	0.410*	0.279*
M-QPPF	0.404*	0.254*	0.435*	0.297*
QPP-LLM (few-shot)	-0.024	-0.031	0.167	0.138
QPP-LLM (fine-tuned)	0.313*	0.215	0.309*	0.254*
QPP-GenRE ($n = 200$)	**0.724**$^{\dagger *}$	0.474$^{\dagger *}$	**0.638**$^{\dagger *}$	**0.469**$^{\dagger *}$
QPP-GenRE ($n = 10$)	0.605*	**0.482***	0.490*	0.323*
QPP-GenRE ($n = 100$)	0.712*	0.472*	0.609*	0.457*
QPP-GenRE ($n = 1,000$)	0.715*	0.477*	0.627*	0.459*

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

Table 5.6: Continued from Table 5.5. 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 21 and 22.

	Ranker: BM25			
	TREC-DL 21		TREC-DL 22	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.137	0.078	0.202	0.090
WIG	0.311*	0.281*	0.350*	0.249*
NQC	0.134	0.221*	0.360*	0.156*
σ_{max}	0.298*	0.258*	0.142*	0.196*
n($\sigma_{x\%}$)	0.328*	0.234*	0.336*	0.228*
SMV	0.327*	0.236*	0.338*	0.155*
UEF(NQC)	0.153	0.232*	0.311*	0.145
RLS(NQC)	0.272*	0.223*	0.337*	0.157*
QPP-PRP	0.027	0.004	0.077	0.012
NQA-QPP	0.286*	0.201*	0.312*	0.194*
BERTQPP	0.351*	0.223*	0.369*	0.229*
qppBERT-PL	0.277*	0.182	0.300*	0.242*
M-QPPF	0.265	0.226*	0.345*	0.204*
QPP-LLM (few-shot)	0.238	0.201	-0.073	-0.077
QPP-LLM (fine-tuned)	0.264	0.198	-0.075	-0.009
QPP-GenRE ($n = 200$)	0.546†*	0.435†*	**0.388***	**0.251***
QPP-GenRE ($n = 10$)	0.462*	0.350*	0.316*	0.245*
QPP-GenRE ($n = 100$)	0.545*	0.427*	0.332*	0.246*
QPP-GenRE ($n = 1,000$)	**0.547***	**0.436***	**0.388***	**0.251***

5.6.2 Predicting an evaluation measure considering recall

To answer **RQ4.2**, we examine the performance of QPP-GenRE along with all the baselines on assessing BM25, ANCE and TAS-B in terms of nDCG@10. Tables 5.5 and 5.6 present the results of predicting BM25 retrieval performance on TREC-DL 19–20 and TREC-DL 21–22, respectively. Tables 5.7 and 5.8 report the results of predicting the retrieval performance of ANCE and TAS-B, respectively, on TREC-DL 19 and 20. 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 **RQ4.1**, 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.

Table 5.7: Correlation coefficients (Pearson's ρ and Kendall's τ) between actual retrieval quality, in terms of nDCG@10, of ANCE and performance predicted by QPP-GenRE/baselines, on TREC-DL 19 and 20. n denotes QPP-GenRE's judgment depth in a ranked list. * indicates statistically significant correlation coefficients (p-value < 0.05). † indicates 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 best value in each column is marked in **bold**.

	Ranker: ANCE			
	TREC-DL 19		TREC-DL 20	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	-0.088	-0.062	-0.091	-0.045
WIG	0.515*	0.368*	0.218	0.150
NQC	0.548*	0.372*	0.411*	0.290*
σ_{max}	0.455*	0.339*	0.403*	0.288*
n($\sigma_{x\%}$)	0.388*	0.315*	0.103	0.075
SMV	0.496*	0.359	0.380*	0.283*
UEF(NQC)	0.548*	0.372*	0.413*	0.290*
RLS(NQC)	0.466*	0.346*	0.333*	0.271*
QPP-PRP	0.129	0.049	0.216	0.121
Dense-QPP	0.565*	0.389*	0.419*	0.318*
NQA-QPP	0.089	-0.038	0.186	0.113
BERTQPP	0.222	0.117	0.137	0.089
qppBERT-PL	0.116	0.098	-0.119	-0.046
M-QPPF	0.287	0.160	0.225	0.177
QPP-LLM (few-shot)	0.136	0.120	-0.130	-0.094
QPP-LLM (fine-tuned)	0.203	0.117	0.081	0.097
QPP-GenRE ($n = 200$)	0.712$^{\dagger *}$	0.483$^{\dagger *}$	**0.457**$^{\dagger *}$	0.343$^{\dagger *}$
QPP-GenRE ($n = 10$)	0.624*	0.406*	0.306*	0.238*
QPP-GenRE ($n = 100$)	**0.719***	0.489*	0.456*	**0.355***
QPP-GenRE ($n = 1,000$)	**0.719***	**0.492***	0.447*	0.321*

5.7 Analysis

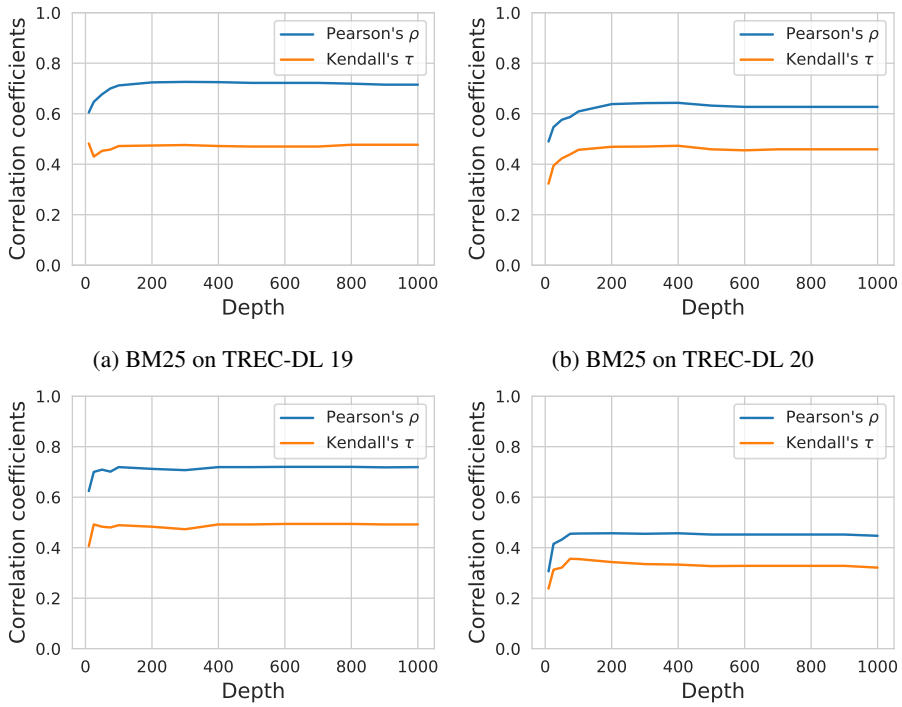
5.7.1 Judging depth analysis

RQ4.3 examines how varying the number of top-ranked documents submitted for relevance judgments impacts QPP quality. To answer **RQ4.3**, as detailed in Section 5.4.4, for predicting IDCG, we devise 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*

Table 5.8: Continued from Table 5.7. Correlation coefficients (Pearson's ρ and Kendall's τ) between actual retrieval quality, in terms of nDCG@10, of TAS-B and performance predicted by QPP-GenRE/baselines, on TREC-DL 19 and 20.

	Ranker: TAS-B			
	TREC-DL 19		TREC-DL 20	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.153	0.049	0.162	0.087
WIG	0.228	0.146	0.227	0.169
NQC	0.330*	0.233*	0.406*	0.264*
σ_{max}	0.220	0.126	0.428*	0.284*
n($\sigma_{x\%}$)	-0.008	-0.031	0.002	-0.020
SMV	0.349*	0.253*	0.425*	0.285*
UEF(NQC)	0.321*	0.246*	0.425*	0.271*
RLS(NQC)	0.314*	0.246*	0.404*	0.272*
QPP-PRP	0.220	0.126	0.267	0.237*
Dense-QPP	0.429*	0.244*	0.126	0.012
NQA-QPP	-0.020	0.060	0.031	0.024
BERTQPP	0.043	0.027	0.178	0.086
qppBERT-PL	0.304*	0.187	0.057	0.057
M-QPPF	0.163	0.051	0.304*	0.171
QPP-LLM (few-shot)	-0.020	0.060	0.108	0.048
QPP-LLM (fine-tuned)	0.262	0.195	0.162	0.111
QPP-GenRE ($n = 200$)	0.501†*	0.346†*	0.449*	0.315*
QPP-GenRE ($n = 10$)	0.490*	0.309*	0.421*	0.290*
QPP-GenRE ($n = 100$)	0.501*	0.336*	**0.450***	**0.317***
QPP-GenRE ($n = 1,000$)	**0.505***	**0.348***	0.449*	0.315*

depth of relevance judgment n do we need to consider to get a satisfactory performance for predicting nDCG@10? In Figure 5.4, 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 5.5, 5.6, 5.7 and 5.8. The tables 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 5.4 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



(a) BM25 on TREC-DL 19

(b) BM25 on TREC-DL 20

(c) ANCE on TREC-DL 19

(d) ANCE on TREC-DL 20

Figure 5.4: Relationship between the QPP effectiveness of predicting nDCG@10 and the judging depth for a ranked list.

speculate that this is because ANCE 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.

5.7.2 Impact of fine-tuning and the choice of large language models

To answer **RQ4.4**, 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 5.5.7), and (ii) few-shot prompted (in-context learning).[6] For the Llama family, besides LLaMA-7B [245], our evaluation includes Llama-3.2-1B-Instruct,

[6]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 5.9: 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 and 20. The best value in each column is marked in **bold**. We do not fine-tune Llama-3-70B-Instruct due to budget constraints.

	TREC-DL 19		TREC-DL 20	
LLM	κ	P-ρ	κ	P-ρ
Few-shot				
Llama-3.2-1B-Instruct	0.013	0.152	0.029	0.099
Llama-3.2-3B-Instruct	0.186	0.293	0.114	0.020
Mistral-7B-Instruct-v0.3	0.224	0.271	0.174	0.499
LLaMA-7B	-0.001	-0.062	-0.003	0.087
Llama-3-8B	0.018	0.042	0.027	0.087
Llama-3-8B-Instruct	0.315	0.510	0.227	0.372
Mistral-22B-Instruct	0.281	0.412	0.238	0.535
Llama-3-70B-Instruct	0.321	0.526	0.245	0.557
Fine-tuned				
Llama-3.2-1B-Instruct	0.351	0.610	0.211	0.596
Llama-3.2-3B-Instruct	0.383	0.710	0.273	0.722
Mistral-7B-Instruct-v0.3	0.403	0.734	0.328	0.720
LLaMA-7B	0.258	**0.715**	0.238	0.627
Llama-3-8B	0.381	0.544	**0.342**	0.681
Llama-3-8B-Instruct	0.397	0.647	0.316	**0.743**
Mistral-22B-Instruct	**0.407**	0.682	0.276	0.640

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).

We do not report the results for a zero-shot setting because our preliminary experiments show that zero-shot prompting 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. [81] 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.[7] The

[7]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. [81] are not available to us.

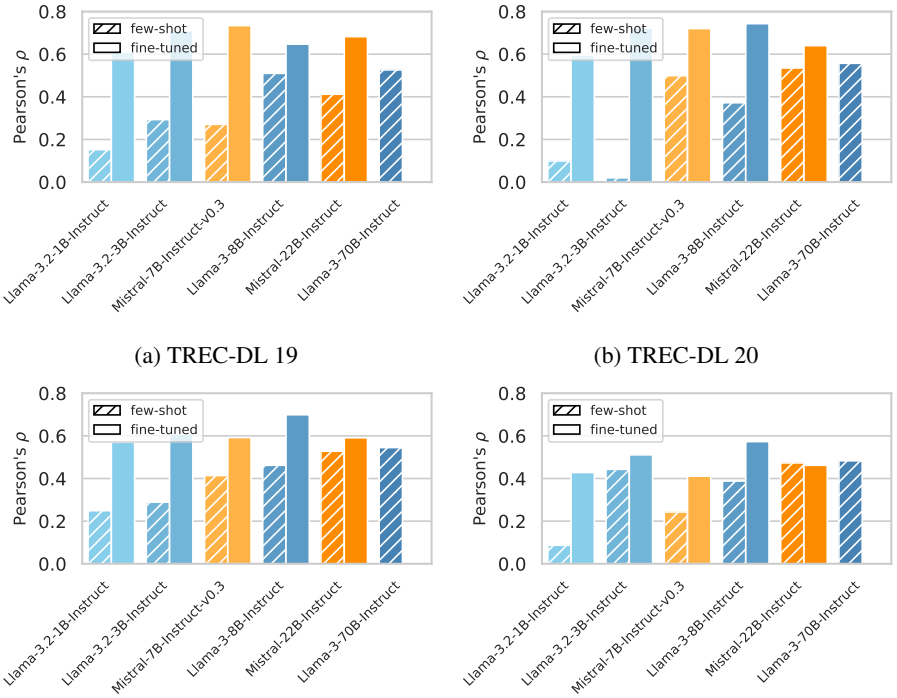
Table 5.10: Continued from Table 5.9. Relevance judgment agreement (Cohen's κ) between TREC assessors and each LLM, and Pearson's ρ correlation coefficients between BM25's actual nDCG@10 values and those predicted by QPP-GenRE integrated with each LLM on TREC-DL 21 and 22.

	TREC-DL 21		TREC-DL 22	
LLM	κ	P-ρ	κ	P-ρ
Few-shot				
GPT-3.5 (text-davinci-003) [81]	0.260	-	-	-
Llama-3.2-1B-Instruct	0.009	0.249	0.079	0.087
Llama-3.2-3B-Instruct	0.165	0.289	0.055	0.443
Mistral-7B-Instruct-v0.3	0.245	0.414	0.042	0.243
LLaMA-7B	0.003	-0.002	-0.010	0.214
Llama-3-8B	0.021	0.180	-0.035	0.087
Llama-3-8B-Instruct	0.238	0.462	0.049	0.388
Mistral-22B-Instruct	0.276	0.528	0.083	0.473
Llama-3-70B-Instruct	0.279	0.545	**0.086**	0.483
Fine-tuned				
Llama-3.2-1B-Instruct	0.197	0.570	0.042	0.428
Llama-3.2-3B-Instruct	0.306	0.608	0.042	0.511
Mistral-7B-Instruct-v0.3	0.373	0.592	0.076	0.411
LLaMA-7B	0.333	0.547	0.038	0.388
Llama-3-8B	0.347	0.612	0.082	0.568
Llama-3-8B-Instruct	**0.418**	**0.699**	0.066	**0.573**
Mistral-22B-Instruct	0.310	0.591	0.071	0.462

judging depth is set to 1000 in a ranked list. We present the results for TREC-DL 19 and 20 in Table 5.9 and for TREC-DL 21 and 22 in Table 5.10. Moreover, we provide a visual representation of the results in Figure 5.5.

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,[8] Llama-3-8B-Instruct (fine-tuned) even achieves "moderate" agreement on TREC-DL 21 (a 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

[8]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 [54]; 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

Figure 5.5: 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 colour 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.

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 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 3-billion-parameter model (Llama-3.2-3B-Instruct) offers the best trade-off between QPP quality and computational overhead: it not only markedly outperforms few-shot 70-billion-parameter LLMs, but also delivers QPP quality comparable to that of fine-tuned 7/8-billion-parameter LLMs.

5.7.3 Predicting relevance judgments with a large language model-based re-ranker

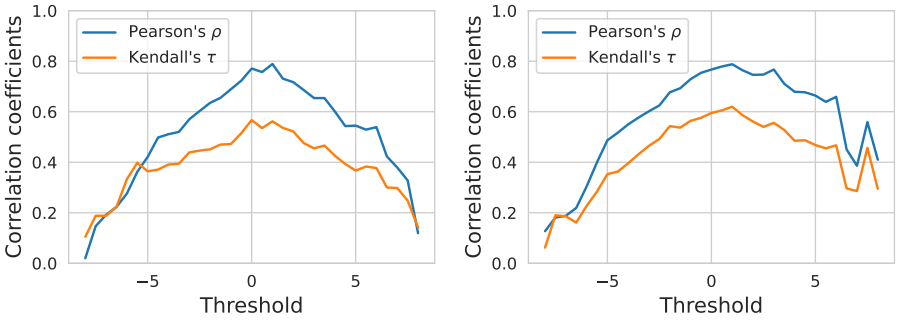
To show QPP-GenRE's compatibility with other types of relevance prediction methods instead of directly asking an LLM to explicitly generate explicit relevance judgments, we adapt a state-of-the-art pointwise LLM-based re-ranker, RankLLaMA [153], into a relevance judgment generator, and then integrate 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.[9] 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 5.6. 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).[10] This means that a state-of-the-art pointwise LLM-based re-ranker can be adapted into an effective relevance judgment generator. 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

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

[10] Note that the comparison is not fair because (i) LLMs and all other supervised QPP methods used in this chapter are trained on the development set of MS MARCO V1, while RankLLaMA [153] 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 [153] uses the Tevatron version of MS MARCO V1, where passages are enriched with document titles; Lassance and Clinchant [130] reveal that using titles leads to enhanced ranking performance.



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

Figure 5.6: QPP quality of QPP-GenRE integrated with RankLLaMA [153] 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.

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.

5.7.4 Generalization to conversational search

To assess the generalizability of QPP-GenRE to new domains, we apply it to the conversational search scenario [179, 183] in a zero-shot manner. We evaluate QPP-GenRE and other baselines on predicting the performance of ConvDR [282], a widely used conversational dense retriever. Given the findings in Section 5.7.2, 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 5.6.2). 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. [176] to provide QPP methods (including QPP-GenRE) with self-contained query rewrites as input. These rewrites are either generated by the T5 query generator[11] or written by humans. Table 5.11 presents the performance of QPP-GenRE along with all the baselines on assessing ConvDR [282] in terms of nDCG@3 on the CAsT-19 [62, 115] and 20 [61] datasets; QPP methods use T5-generated query rewrites. Table 5.12 reports the results when QPP methods use human-written query rewrites. Note that nDCG@3 is the primary evaluation metric officially adopted by TREC CAsT [61, 62, 115]. We have two main observations.

First, QPP-GenRE significantly outperforms all QPP baselines on both datasets

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

Table 5.11: Correlation coefficients (Pearson's ρ and Kendall's τ) between actual retrieval quality, in terms of nDCG@3, of ConvDR [282] and its performance predicted by QPP-GenRE/baselines, on CAsT-19 and 20. Following Meng et al. [176], for QPP methods requiring queries as input, we feed them with T5-generated query rewrites. * indicates statistically significant correlation coefficients (p-value < 0.05). † indicates 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 best value in each column is marked in **bold**. n denotes QPP-GenRE's judgment depth in a ranked list.

	Ranker: ConvDR (T5-generated)			
	CAsT-19		CAsT-20	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.257*	0.176*	0.126	0.088
WIG	0.387*	0.274*	0.377*	0.277*
NQC	0.431*	0.307*	0.339*	0.261*
σ_{max}	0.378*	0.267*	0.282*	0.219*
n($\sigma_{x\%}$)	0.187*	0.175*	0.199*	0.168*
SMV	0.386*	0.285*	0.275*	0.216*
UEF(NQC)	0.435*	0.312*	0.343*	0.265*
RLS(NQC)	0.429*	0.311*	0.337*	0.267*
QPP-PRP	0.350*	0.270*	0.280*	0.210*
NQA-QPP	0.175	0.115	0.082	0.075
BERTQPP	0.243*	0.170*	0.236*	0.185*
qppBERT-PL	0.203*	0.169*	0.181*	0.165*
M-QPPF	0.242*	0.174*	0.285*	0.219*
QPP-GenRE ($n = 200$)	**0.623**$^{\dagger *}$	**0.505**$^{\dagger *}$	0.484$^{\dagger *}$	0.395$^{\dagger *}$
QPP-GenRE ($n = 10$)	0.617*	0.504*	0.471*	0.388*
QPP-GenRE ($n = 100$)	**0.623***	**0.505***	0.485*	0.396*
QPP-GenRE ($n = 1,000$)	**0.623***	**0.505***	**0.487***	**0.398***

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 [176].

5.7.5 QPP-GenRE's interpretability

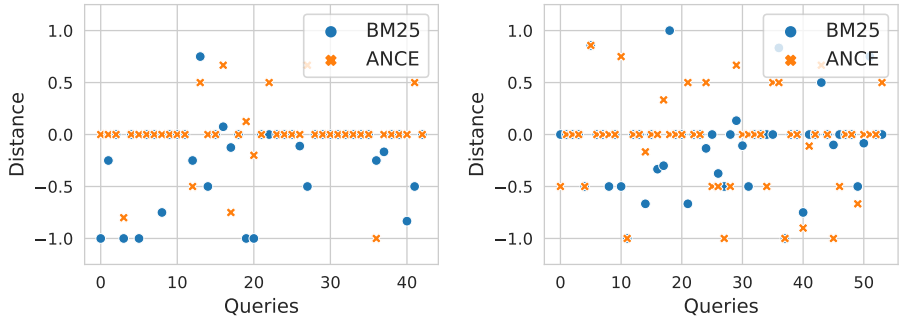
As QPP-GenRE computes QPP based on generated relevance judgments, we analyze QPP errors from the perspective of relevance judgment generation. Figure 5.7 shows the QPP errors of QPP-GenRE integrated with LLaMA-7B in predicting the performance

Table 5.12: Continued from Table 5.11. Correlation coefficients (Pearson's ρ and Kendall's τ) between actual retrieval quality, in terms of nDCG@3, of ConvDR [282] and its performance predicted by QPP-GenRE/baselines, on CAsT-19 and 20, using human-written query rewrites.

	Ranker: ConvDR (Human-written)			
	CAsT-19		CAsT-20	
QPP method	P-ρ	K-τ	P-ρ	K-τ
Clarity	0.257*	0.176*	0.126	0.088
WIG	0.412*	0.285*	0.384*	0.264*
NQC	0.431*	0.307*	0.339*	0.261*
σ_{max}	0.378*	0.267*	0.282*	0.219*
n($\sigma_{x\%}$)	0.216*	0.196*	0.201*	0.156*
SMV	0.386*	0.285*	0.275*	0.216*
UEF(NQC)	0.427*	0.310*	0.341*	0.263*
RLS(NQC)	0.413*	0.308*	0.342*	0.259*
QPP-PRP	0.345*	0.265*	0.275*	0.205*
NQA-QPP	0.142	0.091	0.065	0.058
BERTQPP	0.256*	0.172*	0.262*	0.209*
qppBERT-PL	0.105	0.090	0.166*	0.161*
M-QPPF	0.262*	0.190*	0.313*	0.254*
QPP-GenRE ($n = 200$)	**0.645**†*	**0.529**†*	0.678†*	0.551†*
QPP-GenRE ($n = 10$)	0.619*	0.506*	0.659	0.534
QPP-GenRE ($n = 100$)	0.644*	**0.529***	0.675*	0.547*
QPP-GenRE ($n = 1,000$)	**0.645***	**0.529***	**0.684***	**0.556***

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 5.13 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 5.14 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 QPP baselines, we provide a case study shown in Tables 5.15 and 5.16. Table 5.15 lists the predicted or ground-truth retrieval quality in terms of RR@10 of BM25 for the query



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

Figure 5.7: 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 5.13: 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 5.14: 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

"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 5.16 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 relevant. Specifically, QPP-GenRE does not identify the key part "Captain Robert Gray" in this item. This

Table 5.15: 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 5.16: 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

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

Table 5.17: Inference efficiency of supervised QPP baselines and QPP-GenRE integrated with LLaMA-7B on TREC-DL 19 to predict 1–4 different IR metrics. n denotes QPP-GenRE's judgment depth in a ranked list. Cases with higher latency than QPP-GenRE ($n = 10$) are underlined.

	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	1157.08
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

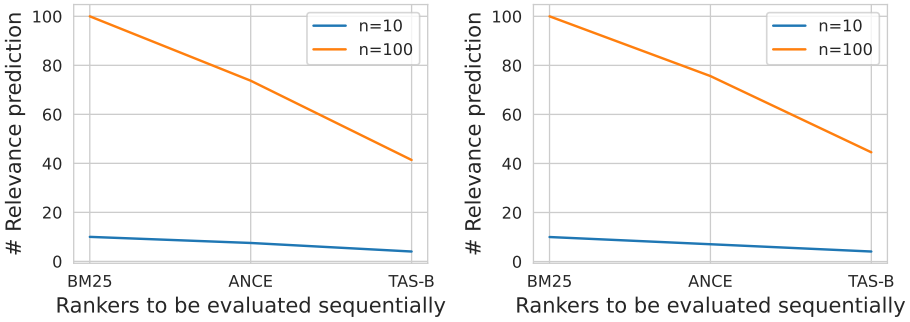
predicts the number of relevant items in each chunk of the ranked list and then aggregates these numbers into an overall score. While it is possible to check the intermediate predictions of the number of relevant items per chunk, this information is too coarse to provide detailed insights. In contrast, QPP-GenRE predicts the relevance of each individual item, offering more granular and informative insights.

5.7.6 Computational cost analysis

Table 5.17 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.[12] 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 2 or more metrics than M-QPPF. Nevertheless, we acknowledge that QPP-GenRE 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 5.6.1 and 5.6.2).

We argue that QPP-GenRE's latency is still much smaller than some high-performing

[12] 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.



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

Figure 5.8: 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.

LLM-based re-rankers. E.g., Sun et al. [234] show that a GPT-4-based listwise re-ranker needs 10 API calls (one call takes 3,200ms) to re-rank 100 items for a query, resulting in 32,000ms 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 [84].

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 5.8 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).

5.8 Conclusions and Future Work

This chapter examines the following thesis-level research question:

RQ4 How can LLMs be used to effectively enhance QPP accuracy?

To answer this question, 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. QPP-GenRE can effectively harness the strong relevance prediction capabilities of LLMs, potentially leading to improved QPP accuracy. We have explored using open-source LLMs for generating relevance judgments. To further improve LLMs' performance in relevance prediction, we have examined training open-source LLMs with parameter-efficient fine-tuning (PEFT) on human-labeled relevance judgments, to improve the quality of relevance judgment generation and QPP. 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 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. Additionally, to mitigate the efficiency issue of calling LLMs, we devised a relevance judgment caching mechanism that improves efficiency by reusing previously predicted relevance judgments.

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 trade-off 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 [20]. 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 an purely offline setting [84], where latency is not necessarily an issue.

This chapter is the final research chapter of this thesis. In the next chapter, we summarize the key findings and discuss potential directions for future research.

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 [153]. 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 [153]. Exploring various LLMs to find the optimal one for relevance prediction is beyond the scope of this chapter. 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 [52–55]. 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 [99], (iii) compressing LLMs by using lower-bit (e.g., 2-bit) quantization [41] or using low-rank factorization [274], 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 trade-off between judgment cost and QPP performance.

6 Conclusions

In Section 1.1, we stated the overarching question that we addressed in this thesis:

> *"What key aspects of agentic workflows for information access should be optimized, and how can we effectively optimize these aspects?"*

To answer this question, this thesis has identified four key issues across three critical components of existing agentic workflows for information access. This final chapter has two sections. In Section 6.1, we revisit the research questions defined in Section 1.1, reflecting on how they have been addressed, summarize the key findings, and discuss their implications. In Section 6.2, we discuss future research directions to further advance agentic workflows for information access.

6.1 Main Findings

This section revisits the research questions posed in Section 1.1, reflecting on how they have been addressed, summarize the key findings, and discuss their implications. We structure our discussion into three parts, each focusing on the optimization of a key component in existing agentic workflows for information access: mixed-initiative strategy planning, ranking strategy planning, and ranking result reflection. Each corresponds to a distinct part of this thesis.

6.1.1 Mixed-initiative strategy planning

The first part of this thesis, which included Chapter 2, focused on optimizing the mixed-initiative strategy planning component. Chapter 2 addressed the issue of a narrow scope of system-initiative actions in predicting the timing of system initiative-taking. To resolve the issue, Chapter 2 broadened the scope of system-initiative actions by defining and modeling a new task, system initiative prediction (SIP). The SIP task aims to predict the timing of system initiative that covers a broad range of specific system initiative-taking actions; SIP functions as a high-level strategic decision and effective SIP has the potential to enhance downstream tasks. Chapter 2 explored the following question:

RQ1 How can we effectively model system initiative prediction (SIP), and how does this prediction benefit downstream tasks?

To address this question, we first conducted an empirical analysis that uncovered structural dependencies between system-initiative decisions and other factors in multi-turn conversations. Motivated by these insights, we proposed a multi-turn system-initiative predictor (MuSIc). We built MuSIc on conditional random fields (CRFs), probabilistic graphical models known for their effectiveness in capturing structural dependencies while providing great interpretability and transparency. We further explored how SIP can benefit two downstream tasks: clarification need prediction and action prediction. For the former, we proposed a SIP-to-clarification transfer learning method, which transferred knowledge gained from SIP to improve clarification need prediction performance. For action prediction, we introduced a SIP-aware hierarchical framework, where action prediction depended on SIP outcomes.

Experimental results demonstrated that MuSIc achieves state-of-the-art SIP performance, surpassing even large language model (LLM)-based baselines. Furthermore, a visual analysis revealed that MuSIc exhibits strong interpretability and transparency by learning transition matrices that explicitly illustrated the dependencies it captured. Moreover, we found that SIP significantly improves both downstream tasks.

Our findings suggest two broad implications. First, SIP and downstream tasks form a hierarchical decision-making process within agentic workflows [5], leading to superior performance compared to relying solely on downstream models. This highlights the effectiveness of structuring agentic workflows hierarchically, where complex tasks are decomposed into sequential steps, each building upon the previous one, ultimately enhancing overall system accuracy. Second, this chapter demonstrated that probabilistic graphical modeling remains highly effective and offers strong interpretability and transparency. This suggests that, even in the era of LLMs, probabilistic graphical models are valuable choices for agents to enhance interpretability and transparency in agentic workflows.

6.1.2 Ranking strategy planning

The second part of this thesis, which included Chapter 3, focused on optimizing the ranking strategy planning component. Chapter 3 addressed a research gap in dynamic per-query re-ranking depth prediction in the context of LLM-based re-ranking. Chapter 3 highlighted two key dimensions within the research gap. First, there was a lack of systematic empirical analysis examining the potential advantages of dynamic per-query re-ranking depths over fixed ones in the context of LLM-based re-ranking. Second, no prior research had explored predicting dynamic per-query re-ranking depth specifically for LLM-based re-ranking. Chapter 3 investigated the use of ranked list truncation (RLT) methods, which had previously been applied to non-LLM re-ranking [285], to model dynamic per-query re-ranking depth prediction in LLM re-ranking. We posed the following question:

RQ2 In the context of LLM-based re-ranking, what are the potential benefits of using dynamic per-query re-ranking depths over fixed ones, and to what extent can RLT methods effectively predict dynamic re-ranking depths?

To address this question, we began by conducting a systematic empirical analysis to identify the limitations of fixed re-ranking depths and explore the potential advantages

of dynamic re-ranking depths in the context of LLM-based re-ranking. Our findings revealed that effective dynamic per-query re-ranking depths can improve both efficiency and effectiveness. For example, in terms of efficiency, when using a highly effective retriever, re-ranking did not enhance ranking quality for 30% of queries. In such cases, assigning a re-ranking depth of zero allows the system to bypass expensive LLM-based re-ranking, leading to substantial computational savings. Regarding effectiveness, a deeper re-ranking depth degraded re-ranking quality for some queries, as the re-ranker mistakenly elevated irrelevant documents to the top of the ranked list. This suggests that dynamically adjusting the re-ranking depth to exclude "false positive" re-ranking candidate documents has the potential to enhance ranking quality.

Next, we carried out a comprehensive study to examine how effectively various RLT methods adapt to predicting dynamic per-query re-ranking depths in the context of LLM-based re-ranking. Our experimental results revealed that while RLT methods showed advantages in certain cases, they did not demonstrate a clear improvement over using a fixed re-ranking depth. We found additional key insights. For example, the choice of retriever had a substantial impact on determining optimal re-ranking depths in LLM-based re-ranking: with an effective retriever, a fixed re-ranking depth of 20 already provided an excellent effectiveness/efficiency trade-off; increasing the fixed depth beyond this point yielded diminishing returns, with no significant improvement in re-ranking effectiveness.

This chapter has one broader implication. Dynamic per-query re-ranking depth prediction intrinsically predicts *the number of LLM calls* on a per-query basis, striking a balance between effectiveness and efficiency. Beyond ranking strategy planning, this principle extends to other components of agentic workflows that involve LLM invocation, such as prompt chaining, where multiple LLM calls are made sequentially to solve a task [227]. By dynamically determining whether to invoke LLMs at all and, if so, how many calls are necessary, agentic workflows can minimize redundant computations, leading to greater scalability and resource efficiency.

6.1.3 Ranking result reflection

The third part of this thesis, which included Chapter 4 and Chapter 5, focused on optimizing the ranking result reflection component. Both chapters focused on query performance prediction (QPP). Chapter 4 focused on filling the research gap in QPP for conversational search, while Chapter 5 explored the underexplored area of LLM-enhanced QPP.

Chapter 4 identified two specific aspects of the research gap in QPP for conversational search. First, the performance of existing QPP methods, originally designed for ad-hoc search, remained unclear in conversational search, where queries are context-dependent, and the ranking quality of top results is emphasized. No prior research had evaluated these methods in this context. Second, conversational search brings new challenges that do not exist in ad-hoc search, e.g., understanding context-dependent queries in conversations. It had been unclear how to adapt existing QPP methods, originally developed for ad-hoc search, to conversational search scenarios. To address the gap, we asked the following question:

RQ3 How can QPP methods originally designed for ad-hoc search be effectively

adapted to conversational search, and how well do QPP methods for ad-hoc search perform in conversational search?

To answer this question, we adapted existing QPP methods that heavily rely on input queries to conversational search, by feeding self-contained query rewrites, generated by off-the-shelf query rewriting models, into these methods. With the adaptation strategy, we conducted a comprehensive study to investigate how well existing QPP methods in ad-hoc search perform in conversational search.

Our extensive experiments revealed four key findings. First, compared to using human-written query rewrites (i.e., ground-truth rewrites), we demonstrated that feeding machine-generated self-contained query rewrites is an effective way to adapt QPP methods that heavily rely on input queries for conversational search. However, we observed that, in general, human-written rewrites leads to to higher QPP quality than machine-generated ones. Second, we found that all QPP methods generally performed worse when predicting an information retrieval (IR) evaluation metric with a shallow cut-off compared to a metric with a deep cut-off. Third, we observed that retrieval score-based methods that model the retrieval scores of rankers exhibit promising results. One possible explanation is that these models bypassed query understanding issues in conversational search by not directly modeling input queries. Fourth, we showed that supervised QPP methods significantly outperformed their unsupervised counterparts, provided they had access to a large-scale training set.

This chapter has two broad implications. First, it effectively enabled existing QPP methods to reflect on ranking results in conversational scenarios. This advancement enhances the generalizability of agentic workflows, enabling them to function more effectively in conversational environments, where modern information access frequently takes place [12]. Second, this chapter used off-the-shelf query rewriting methods as a tool to adapt QPP methods for conversational scenarios without modifying the underlying QPP models. As an external tool, query rewriting provides a flexible and lightweight adaptation strategy. This suggests that integrating external tools with existing agents is an effective approach to adapting agentic workflows to new scenarios, rather than designing entirely new agents.

In Chapter 5, we focused on improving QPP accuracy by leveraging LLMs' capabilities. Despite their promising performance across various IR and natural language processing (NLP) tasks, it had been unclear how LLMs could effectively benefit QPP. We posed the following question:

RQ4 How can LLMs be used to effectively enhance QPP accuracy?

To answer this question, we proposed a framework for modeling QPP using automatically generated relevance judgments (QPP-GenRE), which decomposed QPP into independent subtasks of predicting the relevance of each document in a ranked list to the query, and then predicted different IR evaluation measures based on the relevance predictions. QPP-GenRE used the strong performance of LLMs in generating relevance judgments [249], with prior research showing that LLMs can achieve accuracy comparable to human labelers [242]. To further enhance the quality of generated relevance judgments, we fine-tuned open-source LLMs on human-labeled relevance data, experimenting with two families of models ranging from 1B to 70B parameters. Moreover,

since predicting a recall-based IR evaluation metric by definition requires knowing all relevant documents in the entire corpus, we proposed an approximation strategy for QPP-GenRE that predicts relevance for only a limited set of ranked documents and then uses their relevance judgments to estimate recall-based IR evaluation metrics. Additionally, to mitigate the efficiency issue of calling LLMs, we devised a relevance judgment caching mechanism that improves efficiency by reusing previously predicted relevance judgments.

Extensive experiments demonstrated that QPP-GenRE achieve state-of-the-art QPP accuracy in evaluating both lexical and neural retrievers across ad-hoc and conversational search scenarios. Moreover, we observed that fine-tuning significantly improves LLMs' performance in relevance judgment prediction compared to few-shot prompting, ultimately leading to more accurate QPP. Interestingly, our results showed that a fine-tuned 3B model provides the best trade-off between performance and computational efficiency: it not only outperforms a few-shot prompting-based 70B LLM, but also achieves performance comparable to that of fine-tuned 7B and 8B models. Additionally, we showed that the proposed caching mechanism markedly reduces the number of LLM calls for relevance prediction by approximately 30%. Besides improved QPP accuracy, QPP-GenRE demonstrated strong interpretability, enabling QPP errors to be analyzed in relation to errors in generated relevance judgments.

This chapter has one major broader implication. With high reflection accuracy, QPP-GenRE enhances the overall effectiveness of agentic workflows by ensuring that high-quality documents are either directly presented to users or forwarded to a response generator for further processing. Agentic workflows rely on accurate ranking result reflection to make decisions on whether to proceed with ranked documents or invoke alternative retrieval strategies. More accurate ranking reflection empowers agentic workflows to operate effectively in low-error-tolerance scenarios, where users rely on precise and reliable information to make informed decisions. For instance, in financial analysis, users may query historical stock trends to guide investment decisions, where inaccurate ranking could lead to misguided financial choices. Similarly, in systematic reviews [216], professionals depend on retrieving high-quality, evidence-based sources to answer complex scientific questions, where mis-ranked documents could undermine critical research conclusions.

6.2 Future Directions

We conclude this thesis by outlining promising research directions that have the potential to further advance agentic workflows for information access. Specifically, we examine future directions from two perspectives: potential enhancements to the three critical components explored in this thesis (Section 6.2.1) and broader research avenues that extend beyond these components (Section 6.2.2).

6.2.1 Future improvements of this thesis

We discuss future research directions in three components studied in this thesis: mixed-initiative strategy planning, ranking strategy planning, and ranking result reflection.

For mixed-initiative strategy planning, we identify two promising directions. First, our findings demonstrated that SIP significantly improved two downstream tasks: clarification need prediction and action prediction. While these improvements suggest SIP's potential to enhance the final response generation, we did not verify it with experiments. A valuable future direction is to assess the performance of a complete hierarchical agentic workflow that includes SIP, SIP-aware action prediction, and response generation. Such an investigation can offer deeper insights into how multi-level decision-making contributes to the ultimate response quality. Second, SIP does not consider personalization, which is crucial for accurately predicting the timing of system initiative. For example, the user query "apple price" may seem ambiguous, but if the system is aware that the user is an Apple enthusiast, it should infer that the query refers to Apple products rather than the fruit, avoiding unnecessary clarification. Similarly, if a user prefers to take the lead and dislikes interruptions, the system should adapt its mixed-initiative strategy planning to minimize the frequency of initiative-taking actions. Thus, an important future direction is incorporating users' personal data and preferences into mixed-initiative strategy planning planning.

Regarding ranking strategy planning, we summarize two future directions. First, our findings suggest that while predicting dynamic per-query re-ranking depths in LLM-based re-ranking is a highly meaningful task, it remains challenging, highlighting the need for developing more effective methods. One reflection on the limited advantages of RLT methods over using fixed re-ranking depths is that RLT primarily relies on retriever-side information, such as retrieval scores for candidate documents, without incorporating insights from the re-ranking stage. Thus, incorporating real-time re-ranking signals (e.g., re-ranking scores) into the prediction process holds promise for improving per-query re-ranking depth prediction. Second, this thesis focused on ranking strategy planning for document ranking but did not address the emerging paradigm of tool ranking [222, 275, 296]. While document ranking is effective for general information-seeking queries, many real-world user requests, such as retrieving real-time stock trends or weather updates, require specialized tools beyond document ranking. In practice, tool ranking is essential as the number of available tools continues to grow, with many offering similar functionalities [202]. Thus, a key future direction is to explore planning ranking strategies for tools, e.g., determining when to rank tools.

We identify two promising future research directions for ranking result reflection. First, Chapter 4 found that query rewriting quality plays a crucial role in achieving accurate QPP in conversational search, emphasizing the need for future research on enhancing query rewriting techniques. Second, this thesis focused on QPP itself, such as investigating QPP in conversational search and enhancing QPP with LLMs, but did not explore the application of QPP scores in determining execution paths within agentic workflows. While existing studies have demonstrated that effective QPP can optimize ranking quality by dynamically adjusting execution paths on a per-query basis, such as query routing [215], selective query expansion [13], and retriever selection [122], an important future direction is to investigate to what extent the QPP methods proposed in this thesis, including QPP adapted to conversational search QPP and LLM-enhanced QPP, can further enhance decision-making in agentic workflows for information access. Also, it is valuable to explore new applications of QPP in agentic workflows. One potential application is QPP-aware iterative retrieval, where QPP scores act as feedback

signals to continuously refine retrieval performance.

6.2.2 Broader research directions

In this section, we identify three broader research avenues. First, while this thesis explored adjusting execution paths of agentic workflows on a per-query basis, the overall structure of these workflows remained manually pre-defined. For example, we explicitly designed an agentic workflow where SIP precedes action prediction, and another where retrieval is followed by re-ranking depth prediction and re-ranking. The reliance on pre-defined structures requires significant human effort, limiting the scalability and adaptability of agentic workflows to new, complex domains [290]. Thus, exploring the automation of agentic workflows in information access is a promising direction to minimize reliance on human intervention, enhancing adaptability and scalability [197, 288, 290].

Second, this thesis examined an essential dimension of ranking result reflection, i.e., to automatically assess the utility of ranking results. However, as agentic workflows gain greater autonomy, they also bear increased responsibility [5]. This necessitates expanding the scope of reflection beyond mere utility, to integrate human-valued principles into reflection, whether in ranking result reflection or reflection on response generation.

Third, this thesis focused on text-based agentic workflows, yet real-world information access increasingly involves multi-modal data, including images, videos, and audio. Future research could investigate multi-modal agentic workflows that process different modalities. This could involve aligning multi-modal retrieval techniques with agentic decision-making to create more comprehensive and context-aware information access systems.

Bibliography

[1] Z. Abbasiantaeb, C. Meng, D. Rau, A. Krasakis, H. A. Rahmani, and M. Aliannejadi. LLM-based retrieval and generation pipelines for TREC interactive knowledge assistance track (iKAT) 2023. In *Proceedings of the Thirty-Second Text REtrieval Conference (TREC 2023)*, 2023. (Cited on page 12.)

[2] Z. Abbasiantaeb, C. Meng, L. Azzopardi, and M. Aliannejadi. Can we use large language models to fill relevance judgment holes? In *Joint Proceedings of the 1st Workshop on Evaluation Methodologies, Testbeds and Community for Information Access Research (EMTCIR 2024) and the 1st Workshop on User Modelling in Conversational Information Retrieval (UM-CIR 2024)*, 2024. (Cited on page 98.)

[3] Z. Abbasiantaeb, C. Meng, L. Azzopardi, and M. Aliannejadi. Improving the reusability of conversational search test collections. In *European Conference on Information Retrieval*, pages 196–213, 2025. (Cited on pages 12 and 91.)

[4] A. Abdallah, J. Mozafari, B. Piryani, M. Ali, and A. Jatowt. Rankify: A comprehensive Python toolkit for retrieval, re-ranking, and retrieval-augmented generation. *arXiv preprint arXiv:2502.02464*, 2025. (Cited on page 1.)

[5] D. B. Acharya, K. Kuppan, and B. Divya. Agentic AI: Autonomous intelligence for complex goals–a comprehensive survey. *IEEE Access*, 2025. (Cited on pages 130 and 135.)

[6] H. Al-Thani, T. Elsayed, and B. J. Jansen. Improving conversational search with query reformulation using selective contextual history. *Data and Information Management*, page 100025, 2022. (Cited on page 87.)

[7] M. Aliannejadi, H. Zamani, F. Crestani, and W. B. Croft. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 475–484, 2019. (Cited on pages 2, 21, and 70.)

[8] M. Aliannejadi, M. Chakraborty, E. A. Ríssola, and F. Crestani. Harnessing evolution of multi-turn conversations for effective answer retrieval. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, pages 33–42, 2020. (Cited on page 70.)

[9] M. Aliannejadi, J. Kiseleva, A. Chuklin, J. Dalton, and M. Burtsev. ConvAI3: Generating clarifying questions for open-domain dialogue systems (ClariQ). *arXiv preprint arXiv:2009.11352*, 2020. (Cited on pages 2, 17, 20, 21, 30, and 34.)

[10] M. Aliannejadi, L. Azzopardi, H. Zamani, E. Kanoulas, P. Thomas, and N. Craswell. Analysing mixed initiatives and search strategies during conversational search. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 16–26, 2021. (Cited on pages 3 and 17.)

[11] M. Aliannejadi, J. Kiseleva, A. Chuklin, J. Dalton, and M. Burtsev. Building and evaluating open-domain dialogue corpora with clarifying questions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4473–4484, 2021. (Cited on pages 2, 5, 17, 20, 21, 30, and 34.)

[12] J. Allan, E. Choi, D. P. Lopresti, and H. Zamani. Future of information retrieval research in the age of generative AI. *arXiv preprint arXiv:2412.02043*, 2024. (Cited on pages 2 and 132.)

[13] G. Amati, C. Carpineto, and G. Romano. Query difficulty, robustness, and selective application of query expansion. In *European Conference on Information Retrieval*, pages 127–137, 2004. (Cited on pages 3, 91, and 134.)

[14] N. Arabzadeh, A. Bigdeli, M. Zihayat, and E. Bagheri. Query performance prediction through retrieval coherency. In *European Conference on Information Retrieval*, pages 193–200, 2021. (Cited on page 96.)

[15] N. Arabzadeh, M. Khodabakhsh, and E. Bagheri. BERT-QPP: Contextualized pre-trained transformers for query performance prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2857–2861, 2021. (Cited on pages 3, 6, 70, 71, 75, 77, 86, 92, 96, 97, 104, 105, and 106.)

[16] N. Arabzadeh, M. Seifikar, and C. L. Clarke. Unsupervised question clarity prediction through retrieved item coherency. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3811–3816, 2022. (Cited on pages 2, 17, 21, 34, 35, 70, and 87.)

[17] N. Arabzadeh, R. Hamidi Rad, M. Khodabakhsh, and E. Bagheri. Noisy perturbations for estimating query difficulty in dense retrievers. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3722–3727, 2023. (Cited on pages 92, 104, 105, and 106.)

[18] N. Arabzadeh, C. Meng, M. Aliannejadi, and E. Bagheri. Query performance prediction: From

fundamentals to advanced techniques. In *European Conference on Information Retrieval*, pages 381–388, 2024. (Cited on pages 12, 63, and 91.)

[19] N. Arabzadeh, C. Meng, M. Aliannejadi, and E. Bagheri. Query performance prediction: Techniques and applications in modern information retrieval. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 291–294, 2024.

[20] N. Arabzadeh, C. Meng, M. Aliannejadi, and E. Bagheri. Query performance prediction: Theory, techniques and applications. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 991–994, 2025. (Cited on pages 12, 63, 65, 91, and 127.)

[21] A. Arampatzis, J. Kamps, and S. Robertson. Where to stop reading a ranked list? threshold optimization using truncated score distributions. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 524–531, 2009. (Cited on pages 43, 44, and 62.)

[22] N. Asadi and J. Lin. Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 997–1000, 2013. (Cited on page 63.)

[23] A. Askari, M. Aliannejadi, C. Meng, E. Kanoulas, and S. Verberne. Expand, highlight, generate: RL-driven document generation for passage reranking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10087–10099, 2023. (Cited on pages 12 and 99.)

[24] A. Askari, C. Meng, M. Aliannejadi, Z. Ren, E. Kanoulas, and S. Verberne. Generative retrieval with few-shot indexing. *arXiv preprint arXiv:2408.02152*, 2024.

[25] A. Askari, R. Petcu, C. Meng, M. Aliannejadi, A. Abolghasemi, E. Kanoulas, and S. Verberne. SOLID: Self-seeding and multi-intent self-instructing LLMs for generating intent-aware information-seeking dialogs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6390–6410, 2025. (Cited on page 12.)

[26] J. A. Aslam and V. Pavlu. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In *European Conference on Information Retrieval*, pages 198–209, 2007. (Cited on page 96.)

[27] S. Avula, B. Choi, and J. Arguello. Why and when: Understanding system initiative during conversational collaborative search. *arXiv preprint arXiv:2303.13484*, 2023. (Cited on pages 2, 17, and 21.)

[28] L. Azzopardi, M. Dubiel, M. Halvey, and J. Dalton. Conceptualizing agent-human interactions during the conversational search process. In *The Second International Workshop on Conversational Approaches to Information Retrieval*, 2018. (Cited on page 21.)

[29] L. Azzopardi, M. Aliannejadi, and E. Kanoulas. Towards building economic models of conversational search. In *European Conference on Information Retrieval*, pages 31–38, 2022. (Cited on pages 3 and 17.)

[30] D. Bahri, Y. Tay, C. Zheng, D. Metzler, and A. Tomkins. Choppy: Cut transformer for ranked list truncation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1513–1516, 2020. (Cited on pages 43, 44, 45, 48, 49, 50, 52, and 62.)

[31] D. Bahri, C. Zheng, Y. Tay, D. Metzler, and A. Tomkins. Surprise: Result list truncation via extreme value theory. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2404–2408, 2023. (Cited on pages 43, 48, 49, 50, 53, and 62.)

[32] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2018. (Cited on page 86.)

[33] R. Bommasani, P. Liang, and T. Lee. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525(1):140–146, 2023. (Cited on pages 63, 64, and 99.)

[34] A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, and J. Zien. Efficient query evaluation using a two-level retrieval process. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 426–434, 2003. (Cited on page 63.)

[35] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark,

C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020. (Cited on pages 29 and 97.)

[36] S. Bruch, C. Lucchese, and F. M. Nardini. Efficient and effective tree-based and neural learning to rank. *Foundations and Trends® in Information Retrieval*, 17(1):1–123, 2023. (Cited on pages 2, 44, 47, and 63.)

[37] F. Busolin, C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, and S. Trani. Learning early exit strategies for additive ranking ensembles. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2217–2221, 2021. (Cited on page 63.)

[38] B. B. Cambazoglu, H. Zaragoza, O. Chapelle, J. Chen, C. Liao, Z. Zheng, and J. Degenhardt. Early exit optimizations for additive machine learned ranking systems. In *Proceedings of the Third ACM International Conference on Web search and Data Mining*, pages 411–420, 2010. (Cited on page 63.)

[39] D. Carmel and E. Yom-Tov. *Estimating the Query Difficulty for Information Retrieval*. Morgan & Claypool Publishers, 2010. (Cited on pages 3, 74, 86, 91, 96, 100, and 106.)

[40] D. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018. (Cited on page 25.)

[41] J. Chee, Y. Cai, V. Kuleshov, and C. M. De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems*, volume 36, pages 4396–4429, 2023. (Cited on page 128.)

[42] C.-Y. Chen, Y.-S. Lin, and C.-C. Lee. Emotion-shift aware CRF for decoding emotion sequence in conversation. In *Interspeech*, pages 1148–1152, 2022. (Cited on page 22.)

[43] M. Chen, X. Yu, W. Shi, U. Awasthi, and Z. Yu. Controllable mixed-initiative dialogue generation through prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 951–966, 2023. (Cited on page 21.)

[44] N. Chen, J. Liu, X. Dong, Q. Liu, T. Sakai, and X.-M. Wu. AI can be cognitively biased: An exploratory study on threshold priming in LLM-based batch relevance assessment. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 54–63, 2024. (Cited on page 92.)

[45] X. Chen, B. He, and L. Sun. Groupwise query performance prediction with BERT. In *European Conference on Information Retrieval*, pages 64–74, 2022. (Cited on pages 6, 70, 71, 75, 76, 77, 92, 96, 97, and 105.)

[46] Z. Chen, R. Yang, Z. Zhao, D. Cai, and X. He. Dialogue act recognition via CRF-attentive structured network. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 225–234, 2018. (Cited on page 22.)

[47] Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao, and X. He. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv preprint arXiv:2401.03428*, 2024. (Cited on page 1.)

[48] E. Choi, H. He, M. Iyyer, M. Yatskar, W.-t. Yih, Y. Choi, P. Liang, and L. Zettlemoyer. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, 2018. (Cited on page 76.)

[49] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. (Cited on page 18.)

[50] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. (Cited on pages 93, 97, and 98.)

[51] D. Cohen, B. Mitra, O. Lesota, N. Rekabsaz, and C. Eickhoff. Not all relevance scores are equal:

Efficient uncertainty and calibration modeling for deep retrieval models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 654–664, 2021. (Cited on pages 43 and 62.)

[52] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees. Overview of the TREC 2019 deep learning track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference (TREC 2019)*, 2019. (Cited on pages 45, 52, 93, 102, 103, 106, and 128.)

[53] N. Craswell, B. Mitra, E. Yilmaz, and D. Campos. Overview of the TREC 2020 deep learning track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference (TREC 2020)*, 2020. (Cited on pages 45 and 52.)

[54] N. Craswell, B. Mitra, E. Yilmaz, D. F. Campos, and J. Lin. Overview of the TREC 2021 deep learning track. In *Proceedings of the Thirtieth Text REtrieval Conference (TREC 2021)*, 2021. (Cited on page 117.)

[55] N. Craswell, B. Mitra, E. Yilmaz, D. F. Campos, J. Lin, E. M. Voorhees, and I. Soboroff. Overview of the TREC 2022 deep learning track. In *Proceedings of the Thirty-First Text REtrieval Conference (TREC 2022)*, 2022. (Cited on pages 93, 102, 103, 106, and 128.)

[56] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299–306, 2002. (Cited on pages 3, 69, 74, 96, 104, and 106.)

[57] Y. Cui, Z. Yang, and X. Yao. Efficient and effective text encoding for Chinese LLaMA and Alpaca. *arXiv preprint arXiv:2304.08177*, 2023. (Cited on page 30.)

[58] J. S. Culpepper, C. L. Clarke, and J. Lin. Dynamic cutoff prediction in multi-stage retrieval systems. In *Proceedings of the 21st Australasian Document Computing Symposium*, pages 17–24, 2016. (Cited on pages 2, 6, 43, 44, and 63.)

[59] J. S. Culpepper, F. Diaz, and M. D. Smucker. Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in Lorne (SWIRL 2018). *SIGIR Forum*, 52(1): 34–90, 2018. (Cited on page 21.)

[60] R. Cummins, J. Jose, and C. O'Riordan. Improved query performance prediction using standard deviation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1089–1090, 2011. (Cited on pages 75, 77, 96, and 104.)

[61] J. Dalton, C. Xiong, and J. Callan. CAsT 2020: The conversational assistance track overview. In *Proceedings of the Twenty-Ninth Text REtrieval Conference (TREC 2020)*, 2020. (Cited on pages 6, 70, 71, 72, 73, 76, 77, 86, 94, and 120.)

[62] J. Dalton, C. Xiong, V. Kumar, and J. Callan. CAsT-19: A dataset for conversational information seeking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1985–1988, 2020. (Cited on pages 6, 70, 71, 73, 76, 77, 86, 94, and 120.)

[63] J. Dalton, C. Xiong, and J. Callan. TREC CAsT 2021: The conversational assistance track overview. In *Proceedings of the Thirtieth Text REtrieval Conference (TREC 2021)*, 2021. (Cited on page 70.)

[64] D. A. Darling. The Kolmogorov-Smirnov, Cramér-von Mises tests. *The Annals of Mathematical Statistics*, 28(4):823–838, 1957. (Cited on page 53.)

[65] S. Datta, D. Ganguly, D. Greene, and M. Mitra. Deep-QPP: A pairwise interaction-based deep learning model for supervised query performance prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 201–209, 2022. (Cited on pages 6, 69, 70, 71, 75, 77, 92, 96, and 97.)

[66] S. Datta, D. Ganguly, M. Mitra, and D. Greene. A relative information gain-based query performance prediction framework with generated query variants. *ACM Transactions on Information Systems (TOIS)*, 41(2):1–31, 2022. (Cited on pages 71, 77, 80, and 96.)

[67] S. Datta, S. MacAvaney, D. Ganguly, and D. Greene. A 'Pointwise-Query, Listwise-Document' based query performance prediction approach. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2148–2153, 2022. (Cited on pages 6, 69, 70, 71, 75, 77, 86, 92, 96, 97, 104, 105, and 106.)

[68] Y. Deng, W. Zhang, W. Lam, H. Cheng, and H. Meng. User satisfaction estimation with sequential dialogue act modeling in goal-oriented conversational systems. In *Proceedings of the ACM Web Conference 2022*, pages 2998–3008, 2022. (Cited on page 22.)

[69] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115, 2023. (Cited on pages 93, 97, 98, 101, and 108.)

[70] R. Deveaud, J. Mothe, and J.-Y. Nie. Learning to rank system configurations. In *Proceedings of the*

25th ACM International on Conference on Information and Knowledge Management, page 2001–2004, 2016. (Cited on pages 3 and 91.)

[71] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. (Cited on pages 25, 63, 96, and 97.)

[72] G. M. Di Nunzio and G. Faggioli. A study of a gain based approach for query aspects in recall oriented tasks. *Applied Sciences*, 11(19), 2021. (Cited on page 91.)

[73] F. Diaz. Performance prediction using spatial autocorrelation. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 583–590, 2007. (Cited on page 96.)

[74] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, T. Liu, B. Chang, X. Sun, L. Li, and Z. Sui. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, 2024. (Cited on pages 29 and 97.)

[75] A. Drozdov, H. Zhuang, Z. Dai, Z. Qin, R. Rahimi, X. Wang, D. Alon, M. Iyyer, A. McCallum, D. Metzler, and K. Hui. PaRaDe: Passage ranking using demonstrations with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14242–14252, 2023. (Cited on pages 50, 63, 64, and 99.)

[76] A. Elgohary, D. Peskov, and J. Boyd-Graber. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924, 2019. (Cited on page 76.)

[77] G. Faggioli, M. Ferrante, N. Ferro, R. Perego, and N. Tonellotto. Hierarchical dependence-aware evaluation measures for conversational search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1935–1939, 2021. (Cited on pages 70 and 91.)

[78] G. Faggioli, O. Zendel, J. S. Culpepper, N. Ferro, and F. Scholer. An enhanced evaluation framework for query performance prediction. In *European Conference on Information Retrieval*, pages 115–129, 2021. (Cited on pages 89 and 91.)

[79] G. Faggioli, M. Ferrante, N. Ferro, R. Perego, and N. Tonellotto. A dependency-aware utterances permutation strategy to improve conversational evaluation. In *European Conference on Information Retrieval*, pages 184–198, 2022. (Cited on page 70.)

[80] G. Faggioli, O. Zendel, J. S. Culpepper, N. Ferro, and F. Scholer. sMARE: A new paradigm to evaluate and understand query performance prediction methods. *Information Retrieval Journal*, 25(2):94–122, 2022. (Cited on page 89.)

[81] G. Faggioli, L. Dietz, C. L. A. Clarke, G. Demartini, M. Hagen, C. Hauff, N. Kando, E. Kanoulas, M. Potthast, B. Stein, and H. Wachsmuth. Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 39–50, 2023. (Cited on pages 92, 94, 98, 101, 116, and 117.)

[82] G. Faggioli, N. Ferro, J. Mothe, F. Raiber, and M. Fröbe. Report on the 1st workshop on query performance prediction and its evaluation in new tasks (QPP++ 2023) at ECIR 2023. *SIGIR Forum*, 57 (1):1–7, 2023. (Cited on page 91.)

[83] G. Faggioli, N. Ferro, C. Muntean, R. Perego, and N. Tonellotto. A spatial approach to predict performance of conversational search systems. In *Proceedings of the 13th Italian Information Retrieval Workshop*, volume 3448, pages 41–46, 2023. (Cited on page 96.)

[84] G. Faggioli, N. Ferro, C. I. Muntean, R. Perego, and N. Tonellotto. A geometric framework for query performance prediction in conversational search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1355–1365, 2023. (Cited on pages 91, 94, 96, 126, and 128.)

[85] G. Faggioli, T. Formal, S. Lupart, S. Marchesin, S. Clinchant, N. Ferro, and B. Piwowarski. Towards query performance prediction for neural information retrieval: Challenges and opportunities. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 51–63, 2023. (Cited on pages 96 and 106.)

[86] G. Faggioli, T. Formal, S. Marchesin, S. Clinchant, N. Ferro, and B. Piwowarski. Query performance prediction for neural IR: Are we there yet? In *European Conference on Information Retrieval*, pages 232–248, 2023. (Cited on pages 95 and 96.)

[87] T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant. From distillation to hard negative sampling: Making sparse neural IR models more effective. In *Proceedings of the 45th International ACM SIGIR*

Conference on Research and Development in Information Retrieval, pages 2353–2359, 2022. (Cited on pages 45, 52, 56, and 58.)

[88] M. Fröbe, L. Gienapp, M. Potthast, and M. Hagen. Bootstrapped nDCG estimation in the presence of unjudged documents. In *European Conference on Information Retrieval*, pages 313–329, 2023. (Cited on page 102.)

[89] Y. Fu, C. Tan, B. Bi, M. Chen, Y. Feng, and A. Rush. Latent template induction with gumbel-CRFs. In *Advances in Neural Information Processing Systems*, volume 33, pages 20259–20271, 2020. (Cited on page 18.)

[90] D. Ganguly and E. Yilmaz. Query-specific variable depth pooling via query performance prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2303–2307, 2023. (Cited on pages 63 and 91.)

[91] D. Ganguly, S. Datta, M. Mitra, and D. Greene. An analysis of variations in the effectiveness of query performance prediction. In *European Conference on Information Retrieval*, pages 215–229, 2022. (Cited on pages 69, 77, 91, and 92.)

[92] J. Gao, C. Xiong, P. Bennett, and N. Craswell. *Neural Approaches to Conversational Information Retrieval*, volume 44. Springer, 2023. (Cited on page 21.)

[93] L. Gao, Z. Dai, and J. Callan. Understanding BERT rankers under distillation. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, pages 149–152, 2020. (Cited on page 63.)

[94] A. Gema, P. Minervini, L. Daines, T. Hope, and B. Alex. Parameter-efficient fine-tuning of LLaMA for the clinical domain. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 91–104, 2024. (Cited on page 98.)

[95] S. Ghosh, S. Ghosh, and C. Shah. Toward connecting speech acts and search actions in conversational search tasks. In *2023 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 119–131, 2023. (Cited on page 21.)

[96] L. Gienapp, M. Fröbe, M. Hagen, and M. Potthast. Sparse pairwise re-ranking with pre-trained transformers. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 72–80, 2022. (Cited on page 63.)

[97] F. Gilardi, M. Alizadeh, and M. Kubli. ChatGPT outperforms crowd-workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. (Cited on page 92.)

[98] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30, 2004. (Cited on page 37.)

[99] Y. Gu, L. Dong, F. Wei, and M. Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on page 128.)

[100] A. Gupta, P. Goyal, S. Sarkar, and M. Gattu. Fully contextualized biomedical NER. In *European Conference on Information Retrieval*, pages 117–124, 2019. (Cited on pages 22 and 31.)

[101] S. Gupta, M. Kutlu, V. Khetan, and M. Lease. Correlation, prediction and ranking of evaluation metrics in information retrieval. In *European Conference on Information Retrieval*, pages 636–651, 2019. (Cited on page 92.)

[102] H. Hashemi, H. Zamani, and W. B. Croft. Performance prediction for non-factoid question answering. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 55–58, 2019. (Cited on pages 3, 6, 70, 71, 75, 77, 80, 86, 92, 96, 97, 105, and 106.)

[103] C. Hauff, D. Hiemstra, and F. de Jong. A survey of pre-retrieval query performance predictors. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 1419–1420, 2008. (Cited on pages 86 and 106.)

[104] C. Hauff, L. Azzopardi, D. Hiemstra, and F. de Jong. Query performance prediction: Evaluation contrasted with effectiveness. In *European Conference on Information Retrieval*, pages 204–216, 2010.

[105] B. He and I. Ounis. Query performance prediction. *Information Systems*, 31(7):585–594, 2006. (Cited on pages 69, 70, and 86.)

[106] J. He, M. Larson, and M. de Rijke. Using coherence-based measures to predict query difficulty. In *European Conference on Information Retrieval*, pages 689–694, 2008. (Cited on pages 70 and 86.)

[107] W. He, Y. Dai, Y. Zheng, Y. Wu, Z. Cao, D. Liu, P. Jiang, M. Yang, F. Huang, L. Si, J. Sun, and Y. Li. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10749–10757, 2022. (Cited on pages 22 and 36.)

[108] S. Hofstätter, M. Zlabinger, and A. Hanbury. Interpretable & time-budget-constrained contextualization for re-ranking. In *European Conference on Artificial Intelligence*, pages 1–8, 2020. (Cited on page 63.)

[109] S. Hofstätter, S.-C. Lin, J.-H. Yang, J. Lin, and A. Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122, 2021. (Cited on pages 93, 103, and 106.)

[110] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, and W. X. Zhao. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, pages 364–381, 2024. (Cited on page 99.)

[111] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022. (Cited on pages 97 and 98.)

[112] Z. Huang, W. Xu, and K. Yu. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015. (Cited on pages 22 and 23.)

[113] A. N. Jagannatha and H. Yu. Structured prediction models for RNN based sequence labeling in clinical text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 856–865, 2016. (Cited on pages 22 and 31.)

[114] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. (Cited on pages 48 and 102.)

[115] D. Jeffrey, X. Chenyan, and C. Jamie. CAsT 2019: The conversational assistance track overview. In *Proceedings of the Twenty-Eighth Text REtrieval Conference (TREC 2019)*, 2019. (Cited on pages 6, 70, 71, 73, 76, 77, 86, 94, and 120.)

[116] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7029–7043, 2024. (Cited on page 2.)

[117] M. Jhunjhunwala, C. Bryant, and P. Shah. Multi-action dialog policy learning with interactive human teaching. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 290–296, 2020. (Cited on pages 22 and 36.)

[118] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. (Cited on page 98.)

[119] T. Jones, P. Thomas, F. Scholer, and M. Sanderson. Features of disagreement between retrieval effectiveness measures. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 847–850, 2015. (Cited on page 92.)

[120] O. Khattab and M. Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48, 2020. (Cited on page 80.)

[121] M. Khodabakhsh and E. Bagheri. Learning to rank and predict: Multi-task learning for ad hoc retrieval and query performance prediction. *Information Sciences*, 639:119015, 2023. (Cited on pages 92, 96, 97, 105, and 106.)

[122] E. Khramtsova, S. Zhuang, M. Baktashmotlagh, and G. Zuccon. Leveraging LLMs for unsupervised dense retriever ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1307–1317, 2024. (Cited on pages 3, 93, 98, and 134.)

[123] S. Kim and G. Kim. Saving dense retriever from shortcut dependency in conversational search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10278–10287, 2022. (Cited on pages 70 and 87.)

[124] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *The Third International Conference on Learning Representations*, 2015. (Cited on pages 31, 53, 77, and 107.)

[125] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009. (Cited on page 18.)

[126] H. Kumar, A. Agarwal, R. Dasgupta, and S. Joshi. Dialogue act sequence labeling using hierarchical encoder with CRF. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. (Cited on page 22.)

[127] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 111–119, 2001. (Cited on page 96.)

[128] J. D. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for

segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001. (Cited on pages 18 and 22.)

[129] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, 2016. (Cited on pages 22 and 23.)

[130] C. Lassance and S. Clinchant. The tale of two MSMARCO - and their unfair comparisons. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2431–2435, 2023. (Cited on page 119.)

[131] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127, 2001. (Cited on pages 74, 77, and 104.)

[132] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196, 2014. (Cited on page 53.)

[133] O. Lesota, N. Rekabsaz, D. Cohen, K. A. Grasserbauer, C. Eickhoff, and M. Schedl. A modern perspective on query likelihood with deep generative retrieval models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 185–195, 2021. (Cited on pages 43 and 62.)

[134] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020. (Cited on page 1.)

[135] M. Li, X. Zhang, J. Xin, H. Zhang, and J. Lin. Certified error control of candidate set pruning for two-stage relevance ranking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 333–345, 2022. (Cited on pages 2, 6, 43, 44, and 63.)

[136] X. Li, G. Dong, J. Jin, Y. Zhang, Y. Zhou, Y. Zhu, P. Zhang, and Z. Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025. (Cited on page 1.)

[137] Y. Li, X. Lin, W. Wang, F. Feng, L. Pang, W. Li, L. Nie, X. He, and T.-S. Chua. A survey of generative search and recommendation in the era of large language models. *arXiv preprint arXiv:2404.16924*, 2024. (Cited on page 1.)

[138] Z. Li, J. Kiseleva, and M. de Rijke. Rethinking supervised learning and reinforcement learning in task-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3537–3546, 2020. (Cited on pages 22 and 36.)

[139] Y.-C. Lien, D. Cohen, and W. B. Croft. An assumption-free approach to the dynamic truncation of ranked lists. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 79–82, 2019. (Cited on pages 2, 43, 44, 48, 50, 52, and 62.)

[140] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, and R. Nogueira. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362, 2021. (Cited on page 103.)

[141] J. Lin, R. Nogueira, and A. Yates. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Springer Nature, 2022. (Cited on page 1.)

[142] S.-C. Lin, J.-H. Yang, and J. Lin. Contextualized query embeddings for conversational search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015, 2021. (Cited on pages 70, 72, 87, and 88.)

[143] S.-C. Lin, J.-H. Yang, R. Nogueira, M.-F. Tsai, C.-J. Wang, and J. Lin. Multi-stage conversational passage retrieval: An approach to fusing term importance estimation and neural query rewriting. *ACM Transactions on Information Systems (TOIS)*, 39(4):1–29, 2021. (Cited on pages 70, 72, 87, and 88.)

[144] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 1950–1965, 2022. (Cited on pages 93 and 97.)

[145] T. Liu and B. K. H. Low. Goat: Fine-tuned LLaMA outperforms GPT-4 on arithmetic tasks. *arXiv preprint arXiv:2305.14201*, 2023. (Cited on pages 98 and 105.)

[146] L. Lu, C. Meng, F. Ravenda, M. Aliannejadi, and F. Crestani. Zero-shot and efficient clarification need prediction in conversational search. In *European Conference on Information Retrieval*, pages 389–404, 2025. (Cited on pages 12 and 91.)

[147] X. Lu, A. Moffat, and J. S. Culpepper. The effect of pooling and evaluation depth on IR metrics. *Information Retrieval Journal*, 19(4):416–445, 2016. (Cited on page 102.)

[148] Y. Lu, C. Li, H. Liu, J. Yang, J. Gao, and Y. Shen. An empirical study of scaling instruct-tuned large multimodal models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023. (Cited on page 98.)

[149] C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, and S. Trani. Query-level early exit for additive learning-to-rank ensembles. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2033–2036, 2020. (Cited on page 63.)

[150] M. Lupu and A. Hanbury. Patent retrieval. *Foundations and Trends® in Information Retrieval*, 7(1): 1–97, 2013. (Cited on pages 44 and 62.)

[151] S. Ma, C. Chen, Q. Chu, and J. Mao. Leveraging large language models for relevance judgments in legal case retrieval. *arXiv preprint arXiv:2403.18405*, 2024. (Cited on pages 92, 98, and 99.)

[152] X. Ma, X. Zhang, R. Pradeep, and J. Lin. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*, 2023. (Cited on pages 46, 50, 63, 64, and 99.)

[153] X. Ma, L. Wang, N. Yang, F. Wei, and J. Lin. Fine-tuning LLaMA for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425, 2024. (Cited on pages 3, 5, 43, 44, 45, 46, 47, 50, 52, 53, 56, 58, 63, 64, 65, 94, 99, 100, 119, 120, and 128.)

[154] Y. Ma, Y. Shao, Y. Wu, Y. Liu, R. Zhang, M. Zhang, and S. Ma. LeCaRD: A legal case retrieval dataset for chinese law system. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2342–2348, 2021. (Cited on page 98.)

[155] Y. Ma, Q. Ai, Y. Wu, Y. Shao, Y. Liu, M. Zhang, and S. Ma. Incorporating retrieval information into the truncation of ranking lists for better legal search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 438–448, 2022. (Cited on pages 43, 45, 48, 49, 50, 52, and 65.)

[156] S. MacAvaney and L. Soldaini. One-shot labeling for automatic relevance estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2230–2235, 2023. (Cited on pages 92, 93, 98, and 99.)

[157] S. MacAvaney, F. M. Nardini, R. Perego, N. Tonellotto, N. Goharian, and O. Frieder. Efficient document re-ranking for transformers by precomputing term representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58, 2020. (Cited on page 63.)

[158] S. MacAvaney, N. Tonellotto, and C. Macdonald. Adaptive re-ranking with a corpus graph. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1491–1500, 2022. (Cited on page 53.)

[159] S. MacAvaney, A. Roegiest, A. Lipani, A. Parry, B. Engelmann, C. K. Kreutz, C. Meng, E. Frayling, E. Yang, F. Schlatt, G. Faggioli, H. Scells, I. Atanassova, J. Friese, J. Bevendorff, J. Sanz-Cruzado, J. Trippas, K. Pathak, K. Dhole, L. Azzopardi, M. Fröbe, M. Bertin, N. Prasad, S. Zerhoudi, S. Wang, S. Chatterjee, T. Jänich, U. Kruschwitz, X. Wang, and Z. Long. Report on the collab-a-thon at ECIR 2024. *SIGIR Forum*, 58(1):1–11, 2024. (Cited on page 12.)

[160] C. Macdonald, R. L. Santos, and I. Ounis. On the usefulness of query features for learning to rank. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2559–2562, 2012. (Cited on page 91.)

[161] M. Makary, M. Oakes, and F. Yamout. Towards automatic generation of relevance judgments for a test collection. In *2016 Eleventh International Conference on Digital Information Management (ICDIM)*, pages 121–126, 2016. (Cited on page 98.)

[162] M. Makary, M. Oakes, R. Mitkov, and F. Yammout. Using supervised machine learning to automatically build relevance judgments for a test collection. In *2017 28th International Workshop on Database and Expert Systems Applications (DEXA)*, pages 108–112, 2017. (Cited on page 98.)

[163] R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–275, 2001. (Cited on pages 43 and 62.)

[164] K. Mao, Z. Dou, and H. Qian. Curriculum contrastive context denoising for few-shot conversational dense retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 176–186, 2022. (Cited on pages 70, 72, 77, and 87.)

[165] K. Mao, Z. Dou, H. Qian, F. Mo, X. Cheng, and Z. Cao. ConvTrans: Transforming web search sessions for conversational dense retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2935–2946, 2022. (Cited on pages 70 and 87.)

[166] K. Mao, Z. Dou, F. Mo, J. Hou, H. Chen, and H. Qian. Large language models know your contextual search intent: A prompting framework for conversational search. In *Findings of the Association for*

Computational Linguistics: EMNLP 2023, pages 1211–1225, 2023. (Cited on page 29.)

[167] Y. Matsubara, T. Vu, and A. Moschitti. Reranking for efficient transformer-based answer selection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1577–1580, 2020. (Cited on page 63.)

[168] I. R. McKenzie, A. Lyzhov, M. M. Pieler, A. Parrish, A. Mueller, A. Prabhu, E. McLean, X. Shen, J. Cavanagh, A. G. Gritsevskiy, D. Kauffman, A. T. Kirtland, Z. Zhou, Y. Zhang, S. Huang, D. Wurgaft, M. Weiss, A. Ross, G. Recchia, A. Liu, J. Liu, T. Tseng, T. Korbak, N. Kim, S. R. Bowman, and E. Perez. Inverse scaling: When bigger isn't better. *Transactions on Machine Learning Research*, 2023. (Cited on page 31.)

[169] I. Mele, C. I. Muntean, F. M. Nardini, R. Perego, N. Tonellotto, and O. Frieder. Topic propagation in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2057–2060, 2020. (Cited on pages 70, 72, and 87.)

[170] C. Meng. Query performance prediction for conversational search and beyond. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3077–3077, 2024. (Cited on page 12.)

[171] C. Meng, P. Ren, Z. Chen, C. Monz, J. Ma, and M. de Rijke. RefNet: A reference-aware network for background based conversation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8496–8503, 2020. (Cited on pages 21 and 70.)

[172] C. Meng, P. Ren, Z. Chen, W. Sun, Z. Ren, Z. Tu, and M. de Rijke. DukeNet: A dual knowledge interaction network for knowledge-grounded conversation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1151–1160, 2020.

[173] C. Meng, P. Ren, Z. Chen, Z. Ren, T. Xi, and M. de Rijke. Initiative-aware self-supervised learning for knowledge-grounded conversations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 522–532, 2021. (Cited on pages 21 and 70.)

[174] C. Meng, M. Aliannejadi, and M. de Rijke. Performance prediction for conversational search using perplexities of query rewrites. In *Proceedings of the QPP++ 2023: Query Performance Prediction and Its Evaluation in New Tasks Workshop co-located with the 45th European Conference on Information Retrieval*, pages 25–28, 2023. (Cited on pages 12, 38, 39, 70, and 87.)

[175] C. Meng, M. Aliannejadi, and M. de Rijke. System initiative prediction for multi-turn conversational information seeking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1807–1817, 2023. (Cited on page 11.)

[176] C. Meng, N. Arabzadeh, M. Aliannejadi, and M. de Rijke. Query performance prediction: From ad-hoc to conversational search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2583–2593, 2023. (Cited on pages 38, 39, 120, and 121.)

[177] C. Meng, N. Arabzadeh, A. Askari, M. Aliannejadi, and M. de Rijke. Ranked list truncation for large language model-based re-ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 141–151, 2024. (Cited on pages 11, 95, and 99.)

[178] C. Meng, G. Faggioli, M. Aliannejadi, N. Ferro, and J. Mothe. QPP++ 2025: Query performance prediction and its applications in the era of large language models. In *European Conference on Information Retrieval*, pages 319–325, 2025. (Cited on pages 12 and 91.)

[179] C. Meng, F. Tonolini, F. Mo, N. Aletras, E. Yilmaz, and G. Kazai. Bridging the gap: From ad-hoc to proactive search in conversations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025. (Cited on pages 12 and 120.)

[180] C. Meng, N. Arabzadeh, A. Askari, M. Aliannejadi, and M. de Rijke. Query performance prediction using relevance judgments generated by large language models. *ACM Transactions on Information Systems (TOIS)*, to appear. (Cited on pages 3 and 11.)

[181] S. Mizzaro, J. Mothe, K. Roitero, and M. Z. Ullah. Query performance prediction and effectiveness evaluation without relevance judgments: Two sides of the same coin. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1233–1236, 2018. (Cited on page 97.)

[182] F. Mo, K. Mao, Z. Zhao, H. Qian, H. Chen, Y. Cheng, X. Li, Y. Zhu, Z. Dou, and J.-Y. Nie. A survey of conversational search. *arXiv preprint arXiv:2410.15576*, 2024. (Cited on pages 4, 6, 65, and 70.)

[183] F. Mo, C. Meng, M. Aliannejadi, and J.-Y. Nie. Conversational search: From fundamentals to frontiers in the LLM era. In *Proceedings of the 48th International ACM SIGIR Conference on Research and*

Development in Information Retrieval, 2025. (Cited on pages 12 and 120.)
[184] A. Moffat. Computing maximized effectiveness distance for recall-based metrics. *IEEE Transactions on Knowledge and Data Engineering*, 30(1):198–203, 2018. (Cited on page 102.)
[185] R. Nogueira and K. Cho. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019. (Cited on pages 44 and 62.)
[186] R. Nogueira, W. Yang, K. Cho, and J. Lin. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424*, 2019. (Cited on pages 53 and 63.)
[187] R. Nogueira, Z. Jiang, R. Pradeep, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, 2020. (Cited on pages 45, 52, and 58.)
[188] R. Nuray and F. Can. Automatic ranking of retrieval systems in imperfect environments. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 379–380, 2003. (Cited on page 98.)
[189] R. Nuray and F. Can. Automatic ranking of information retrieval systems using data fusion. *Information Processing & Management*, 42(3):595–614, 2006. (Cited on page 98.)
[190] P. Owoicho, J. Dalton, M. Aliannejadi, L. Azzopardi, J. R. Trippas, and S. Vakulenko. TREC CAsT 2022: Going beyond user ask and system retrieve with initiative and response generation. In *Proceedings of the Thirty-First Text REtrieval Conference (TREC 2022)*, 2022. (Cited on page 70.)
[191] J. Pérez-Iglesias and L. Araujo. Standard deviation as a query hardness estimator. In *International Symposium on String Processing and Information Retrieval*, pages 207–212, 2010. (Cited on pages 75, 96, and 104.)
[192] J. Pickands III. Statistical inference using extreme order statistics. *The Annals of Statistics*, 3(1): 119–131, 1975. (Cited on page 50.)
[193] E. Poesina, R. T. Ionescu, and J. Mothe. iQPP: A benchmark for image query performance prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2953–2963, 2023. (Cited on page 96.)
[194] R. Pradeep, R. Nogueira, and J. Lin. The Expando-Mono-Duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*, 2021. (Cited on pages 96 and 104.)
[195] R. Pradeep, S. Sharifymoghaddam, and J. Lin. RankVicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*, 2023. (Cited on pages 3, 5, 43, 44, 46, 50, 63, 64, 65, 93, and 99.)
[196] R. Pradeep, S. Sharifymoghaddam, and J. Lin. RankZephyr: Effective and robust zero-shot listwise reranking is a breeze! *arXiv preprint arXiv:2312.02724*, 2023. (Cited on pages 3, 5, 43, 44, 46, 48, 50, 52, 63, 64, 65, 93, and 99.)
[197] S. Qiao, R. Fang, Z. Qiu, X. Wang, N. Zhang, Y. Jiang, P. Xie, F. Huang, and H. Chen. Benchmarking agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*, 2025. (Cited on page 135.)
[198] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, L. Yan, J. Shen, T. Liu, J. Liu, D. Metzler, X. Wang, and M. Bendersky. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, 2024. (Cited on pages 3, 5, 43, 44, 46, 50, 63, 64, 65, 93, 97, and 99.)
[199] C. Qu, L. Yang, W. B. Croft, J. R. Trippas, Y. Zhang, and M. Qiu. Analyzing and characterizing user intent in information-seeking conversations. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 989–992, 2018. (Cited on pages 19 and 28.)
[200] C. Qu, L. Yang, W. B. Croft, Y. Zhang, J. R. Trippas, and M. Qiu. User intent prediction in information-seeking conversations. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, pages 25–33, 2019. (Cited on pages 18, 28, 29, and 37.)
[201] C. Qu, L. Yang, C. Chen, M. Qiu, W. B. Croft, and M. Iyyer. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 539–548, 2020. (Cited on pages 70, 71, 76, and 87.)
[202] C. Qu, S. Dai, X. Wei, H. Cai, S. Wang, D. Yin, J. Xu, and J.-R. Wen. Towards completeness-oriented tool retrieval for large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1930–1940, 2024. (Cited on page 134.)
[203] F. Radlinski and N. Craswell. A theoretical framework for conversational search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, pages 117–126, 2017. (Cited on pages 2, 21, and 86.)

[204] F. Radlinski, K. Balog, B. Byrne, and K. Krishnamoorthi. Coached conversational preference elicitation: A case study in understanding movie preferences. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 353–360, 2019. (Cited on pages 2 and 21.)

[205] V. Raheja and J. Tetreault. Dialogue act classification with context-aware self-attention. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3727–3733, 2019. (Cited on page 22.)

[206] S. D. Ravana, P. Rajagopal, and V. Balakrishnan. Ranking retrieval systems using pseudo relevance judgments. *Aslib Journal of Information Management*, 67(6):700–714, 2015. (Cited on page 98.)

[207] P. Ren, Z. Liu, X. Song, H. Tian, Z. Chen, Z. Ren, and M. de Rijke. Wizard of search engine: Access to information through conversations with search engines. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 533–543, 2021. (Cited on pages 18, 19, 21, 28, and 29.)

[208] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 109–123, 1995. (Cited on pages 52, 96, and 103.)

[209] H. Roitman. An enhanced approach to query performance prediction using reference lists. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 869–872, 2017. (Cited on pages 3 and 104.)

[210] H. Roitman, S. Erera, and G. Feigenblat. A study of query performance prediction for answer quality determination. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 43–46, 2019. (Cited on pages 70 and 87.)

[211] D. Sachan, M. Lewis, M. Joshi, A. Aghajanyan, W.-t. Yih, J. Pineau, and L. Zettlemoyer. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, 2022. (Cited on pages 50, 53, 63, 64, and 99.)

[212] A. Salemi and H. Zamani. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2395–2400, 2024. (Cited on pages 93, 98, and 99.)

[213] M. Samadi and D. Rafiei. Performance prediction for multi-hop questions. *arXiv preprint arXiv:2308.06431*, 2023. (Cited on page 96.)

[214] S. Sandmann, S. Riepenhausen, L. Plagwitz, and J. Varghese. Systematic analysis of ChatGPT, Google search and Llama 2 for clinical decision support tasks. *Nature Communications*, 15:2050, 2024. (Cited on page 1.)

[215] S. Sarnikar, Z. Zhang, and J. L. Zhao. Query-performance prediction for effective query routing in domain-specific repositories. *Journal of the Association for Information Science and Technology*, 65 (8):1597–1614, 2014. (Cited on pages 3, 86, and 134.)

[216] H. Scells, L. Azzopardi, G. Zuccon, and B. Koopman. Query variation performance prediction for systematic reviews. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1089–1092, 2018. (Cited on pages 91 and 133.)

[217] P. Schneider, A. Afzal, J. Vladika, D. Braun, and F. Matthes. Investigating conversational search behavior for domain exploration. In *European Conference on Information Retrieval*, page 608–616, 2023. (Cited on page 21.)

[218] I. Sekulić, M. Aliannejadi, and F. Crestani. Exploiting document-based features for clarification in conversational search. In *European Conference on Information Retrieval*, pages 413–427, 2022. (Cited on pages 2 and 21.)

[219] A. Sepliarskaia, J. Kiseleva, F. Radlinski, and M. de Rijke. Preference elicitation as an optimization problem. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 172–180, 2018. (Cited on pages 2 and 21.)

[220] C. Shah and R. W. White. Agents are not enough. *arXiv preprint arXiv:2412.16241*, 2024. (Cited on page 1.)

[221] G. Shang, A. Tixier, M. Vazirgiannis, and J.-P. Lorré. Speaker-change aware CRF for dialogue act classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 450–464, 2020. (Cited on page 22.)

[222] Z. Shi, Y. Wang, L. Yan, P. Ren, S. Wang, D. Yin, and Z. Ren. Retrieval models aren't tool-savvy: Benchmarking tool retrieval for large language models. *arXiv preprint arXiv:2503.01763*, 2025. (Cited on page 134.)

[223] A. Shtok, O. Kurland, and D. Carmel. Using statistical decision theory and relevance models for

query-performance prediction. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 259–266, 2010. (Cited on pages 3, 77, 86, and 104.)

[224] A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits. Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems (TOIS)*, 30(2):1–35, 2012. (Cited on pages 3, 74, 77, 80, 86, 92, 96, and 104.)

[225] L. Shu, H. Xu, B. Liu, and P. Molino. Modeling multi-action policy for task-oriented dialogues. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1304–1310, 2019. (Cited on pages 22 and 36.)

[226] A. Singh, D. Ganguly, S. Datta, and C. McDonald. Unsupervised query performance prediction for neural models utilising pairwise rank preferences. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2486–2490, 2023. (Cited on pages 92, 96, 104, and 110.)

[227] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei. Agentic retrieval-augmented generation: A survey on agentic RAG. *arXiv preprint arXiv:2501.09136*, 2025. (Cited on pages 1, 3, 69, and 131.)

[228] C. Siro, M. Aliannejadi, and M. de Rijke. Understanding user satisfaction with task-oriented dialogue systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2018–2023, 2022. (Cited on page 18.)

[229] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 66–73, 2001. (Cited on pages 97 and 98.)

[230] L. Soldaini and A. Moschitti. The cascade transformer: an application for efficient answer sentence selection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5697–5708, 2020. (Cited on page 63.)

[231] Y. Su, D. Cai, Y. Wang, D. Vandyke, S. Baker, P. Li, and N. Collier. Non-autoregressive text generation with pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 234–243, 2021. (Cited on pages 22 and 31.)

[232] J. Sun, C. Shaib, and B. C. Wallace. Evaluating the zero-shot robustness of instruction-tuned language models. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on page 97.)

[233] W. Sun, C. Meng, Q. Meng, Z. Ren, P. Ren, Z. Chen, and M. de Rijke. Conversations powered by cross-lingual knowledge. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1442–1451, 2021. (Cited on page 12.)

[234] W. Sun, L. Yan, X. Ma, S. Wang, P. Ren, Z. Chen, D. Yin, and Z. Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, 2023. (Cited on pages 3, 5, 43, 44, 46, 50, 63, 64, 94, 95, 99, 101, 108, and 126.)

[235] Z. Sun, Z. Li, H. Wang, D. He, Z. Lin, and Z. Deng. Fast structured decoding for sequence models. In *Advances in Neural Information Processing Systems*, volume 32, 2019. (Cited on pages 22 and 31.)

[236] C. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373, 2012. (Cited on pages 18 and 22.)

[237] R. Takehi, E. M. Voorhees, and T. Sakai. LLM-assisted relevance assessments: When should we ask LLMs for help? *arXiv preprint arXiv:2411.06877*, 2024. (Cited on page 92.)

[238] R. Tang, C. Zhang, X. Ma, J. Lin, and F. Türe. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2327–2340, 2024. (Cited on pages 63, 64, and 99.)

[239] Y. Tao and S. Wu. Query performance prediction by considering score magnitude and variance together. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1891–1894, 2014. (Cited on pages 3, 74, 75, 77, 92, 96, and 104.)

[240] Y. Tay, M. Dehghani, V. Q. Tran, X. Garcia, J. Wei, X. Wang, H. W. Chung, D. Bahri, T. Schuster, S. Zheng, D. Zhou, N. Houlsby, and D. Metzler. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*, 2023. (Cited on page 97.)

[241] P. Thomas, F. Scholer, P. Bailey, and A. Moffat. Tasks, queries, and rankers in pre-retrieval performance prediction. In *Proceedings of the 22nd Australasian Document Computing Symposium*, pages 1–4, 2017. (Cited on page 91.)

[242] P. Thomas, S. Spielman, N. Craswell, and B. Mitra. Large language models can accurately predict searcher preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1930–1940, 2024. (Cited on pages 7, 92, 98, and 132.)

[243] S. Tomlinson, D. W. Oard, J. R. Baron, and P. Thompson. Overview of the TREC 2007 legal track. In *Proceedings of The Sixteenth Text REtrieval Conference (TREC 2007)*, 2007. (Cited on pages 44, 62, and 91.)

[244] N. Tonellotto, C. Macdonald, and I. Ounis. Efficient and effective retrieval using selective pruning. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pages 63–72, 2013. (Cited on pages 3, 63, and 91.)

[245] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. (Cited on pages 18, 19, 29, and 115.)

[246] J. R. Trippas, D. Spina, P. Thomas, M. Sanderson, H. Joho, and L. Cavedon. Towards a model for spoken conversational search. *Information Processing & Management*, 57(2):102162, 2020. (Cited on pages 2, 17, and 21.)

[247] S. Upadhyay, E. Kamalloo, and J. Lin. LLMs can patch up missing relevance judgments in evaluation. *arXiv preprint arXiv:2405.04727*, 2024. (Cited on pages 92, 93, and 98.)

[248] S. Upadhyay, R. Pradeep, N. Thakur, D. Campos, N. Craswell, I. Soboroff, H. T. Dang, and J. Lin. A large-scale study of relevance assessments with large language models: An initial look. *arXiv preprint arXiv:2411.08275*, 2024. (Cited on page 92.)

[249] S. Upadhyay, R. Pradeep, N. Thakur, N. Craswell, and J. Lin. UMBRELA: Umbrela is the (open-source reproduction of the) Bing relevance assessor. *arXiv preprint arXiv:2406.06519*, 2024. (Cited on pages 7, 92, 98, and 132.)

[250] S. Vakulenko, K. Revoredo, C. D. Ciccio, and M. de Rijke. QRFA: A data-driven model of information-seeking dialogues. In *European Conference on Information Retrieval*, pages 541–557, 2019. (Cited on pages 2, 17, and 21.)

[251] S. Vakulenko, E. Kanoulas, and M. de Rijke. A large-scale analysis of mixed initiative in information-seeking dialogues for conversational search. *ACM Transactions on Information Systems (TOIS)*, 39(4): 1–32, 2021. (Cited on page 29.)

[252] S. Vakulenko, N. Voskarides, Z. Tu, and S. Longpre. A comparison of question rewriting methods for conversational passage retrieval. In *European Conference on Information Retrieval*, pages 418–424, 2021. (Cited on pages 70, 72, and 87.)

[253] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. (Cited on page 50.)

[254] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967. (Cited on page 28.)

[255] M. Vlachou and C. Macdonald. Performance predictors for conversational fashion recommendation. In *Proceedings of the Fourth Knowledge-aware and Conversational Recommender Systems Workshop*, pages 91–100, 2022. (Cited on page 87.)

[256] M. Vlachou and C. MacDonald. Coherence-based query performance measures for dense retrieval. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 15–24, 2024. (Cited on page 96.)

[257] N. Voskarides, D. Li, P. Ren, E. Kanoulas, and M. de Rijke. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 921–930, 2020. (Cited on pages 70, 72, 76, 87, and 88.)

[258] S. Wadhwa and H. Zamani. Towards system-initiative conversational information seeking. In *Proceedings of the Second International Conference on Design of Experimental Search & Information REtrieval Systems*, pages 102–116, 2021. (Cited on page 21.)

[259] D. Wang, J. Li, T. Zhu, H. Zhou, Q. Zhu, Y. Wen, and H. Piao. MtCut: A multi-task framework for ranked list truncation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1054–1062, 2022. (Cited on pages 43, 44, 45, 48, 49, 50, 52, 53, 62, and 65.)

[260] K. Wang, J. Tian, R. Wang, X. Quan, and J. Yu. Multi-domain dialogue acts and response co-generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7125–7134, 2020. (Cited on pages 22 and 36.)

[261] L. Wang, J. Lin, and D. Metzler. Learning to efficiently rank. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138–145, 2010. (Cited on page 51.)

[262] L. Wang, J. Lin, and D. Metzler. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 105–114, 2011. (Cited on pages 2, 6, 43, 44, and 63.)

[263] Z. Wang and Q. Ai. Controlling the risk of conversational search via reinforcement learning. In *Proceedings of the Web Conference 2021*, pages 1968–1977, 2021. (Cited on pages 2, 17, 21, 29, 30, and 34.)

[264] Z. Wang and Q. Ai. Simulating and modeling the risk of conversational search. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–33, 2022. (Cited on pages 2, 17, 21, 29, 30, and 34.)

[265] Z. Wang, Y. Tu, C. Rosset, N. Craswell, M. Wu, and Q. Ai. Zero-shot clarifying question generation for conversational search. In *Proceedings of the ACM Web Conference 2023*, page 3288–3298, 2023. (Cited on page 21.)

[266] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2022. (Cited on page 98.)

[267] R. W. White and C. Shah. *Information Access in the Era of Generative AI*. Springer, 2025. (Cited on pages 1 and 3.)

[268] C. Wu, R. Zhang, J. Guo, Y. Fan, Y. Lan, and X. Cheng. Learning to truncate ranked lists for information retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4453–4461, 2021. (Cited on pages 43, 44, 45, 48, 49, 50, 52, and 53.)

[269] C.-S. Wu, S. C. Hoi, R. Socher, and C. Xiong. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, 2020. (Cited on pages 22 and 36.)

[270] Z. Wu, Y. Luan, H. Rashkin, D. Reitter, and G. S. Tomar. CONQRR: Conversational query rewriting for retrieval with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10000–10014, 2022. (Cited on pages 70 and 87.)

[271] J. Xin, R. Nogueira, Y. Yu, and J. Lin. Early exiting BERT for efficient document ranking. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 83–88, 2020. (Cited on page 63.)

[272] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. N. Bennett, J. Ahmed, and A. Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *The Ninth International Conference on Learning Representations*, 2021. (Cited on pages 87, 89, 93, 103, and 106.)

[273] J. Xu, Y. Wang, D. Tang, N. Duan, P. Yang, Q. Zeng, M. Zhou, and X. Sun. Asking clarification questions in knowledge-based question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1618–1629, 2019. (Cited on pages 2, 17, 21, 30, and 34.)

[274] M. Xu, Y. L. Xu, and D. P. Mandic. TensorGPT: Efficient compression of large language models based on tensor-train decomposition. *arXiv preprint arXiv:2307.00526*, 2023. (Cited on page 128.)

[275] Q. Xu, Y. Li, H. Xia, and W. Li. Enhancing tool retrieval with iterative feedback from large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9609–9619, 2024. (Cited on page 134.)

[276] L. Yan, Z. Qin, H. Zhuang, R. Jagerman, X. Wang, M. Bendersky, and H. Oosterhuis. Consolidating ranking and relevance predictions of large language models through post-processing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 410–423, 2024. (Cited on page 98.)

[277] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024. (Cited on pages 3 and 69.)

[278] L. Yang, M. Qiu, C. Qu, J. Guo, Y. Zhang, W. B. Croft, J. Huang, and H. Chen. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 245–254, 2018. (Cited on page 28.)

[279] Y. Yang, A. Shin, M. Kang, J. Kang, and J. Y. Song. Can we delegate learning to automation?: A comparative study of LLM chatbots, search engines, and books. *arXiv preprint arXiv:2410.01396*, 2024. (Cited on page 1.)

[280] C. Ye, L. Liao, F. Feng, W. Ji, and T.-S. Chua. Structured and natural responses co-generation for

conversational search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 155–164, 2022. (Cited on pages 18, 22, 30, 36, and 37.)

[281] S. Yu, J. Liu, J. Yang, C. Xiong, P. Bennett, J. Gao, and Z. Liu. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieva*, pages 1933–1936, 2020. (Cited on pages 70, 72, and 87.)

[282] S. Yu, Z. Liu, C. Xiong, T. Feng, and Z. Liu. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 829–838, 2021. (Cited on pages 70, 72, 73, 76, 77, 87, 94, 120, 121, and 122.)

[283] H. Zamani, W. B. Croft, and J. S. Culpepper. Neural query performance prediction using weak supervision from multiple signals. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 105–114, 2018. (Cited on pages 6, 69, 70, 71, 74, 75, 77, 83, 86, 96, and 106.)

[284] H. Zamani, S. Dumais, N. Craswell, P. Bennett, and G. Lueck. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*, pages 418–428, 2020. (Cited on pages 2, 21, and 70.)

[285] H. Zamani, M. Bendersky, D. Metzler, H. Zhuang, and X. Wang. Stochastic retrieval-conditioned reranking. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 81–91, 2022. (Cited on pages 2, 6, 43, 44, 48, 62, and 130.)

[286] H. Zamani, J. R. Trippas, J. Dalton, and F. Radlinski. Conversational information seeking. *Foundations and Trends® in Information Retrieval*, 17(3-4):244–456, 2023. (Cited on pages 2, 3, 6, 17, 21, and 70.)

[287] O. Zendel, B. Liu, J. S. Culpepper, and F. Scholer. Entropy-based query performance prediction for neural information retrieval systems. In *Proceedings of the QPP++ 2023: Query Performance Prediction and Its Evaluation in New Tasks Workshop co-located with the 45th European Conference on Information Retrieval*, pages 37–44, 2023. (Cited on page 96.)

[288] G. Zhang, K. Chen, G. Wan, H. Chang, H. Cheng, K. Wang, S. Hu, and L. Bai. EvoFlow: Evolving diverse agentic workflows on the fly. *arXiv preprint arXiv:2502.07373*, 2025. (Cited on pages 1 and 135.)

[289] H. Zhang, R. Zhang, J. Guo, M. de Rijke, Y. Fan, and X. Cheng. Are large language models good at utility judgments? In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1941–1951, 2024. (Cited on pages 92, 93, 98, and 99.)

[290] J. Zhang, J. Xiang, Z. Yu, F. Teng, X.-H. Chen, J. Chen, M. Zhuge, X. Cheng, S. Hong, J. Wang, B. Zheng, B. Liu, Y. Luo, and C. Wu. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*, 2025. (Cited on page 135.)

[291] S. Zhang, J. Zhao, P. Wang, Y. Li, Y. Huang, and J. Feng. "Think Before You Speak": Improving multi-action dialog policy by planning single-action dialogs. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4510–4516, 2022. (Cited on page 21.)

[292] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2024. (Cited on page 97.)

[293] X. Zhang, S. Hofstätter, P. Lewis, R. Tang, and J. Lin. Rank-without-GPT: Building GPT-independent listwise rerankers on open-source large language models. *arXiv preprint arXiv:2312.02969*, 2023. (Cited on pages 3, 5, 43, 44, 46, 50, 63, 64, 65, 93, and 99.)

[294] Y. Zhang, C. Hu, Y. Liu, H. Fang, and J. Lin. Learning to rank in the age of Muppets: Effectiveness–efficiency tradeoffs in multi-stage ranking. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 64–73, 2021. (Cited on page 63.)

[295] Y. Zhang, L. Cui, D. Cai, X. Huang, T. Fang, and W. Bi. Multi-task instruction tuning of LLaMa for specific scenarios: A preliminary study on writing assistance. *arXiv preprint arXiv:2305.13225*, 2023. (Cited on page 98.)

[296] Y. Zhang, X. Fan, J. Wang, C. Chen, F. Mo, T. Sakai, and H. Yamana. Data-efficient massive tool retrieval: A reinforcement learning approach for query-tool alignment with language models. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 226–235, 2024. (Cited on page 134.)

[297] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2025. (Cited on pages 4, 7, and 18.)

[298] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing,

H. Zhang, J. E. Gonzalez, and I. Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623, 2023. (Cited on page 98.)

[299] Y. Zhou and W. B. Croft. Ranking robustness: A novel framework to predict query performance. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 567–574, 2006. (Cited on page 96.)

[300] Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 543–550, 2007. (Cited on pages 3, 69, 74, 77, 86, 92, 96, and 104.)

[301] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, H. Chen, Z. Liu, Z. Dou, and J.-R. Wen. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*, 2024. (Cited on pages 4, 7, and 97.)

[302] H. Zhuang, Z. Qin, K. Hui, J. Wu, L. Yan, X. Wang, and M. Bendersky. Beyond yes and no: Improving zero-shot LLM rankers via scoring fine-grained relevance labels. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 358–370, 2024. (Cited on pages 3, 5, 43, 44, 46, 50, 63, 64, and 99.)

[303] S. Zhuang, B. Liu, B. Koopman, and G. Zuccon. Open-source large language models are strong zero-shot query likelihood models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8807–8817, 2023. (Cited on page 64.)

[304] S. Zhuang, H. Zhuang, B. Koopman, and G. Zuccon. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 38–47, 2024. (Cited on pages 3, 44, 50, 52, 63, 64, and 99.)

[305] J. Zou, M. Aliannejadi, E. Kanoulas, M. S. Pera, and Y. Liu. Users meet clarifying questions: Toward a better understanding of user interactions for search clarification. *ACM Transactions on Information Systems (TOIS)*, 41(1):1–25, 2023. (Cited on pages 2, 17, 21, and 70.)

[306] J. Zou, A. Sun, C. Long, M. Aliannejadi, and E. Kanoulas. Asking clarifying questions: To benefit or to disturb users in web search? *Information Processing & Management*, 60(2):103176, 2023. (Cited on pages 2, 17, and 21.)

Summary

Information access systems have been embedded into the capillaries of human society, serving as essential tools for connecting people to information that is crucial for decision-making and taking actions in the world. Many information access systems rely on static workflows, which follow a fixed execution process for all user queries. However, this "one-size-fits-all" approach limits the ability of information access systems to address real-world user queries in complex scenarios that demand adaptive and case-by-case handling. To overcome the constraints of static workflows, prior work has explored agentic workflows for information access, in which one or multiple autonomous agents dynamically adjust execution paths to each user query. This thesis targets optimizing agentic workflows for information access by improving three critical components of the workflows: mixed-initiative strategy planning, ranking strategy planning and ranking result reflection.

The first part of the thesis focuses on optimizing mixed-initiative strategy planning. This part comprises one chapter that focuses on resolving the issue of a narrow scope in system-initiative actions for predicting the timing of system initiative-taking. To solve this issue, this chapter broadens the scope of system-initiative actions by defining and modeling a new task, system initiative prediction (SIP). The SIP task aims to predict the timing of system initiative that covers a broad range of specific system initiative-taking actions. SIP functions as a high-level strategic decision, and effective SIP can enhance related downstream tasks, for instance, by forming a hierarchical decision-making process.

The second part of this thesis aims to optimize ranking strategy planning. This part consists of one chapter that specifically focuses on dynamic per-query re-ranking depth prediction, an important task in ranking strategy planning. Given the limited prior research on dynamic per-query re-ranking depth prediction in the context of the emerging area of large language model (LLM)-based re-ranking, this chapter explores dynamic per-query re-ranking depth prediction in this new context. It conducts a systematic empirical analysis that motivates the need for dynamic per-query re-ranking depths, and explores how to model the prediction in this context.

The third part of the thesis aims to optimize ranking result reflection. This part comprises two chapters that specifically focus on query performance prediction (QPP), a long-standing and effective methodology in ranking result reflection. However, little research explores QPP in the emerging area of conversational search, and little research explores improving QPP accuracy by using the capabilities of LLMs. This part addresses both gaps. The first chapter in this part adapts QPP methods, originally designed for ad-hoc search, to conversational search scenarios, and systematically investigates the performance of existing QPP methods in conversational search. The second chapter enhances QPP accuracy by leveraging LLMs' capabilities.

In summary, the thesis optimizes agentic workflows for information access by addressing limitations in three critical components: mixed-initiative strategy planning, ranking strategy planning, and ranking result reflection.

Samenvatting

Systemen voor toegang tot informatie zijn diep geïntegreerd geraakt in alle lagen van onze samenleving. Veelgebruikte systemen volgen echter meestal een statische werkwijze, waarbij een vaste aanpak wordt gehanteerd voor alle gebruikersvragen. Dit *one-size-fits-all*-model beperkt echter de effectiviteit van systemen bij complexe, realistische scenario's waarin gebruikersvragen een adaptieve en situatie-specifieke behandeling vereisen. Om deze beperking aan te pakken, richt recent onderzoek zich op agentgebaseerde informatietoegang (*agentic workflows*), waarbij één of meerdere autonome agenten de uitvoeringsstappen dynamisch aanpassen aan elke specifieke zoekvraag. Deze thesis richt zich op het optimaliseren van zulke agent-gebaseerde workflows door te focussen op drie cruciale componenten: *mixed-initiative* strategieplanning, planning van de *ranking*strategie, en reflectie op de *ranking*resultaten.

Het eerste deel van dit proefschrift omvat één hoofdstuk dat gericht is op de optimalisatie van *mixed-initiative* strategieplanning. Huidig onderzoek beperkt zich doorgaans tot een smal bereik aan acties waarbij het systeem het initiatief neemt, waardoor de toepasbaarheid van zulke methoden beperkt blijft. Om dit probleem aan te pakken, verbreedt dit hoofdstuk de reikwijdte van mogelijke systeemgestuurde acties door het introduceren van een nieuwe taak: *system initiative prediction* (SIP). De SIP-taak richt zich specifiek op het voorspellen van het moment waarop het systeem initiatief moet nemen, en omvat daarbij een breed scala aan strategische systeemacties, waardoor betere interactie met gebruikers mogelijk wordt.

Het tweede deel van het proefschrift bestaat uit één hoofdstuk dat zich richt op het optimaliseren van de planning van de rankingstrategie. Concreet onderzoekt dit hoofdstuk dynamische, query-afhankelijke her-ordeningsdiepte, een belangrijk onderdeel van de planning van de *ranking*strategie. Aangezien eerdere onderzoek naar dynamische her-ordeningsdiepten in de context van de opkomende grote taalmodellen-gebaseerde benaderingen voor her-ordening beperkt zijn, verkent dit hoofdstuk deze problematiek specifiek binnen systemen die gebaseerd zijn op het gebruik van grote taalmodellen voor her-ordening. Er wordt een systematische empirische analyse uitgevoerd die aantoont waarom dynamische, query-specifieke her-ordeningsdiepten noodzakelijk zijn en hoe deze effectief gemodelleerd kunnen worden.

Het derde deel van het proefschrift richt zich op het verbeteren van de reflectie op *ranking*resultaten, oftewel het automatische beoordelen van de kwaliteit van *ranking*resultaten. Hoewel *query performance prediction* (QPP) een belangrijke techniek is binnen dit domein, is er weinig onderzoek gedaan naar de toepassing ervan binnen het opkomende veld van conversatie-gestuurd zoeken en naar het verbeteren van QPP met behulp van grote taalmodellen. Dit deel van het proefschrift pakt beide kennishiaten aan. In het eerste hoofdstuk worden bestaande QPP-methoden, oorspronkelijk ontwikkeld voor ad-hoc zoekscenario's, aangepast en systematisch geëvalueerd binnen conversatiegebaseerd zoeken. Het tweede hoofdstuk benut vervolgens expliciet de mogelijkheden van grote taalmodellen om de nauwkeurigheid van QPP verder te verbeteren.

Samenvattend levert dit proefschrift bijdragen aan de ontwikkeling van agentgebaseerde systemen voor toegang tot informatie door drie essentiële componenten te verbeteren: *mixed-initiative* strategieplanning, planning van de *ranking*strategie, en reflectie op de *ranking*resultaten.