Performance Prediction for Multi-hop Question Answering

by

Mohammadreza Samadi

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

In this thesis, we study the problem of performance prediction for open-domain multi-hop Question Answering (QA), where the task is to estimate the difficulty of evaluating a multi-hop question over a corpus. Despite the extensive research on predicting the performance of ad-hoc and QA retrieval models, there has been a lack of study on the estimation of the difficulty of multi-hop questions. The problem is challenging due to the multi-step nature of the retrieval process, potential dependency of the steps and the reasoning involved. To tackle this challenge, we propose multHP, a novel pre-retrieval method for predicting the performance of open-domain multi-hop questions. Our evaluation on one of the largest multi-hop QA dataset shows that the proposed model is a strong predictor of the performance of several modern QA systems, outperforming traditional single-hop query performance prediction methods. Furthermore, given the dynamic nature of information retrieval in multi-hop question answering, post-retrieval methods offer a more accurate means of measuring the difficulty of multi-hop questions compared to pre-retrieval methods. Thus, we present a post-retrieval method tailored for multi-hop question answering, highlighting the limitations of other methods proposed in the ad-hoc retrieval domain that may not be applicable in this specific context. We demonstrate that our approach can be effectively used to optimize the parameters of the systems, such as the number of documents to be retrieved, resulting in improved overall retrieval performance.

Preface

This thesis work is a collaborative effort with my supervisor, Dr. Davood Rafiei. While I performed the experiments, analyzed the results, and derived conclusions, Dr. Rafiei provided guidance and feedback on improving the approach, experimental design, presentation, and writing.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Davood Rafiei, for his unwavering guidance and exceptional support throughout the entire journey of my master's thesis. His insights, patience, and encouragement played a pivotal role in shaping the direction of my research. I am also indebted to Dr. Jorg Sander and Dr. Ehab Elmallah, both esteemed members of my thesis committee. Their valuable feedback and comments enhanced the quality of my work.

Furthermore, I would like to extend my heartfelt thanks to my parents, whom I have missed for more than two years. Their unwavering support has been the cornerstone of my academic journey. This accomplishment would not have been possible without their love and guidance

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The task of open-domain Question Answering (QA)–answering questions over a massive collection of documents–has received much attention lately due to the rise of conversational assistant systems such as Apple Siri, Amazon Alexa, Microsoft Cortana, and Google Assistant. Refer to Figure 1.1 for an illustrative example: imagine someone running low on gas, asking Siri for the closest gas station. Without delay, Siri promptly supplies the answer.

Traditional IR models (e.g., BM25 [1]) have been particularly popular as retrievers in this task [2–5] thanks to their simplicity and fast response time. The *retrieve-and-read* framework consists of two components, a retriever and a reader, where the retriever extracts relevant documents from a large collection of documents, and the reader aggregates information in the retrieved documents and extracts the answer [2]. Our work is focused on a particular open-domain QA, referred to as multi-hop QA, where one has to reason with the information that is spread over more than one document in the collection to reach an answer.

Despite extensive prior studies on performance prediction in ad-hoc retrieval systems, to the best of our knowledge, there has been no study on performance prediction of multi-hop questions. Ad-hoc retrieval refers to the process of retrieving relevant information from a large collection of unstructured data in response to a user's query.

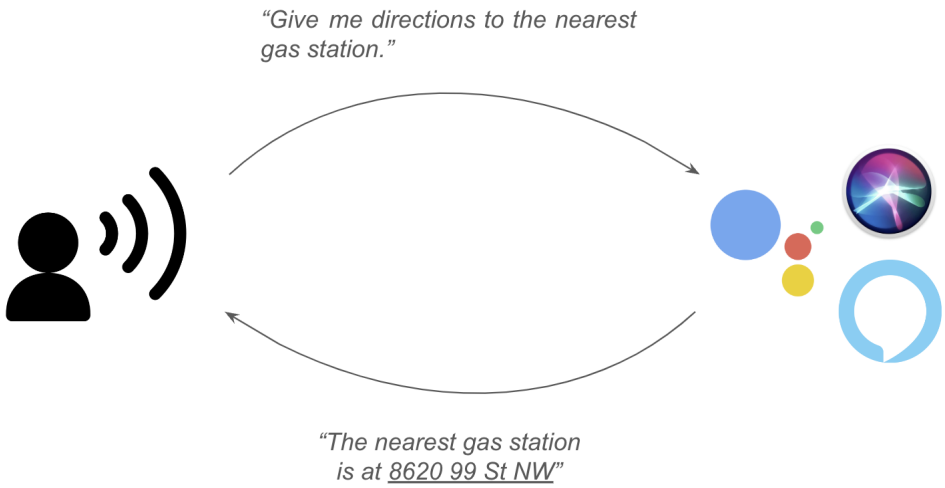


Figure 1.1: One real-world application of QA in our daily life

We claim that multi-hop QA can be different from the ad-hoc retrieval or even question answering due to the following reasons:

- Previous works on performance prediction in ad-hoc retrieval focus on the question-document interaction to capture syntactic or semantic similarity in order to estimate the performance of the retriever. However, in multi-hop QA, missing clues play a crucial role in satisfying the information need in the question.

- In ad-hoc retrieval, the estimator needs to estimate the performance based on a single query. Nevertheless, the nature of multi-hop retrieval is an iterative task that involves updating the question and extracting new information related to the question's information need. Therefore, the performance predictor needs to incorporate multiple questions rather than just a single one.

- Unlike ad-hoc retrieval systems, where we look for a single set of retrieved documents and evaluate the retriever's performance using a single metric (e.g., average precision), in multi-hop QA, we have different sets of retrieved documents corresponding to different questions with varying information needs.

- Although there have been a few studies that focus on performance prediction in question answering systems, they do not consider the unique characteristics of open-domain QA. Thus, there has been a lack of extensive study on performance prediction in the context of open-domain question answering.

1.2 Background

Consider the question "In what year did the young actor who co-starred with Sidney Poitier in Little Nikita die, and what was the cause of death?". To answer this question, one may first retrieve a document that mentions both Sidney Poitier and Little Nikita. This can be, for example, a document that lists the cast members of Little Nikita. From this document, one may find out that the question is referring to River Phoenix. The document about Little Nikita is less likely to give more information about River Phoenix though. Hence in another retrieval step, the Wikipedia document of River Phoenix may be retrieved to find out that the actor died in 1993 due to drug intoxication. This is one form of multi-hop, referred to as a *bridge question*. It is called a bridge question because we need to find a missing entity, River Phoenix, that serves as a bridge to the information need expressed in the question (cause of death). Another form of multi-hop include questions that must compare pieces of information or reason over facts spread in multiple documents. For instance, the question "Were Stanley Kubrick and Elio Petri from different countries?" is a *comparison question*. First, both documents about Stanley Kubrick and Elio Petri will be retrieved, and after a reasoning process that compares their nationalities, the question can be answered.

The examples mentioned above demonstrate that answering a question only requires two documents to fulfill the information need. However, in the case of multi-hop questions, the answer may require more than two documents, or in a broader sense, n documents may be needed. This leads to the notion of creating more complex questions by combining a sequence of bridge and comparison reasoning. An

instance of such a question is "Do the director of the film 2001: A Space Odyssey and the director of the film The Assassin belong to the same country?". This question involves a 4-hop *bridge-comparison* scenario, where one must deduce the bridge entities and perform comparisons to obtain an answer.

The models that utilize the retrieve-and-read framework [2] in open-domain QA fail in multi-hop questions for a few reasons. First, the clues for answering the questions are often spread over multiple supporting documents, and the retriever cannot find all required clues in one step of the retrieval, as shown in our Little Nikita example. Hence an iterative retrieve-and-read process may be needed [6]. Second, many questions require some form of reasoning over the facts described in supporting documents, and standard retrievers are oblivious to such a chain of reasoning when the facts are spread in multiple documents.

Several datasets have been proposed for the task of answering multi-hop questions, taking into account the aforementioned types of such questions. HotpotQA was introduced by Yang *et al.* [7] as a large-scale dataset that includes questions which demand reasoning over one or two Wikipedia documents. The questions in HotpotQA are categorized as either bridge or comparison retrieval path types, as they can be answered by referring to a maximum of two documents. A shortcoming of HotpotQA is the lack of a complete explanation for the reasoning process from the question to the answer. To overcome this challenge, Ho *et al.* [8] present 2WikiMultiHopQA, a dataset that includes 2-hop and 4-hop questions. Unlike HotpotQA, 2WikiMultiHopQA includes a new type of information called evidence, which is a set of triples collected from Wikidata. In addition to both bridge and comparison retrieval paths, 2WikiMultiHopQA also includes bridge-comparison questions that require the inference of a bridge entity and a comparison to be made. Trivedi *et al.* [9] claim that models trained on previous benchmarks such as HotpotQA can rely on shortcuts to find a correct answer due to reasons such as overly specific question and train-test leakage. To develop a high-quality multi-hop dataset that addresses these issues, they introduce a bottom-up

Dataset	Question Type	#Examples
HotpotQA [7]	1-hop	18,089
	2-hop	94,690
	Total	112,779
2WikiMultiHopQA [8]	2-hop	152,446
	4-hop	40,160
	Total	192,606
MuSiQue-Ans [9]	2-hop	16,899
	3-hop	5,910
	4-hop	2,005
	Total	24,814

Table 1.1: Number of examples in different datasets for various question types, including 1 to 4 hops.

approach for constructing the MuSiQue-Ans dataset. MuSiQue-Ans includes more complex questions that require 2 to 4 pieces of information to answer. Table 1.1 shows the statistics of HotpotQA, 2WikiMultiHopQA, and MuSiQue-Ans.

The difficulty of answering questions can be affected by various factors, including the information need in the question and the statistics of the corpus (e.g., Wikipedia). We can categorize a question as difficult when the retriever fails to retrieve relevant documents effectively for that particular question. Based on this definition, an exploration of the task of query performance prediction is justified. Query performance prediction refers to the process of estimating the probable effectiveness or quality of search results for a given query prior to the actual retrieval process. It involves predicting the extent to which a set of documents or search results will align with the user's information needs and expectations. Drawing upon the comprehensive discourse presented by Carmel and colleagues [10], it becomes evident that the multifaceted landscape of existing prediction approaches can be adeptly organized into a unified taxonomy, as illustrated in Figure 1.2.

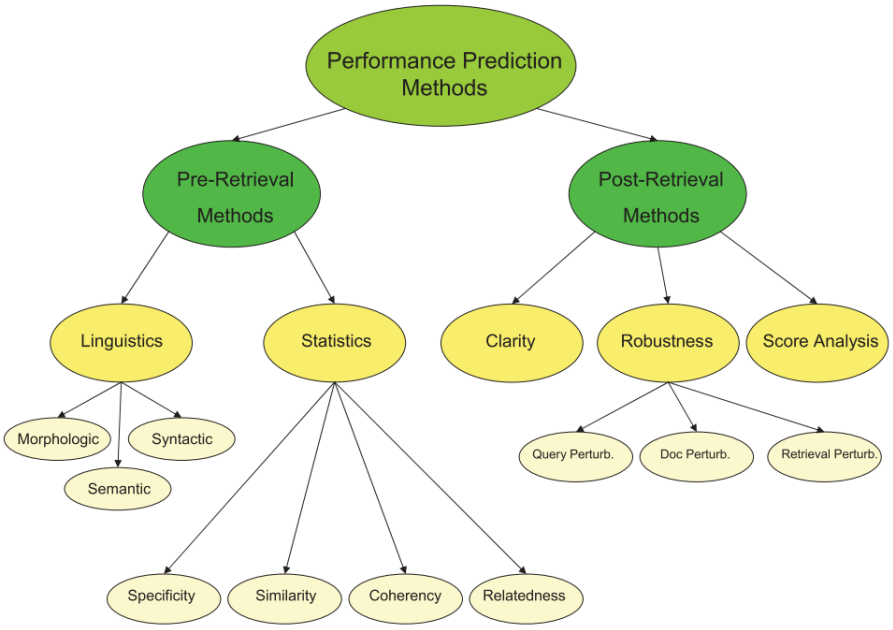
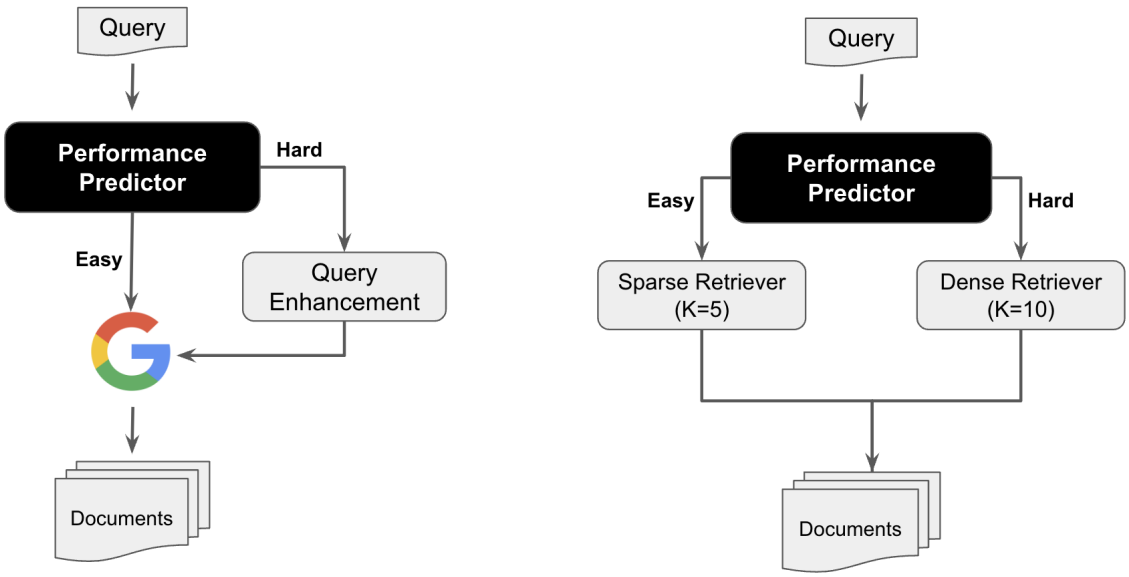


Figure 1.2: Taxonomy of existing query performance prediction methods [10].

When it comes to answering multi-hop questions, relevant documents need to be retrieved from different sources before attempting to answer the question. These documents are then used to derive the answer, making the process more complex and challenging.

The primary focus of this study is to assess the level of difficulty involved in finding all the necessary information required to answer multi-hop questions. Assessing the difficulty holds significant implications for enhancing retrieval systems in various aspects. Gaining insights into the difficulty level of queries empowers us to optimize retriever performance through a multitude of strategies (Figure 1.3b). One of its prime applications lies in the initiation of targeted query expansion for queries that are underspecified or ambiguous, thereby refining the retrieval outcomes (Figure 1.3a). This predictive understanding can also facilitate query enhancement before the search even begins, enabling users to formulate more effective queries initially. Furthermore, the ability to anticipate query difficulty contributes to the refinement of user feedback mechanisms. Users can be presented with personalized suggestions to improve their query based on predicted retrieval challenges.



(a) Query Enhancement (b) Optimizing Retrieval Process

Figure 1.3: Two applications of performance prediction

The integration of these approaches underscores the value of query performance prediction as a proactive strategy for enhancing retrieval effectiveness and user satisfaction. Consequently, investigating this field becomes imperative, offering a deeper understanding of query intricacies and opening avenues for innovative approaches to search optimization.

1.3 Problem Definition

The specific question we investigate in this study is if the performance of a multi-hop question can be predicted and if such a prediction is a good proxy for the performance of multi-hop QA systems, some of which are more complex. Query Performance Prediction (QPP) plays an important role in resource allocation and evaluating the performance of different systems. For example, QPP can assist a search engine in allocating additional resources to more difficult questions. Even though QPP has been extensively studied, with methods ranging from statistical approaches [11–14] to more recent neural models [15–17], we are not aware of any such work on multi-hop

questions.

There are two paradigms for the task of performance prediction. In the pre-retrieval paradigm, the predictor estimates the performance of a retriever for a particular query based on both the query's terms and the statistics of the document collection. This approach, known as pre-retrieval Query Performance Prediction (pre-retrieval QPP), can be formally viewed as a function where the inputs are a question q and a large set of documents D_{col}. The goal is to estimate the query's performance without executing an actual retrieval process, i.e.,

$$\hat{y}_q \leftarrow \mu_{pre}(q, D_{col}). \tag{1.1}$$

The estimate here is independent of the retriever. Although pre-retrieval methods have the advantage of low processing costs, the query alone is often not expressive enough for reliable prediction of the quality of the search results. To tackle this difficulty, post-retrieval methods have been proposed as an alternative to measure the quality of the retrieved results (D_q), based on the given query. These methods may lead to stronger performance prediction as they have access to the retrieved documents. A post-retrieval predictor can be formulated as the following function:

$$\hat{y}_q \leftarrow \mu_{post}(q, D_q, D_{col}, \mathcal{M}) \tag{1.2}$$

In contrast to pre-retrieval methods, post-retrieval approaches are highly reliant on specific retrievers and necessitate updates when the retriever or its parameters are altered.

In this thesis, we aim to investigate the problem of measuring the difficulty of multi-hop questions in both QPP paradigm. First, we introduce a pre-retrieval method for multi-hop QA called *multHP*, which utilizes retrieval paths to decompose each question into a few retrieval steps, starting from the question and ending with a document that has an answer. Our method is independent of any specific types of retrievers, allowing us to predict the performance of various retrievers. Additionally, our approach

builds upon the concept where the retriever selects documents randomly based solely on the information of the most specific term in the question. This approach, referred to as the specificity method, is well-documented in the literature. We show that retrieval path types can be detected from questions and that the performance of a multi-hop question can be estimated based on the performance of its retrieval steps, for which corpus-based statistics are more likely to be available and standard QPP models may be applicable. Considering the dynamic nature of information retrieval in multi-hop QA, post-retrieval methods can measure the difficulty of a multi-hop question much more precisely compared to pre-retrieval methods. Therefore, we introduce *Unified Model*, a post-retrieval method specifically designed for multi-hop QA and show that other methods proposed in ad-hoc retrieval may not be applicable in this domain. It is important to note that our work is constrained by the abundance of resources available for 2-hop retrieval [6, 7, 18] and the scarcity of comprehensive studies conducted on n-hop retrieval. As a result, our focus is limited to performance prediction exclusively for 2-hop questions.

1.4 Contribution

Our contributions can be summarized as follows:

- We define the task of performance prediction in multi-hop question answering. To the best of our knowledge, our work is the first to study performance prediction specifically in the domain of multi-hop QA.
- We introduce retrieval paths as a representation of the sequential retrieval steps taken by a QA system, starting from a multi-hop question and leading to an answer. Building upon these paths, we develop a pre-retrieval performance prediction model that leverages retrieval paths to estimate the level of difficulty associated with answering a multi-hop question.

- We further propose a post-retrieval approach for multi-hop question performance prediction, utilizing semantic features of retrieved documents and the given question.

- Our evaluation reveals that the proposed methods serve as strong predictors of the actual performance of several Open-domain QA retrievers, including both sparse and dense models. Furthermore, our methods outperform the relevant QPP baselines from the existing literature.

1.5 Outline

The rest of this thesis is structured as follows. In Chapter 2, we provide a comprehensive review of the existing literature on Multi-hop QA and Query Performance Prediction. This chapter serves as a foundation for understanding the current state of the field and the research gaps that exist. Chapter 3 presents our retrieval paths and our proposed method of estimating difficulty scores using a pre-retrieval paradigm. Additionally, we present the results of our experiments, demonstrating the superiority of our method compared to previous pre-retrieval methods proposed for ad-hoc retrieval. In Chapter 4, we introduce our post-retrieval method, which builds upon the idea of retrieval-paths in our pre-retrieval framework. We outline the improvements made and discuss the limitations. Finally, Chapter 5 concludes this study by summarizing the key findings and contributions. We also discuss potential future research directions, highlighting areas where further investigation can expand upon the work presented in this thesis.

Chapter 2

Literature Review

Multi-hop question answering is a complex task that involves finding answers to questions that require multiple pieces of information from various sources. This task can be considered as an intersection of two domains, namely information retrieval (IR) and question answering (QA). On the retriever side, the focus is on answering the question *"How relevant information scattered between multiple documents can be retrieved from a large collection of documents?"*. The line of studies on question answering aims to answer the question *"How scattered information in a limited set of documents can be processed to answer the question?"*. In this study, our primary focus lies on the former question as it pertains to the performance of the retriever, while the second question is associated with the performance of the QA module.

Query performance prediction (QPP) holds a significant position in the literature of information retrieval (IR), aiming to address the question of *"What would be the performance of a retriever for a given query?"*. QPP methods find applications in various domains, including search engine optimization, query reformulation, resource allocation, and user assistance. Existing approaches in the QPP literature can be classified into two main categories: pre-retrieval and post-retrieval. Both categories will be thoroughly discussed in the following sections.

2.1 Open-domain Multi-hop Question Answering

The release of large-scale datasets for multi-hop QA [7–9, 19] has invigorated the research interest in this area. The studies predominantly utilize an iterative *retrieve-and-read* framework to secure the supporting documents, where the performance is closely linked to that of the retriever (see [20] for a survey).

2.1.1 Iterative retrievers

An essential step in iterative retrievers is extracting the clues from the documents retrieved for the current hop and updating the query for the next hop. Qi *et al.* [6] propose GoldEn, an iterative *retrieve-and-read* approach that uses both the question and the retrieved documents at each step to generate a query for the next step to find all missing entities. The query generation is done using a supervised model that is trained on the semantic overlap between retrieved contexts and the documents to be retrieved. This approach is shown to perform well on HotpotQA when integrated with DrQA reader [2]. Since missing entities play an important role in the retrieval process, Xiong *et al.* [21] employ entity linking to extract bridge entities from the retrieved documents instead of query generation. They leverage a hybrid TF-IDF and BM25 approach to retrieve the first set of documents. Similarly, Zhang *et al.* [22] propose an iterative document reranking method, where they retrieve documents using TF-IDF. For each subsequent hop, the authors update the question with extracted information and repeat this process until either an answer is found or a maximum number of hop-retrieval is reached. It is worth noting that these approaches all utilize sparse retrievers.

Due to the great success of deep contextualized language model (e.g., BERT[23]) in open-domain retrieval tasks, such as DPR [24], dense retrievers are used in multi-hop QA for better capturing the query semantics as well. Xiong *et al.* [18] propose a dense retriever within an iterative "read-and-retrieve" framework that extracts top-

k document paths, including chains of documents that lead us to all the required information. Relevant documents are retrieved based on their MIPS between the query and document dense vectors. Zhang *et al.* [25] take another step further and developed a pure rank-based framework that generates document sequences for each path and reranks paths for answer extraction. This framework considers both the local and global information of hops during the ranking process, ensuring the identification of the best paths among all steps. In an attempt to address the explainability of dense retrievers, Wu *et al.* [26] represent each document with a set of triple facts in the form of <subject, predicate, object>. In each retrieval step, the question is updated with a fact from a supporting document, instead of using the whole document.

Furthermore, there is a line of study [27, 28] that leverages both sparse and dense retrievers. Sparse retrievers, due to their ability to quickly retrieve documents based on syntactic information, have been used to narrow down the search space. Subsequently, dense retrievers extract documents with high semantic similarity to the given question. Feldman and El-Yaniv [27] employ a TF-IDF retriever to narrow the search space and rank documents based on the maximum inner product between the encoded question and the top paragraphs. Similarly, Nie *et al.* [28] introduce a pipeline system for Machine Reading at Scale (MRS), which utilizes hierarchical semantic retrieval at both paragraph and sentence levels following the initial narrowing of the search space using TF-IDF.

Unlike previous iterative studies, Asai *et al.* [29] introduce an approach that does not require determining the number of reasoning steps, as it can adapt to a varied number of hops. Their approach is based on constructing a graph of paragraphs from a large collection, such as Wikipedia, using hyperlinks to represent the relationships between documents. They train a retriever using a recurrent neural network to score each reasoning path in the graph, and choose top-1 evidence path for the QA module. Zhu *et al.* [30] present an adaptive iterative retriever for open-domain QA. They claimed that applying the same retrieval function (e.g., BM25, Dense Paragraph

Retriever, or hyperlink) multiple times may not extract new relevant information, so they proposed an adaptive retriever that can determine which retrieval function to use at each step based on the information need.

Since answering a multi-hop question involves an iterative process, our study primarily focuses on methods that utilize the iterative retrieve-and-read framework. Additionally, considering that dense retrievers employ semantic features while sparse retrievers primarily extract syntactic features, our aim is to analyze the impact of different retriever types. Our proposed Query Performance Prediction (QPP) method estimates the performance independent of the specific retriever model being used. Therefore, for our evaluation, we select MDR as our fixed dense retriever and GoldEn as our sparse method.

2.1.2 Using entities in retrievals

The main challenge in retrieving supporting documents for multi-hop QA is missing entities and that the question text may not be enough to find all relevant documents. Several studies utilize the relationship between different entities, including those explicitly mentioned and those missing, to retrieve relevant evidence. Das *et al.* [31] propose an entity-centric retrieval method. They extract an initial set of relevant documents for a question using a sparse retriever (e.g., BM25) and then link all entity mentions in the retrieved paragraphs to the corresponding paragraphs in the Wikipedia corpus. The extracted paragraphs are ranked based on their relevance to the question using a BERT-based reranker. Ding *et al.* [32] propose a dual-system for extracting the supporting clues and an answer. System 1 extracts all question-relevant entities and candidate answers from extracted paragraphs and encode their semantics. System 2, conduct the reasoning procedure and collects clues to help System 1 to extract better entities and answers for the next hop. Shao *et al.* [33] reduce the search space by retrieving the top-k paragraphs using DrQA [2]. Additionally, they include both paragraphs whose titles are mentioned in the query and paragraphs that

are connected through hyperlinks. Other works [34, 35] also incorporate entity-level relations in their multi-step retrievers.

Similar to the aforementioned approaches, our performance prediction model also incorporates entities mentioned in questions in its estimation. Since sparse retrievers utilize the term distribution of a query and corpus statistics to retrieve relevant documents, we demonstrate that the difficulty of a question can be estimated by considering its entity mentions.

2.2 Query Performance Prediction

Numerous studies have been conducted on query performance prediction, and various methods have been proposed based on different hypotheses and principles. Kurland *et al.* [36] establish a mathematical foundation for this task and classify the query performance prediction methods into pre-retrieval and post-retrieval. Pre-retrieval methods attempt to predict query performance before the retrieval process, while post-retrieval methods focus on predicting performance after the results have been obtained. In the following subsections, we review the literature on these two paradigms and discuss their strengths and limitations.

2.2.1 Pre-retrieval Methods

Pre-retrieval approaches are specifically designed to predict query performance without executing a retrieval step. These methods rely solely on the content of the query itself and corpus statistics. Statistical pre-retrieval query performance prediction methods may be classified into several groups including similarity-based, coherence-bsed and specificity-based [10]. Similarity-based approaches estimate the query performance based on the similarity between the query and the document collection. Collection Query Similarity [37] assumes that the collection can be treated as a single large document, and measures the similarity of a query to this document. Queries that are more similar to the collection are considered easier to evaluate.

Coherency-based approaches, such as Query Coherence Score [38] and VAR [37], measure the inter-similarity of documents containing the query terms. Query Coherence Score [38] reflects the average pairwise similarity between all pairs of documents. Computing a coherence score is usually has a high computation cost. VAR [37] is an alternative approach that measures the variance of the term weights (e.g., TF-IDF) over the documents containing it in the collection. If the variation of term weight distribution is low, it will be harder for the retrieval model to distinguish between highly relevant and less relevant documents, making the query more difficult. Relatedness-based methods, such as Point-wise Mutual Information [39], make use of the co-occurrence of query terms in the collection. The frequency of co-occurring query terms indicates how easy or difficult the query is to answer.

Finally, specificity-based methods calculate the specificity of query terms based on their distribution over the given collection of documents. Examples of specificity-based methods include Inverse Document Frequency [40] and Simplified Coherence Score [41]. These methods aim to determine the relevance of query terms to the document collection, which in turn affects the predicted performance of the query. The main shortcomings of frequency-based specificity methods is ignoring semantic equivalency between query and corpus terms. To tackle this challenge, Roy *et al.* [42] introduced a specificity-based metric, called $P_{clarity}$, based on the idea that the number of clusters around the neighborhood of a term may indicate its specificity. In another similar idea, Arabzadeh *et al.* [43] proposed three specificity metrics: neighborhood-based, graph-based, and cluster-based. These metrics are based on neural embeddings and the geometric relations between the embedding vectors.

Being a pre-retrieval method, our approach also uses query terms and corpus statistics in the estimation process but it differs from previous specificity-based approaches in two important aspects. First, our approach does the estimation in the context of retrieval paths and question types, allowing us to provide more accurate estimation for each retrieval path. Second, given the nature of questions in open-domain set-

tings, we aim at using in our estimation salient question terms that play a role in an effective retrieval, rather than considering all question terms.

2.2.2 Post-retrieval Methods

Unsupervised QPP

In contrast to pre-retrieval approaches, post-retrieval methods estimate the difficulty of a query after retrieving a result set. The post-retrieval methods have been classified into three paradigms [10]: clarity-based, robustness-based, and similarity distribution. Clarity-based approaches evaluate how well the retrieved results relate to the query's topics, where a strong relationship between the results and the query indicates good results. Cronen-Townsend *et al.* [40] present Clarity Score, as a measures of query ambiguity, based on the idea that if the query is unambiguous, the top-ranked results will have a cohesive topic. In another study, Carmel *et al.* [44] show that the difficulty of a question strongly depends on the distances between three components: the textual expression of the query, the set of relevant documents, and the entire collection. They use Jensen-Shanon divergence (JSD) to measure the distance and calculate a difficulty score. Hauff *et al.* [11] analyze the sensitivity of a few query prediction algorithms, both pre-retrieval and post-retrieval, in the context of a web search engine, showing that these algorithms are sensitive to the choice of parameters and the retrieval method. They further propose an improved clarity score for web collections.

The robustness paradigm focuses on evaluating the resilience of retrieved results when subjected to perturbations in the query, retrieved results, and retrieval method. The aim is to determine how well the retrieved documents withstand changes and variations. The level of difficulty posed by the query is inversely proportional to the robustness of the results. In other words, if the retrieved documents are more robust, the query is less challenging. Zhou and Croft [45] introduced Query Feedback (QF) as a way to evaluate the robustness of a query by using query perturbation. This

involves updating the query with terms from the retrieved results, and then retrieving a second list of results based on the updated query. The degree of overlap between the two result lists is then used to determine a robustness score for the query. The higher the degree of overlap between the two lists, the easier the query is considered to be.

Score-based methods aim to analyze the score distribution of the retrieved documents in order to determine the level of difficulty of a query. Weighted Information Gain (WIG) [45] is a technique used to determine this level of difficulty by calculating the divergence between the mean retrieval score of the top-k documents and that of the collection. Essentially, the more closely related the top-k documents are to the query, in terms of their similarity to a general non-relevant document (the collection), the more successful the retrieval is likely to be. Normalized Query Commitment (NQC) [14] estimates the potential query drift in the top-retrieved documents by measuring the standard deviation of retrieval scores in these documents, and normalizing it by the score of the entire collection. A high standard deviation indicates lower query drift of the top-k documents, which leads to better query performance.

Supervised QPP

In recent years, the success of neural-based models in various domains of information retrieval and natural language processing has led to a growing interest in formulating post-retrieval query performance prediction as a supervised learning problem. This approach involves predicting the quality of retrieved results, measured in terms of, for example, in terms of average precision, based on the given question and the retrieved documents.

Zamani *et al.* [46] proposed a novel neural-based model, called NeuralQPP, which leveraged multiple statistical post-retrieval methods such as Clarity, Normalized Query Commitment, and Utility Estimation Framework as weak supervision signals. Arabzadeh *et al.* [16] proposed an alternative approach to extracting hand-crafted features from

retrieved documents. They instead fine-tuned a BERT [23] model to generate contextualized embeddings specifically for the task of predicting query performance. They developed two neural network architectures: cross-encoder and bi-encoder networks. The cross-encoder network allowed for deeper associations between the query and documents by processing them together in a single model. However, this architecture was slower during inference due to the lack of offline computation. To address this issue, they introduced a bi-encoder architecture, which utilized a Siamese network with parallel encoders for queries and documents. While this architecture reduced inference time, the encoders needed to be updated separately.

While other works aim to predict a real number as a model performance on a specific query, Datta *et al.* [17] argue that measuring the difficulty of a query requires considering its relative difficulty compared to other queries rather than evaluating it in isolation. To address this issue, they introduced a supervised model that can learn a comparison function for evaluating the relative difficulty between pairs of queries. Moreover, they utilized an interaction-based model, rather than a representation-based one, to learn the correlation between the query and the top-k documents. Building upon the previous idea, Chen *et al.* [47] propose a group-wise Query Performance Prediction approach that incorporates cross-document and cross-query context. This extended framework allows estimating the query difficulty for not just two queries, but for multiple queries within a group. By leveraging the context provided by multiple documents and queries together, it can gain a more comprehensive understanding of query difficulty and improve the accuracy of performance estimation. Datta *et al.* [48] present qppBERT-PL, an end-to-end neural cross-encoder trained in a point-wise manner on individual queries, but in a list-wise manner over the top ranked documents. Unlike previous works, they formulate the performance prediction as a classification problem, aiming to estimate the number of relevant documents in each document list.

While the aforementioned supervised QPP methods primarily focus on ad-hoc re-

trieval systems, it is worth noting that there are multiple studies conducted in the context of Question Answering (QA). This is due to the crucial role that retrieval systems play in open-domain QA pipelines. Retrieval systems are essential for retrieving relevant information from a large corpus, which is then used by downstream QA components to generate answers. The performance and effectiveness of retrieval systems significantly impact the overall performance of open-domain QA systems. Consequently, researchers have dedicated efforts to investigate and enhance retrieval systems specifically tailored for the QA domain.

In pursuit of this goal, Krikon *et al.* [49] introduced a framework for predicting the effectiveness of a set of passages retrieved for a given question. They decomposed the effectiveness into two components: (1) the probability that the retrieved passages satisfy the information need of the question, and (2) the probability that the passages contain the answers. The first component was estimated using post-retrieval query performance predictors, such as the clarity score [40], as well as other metrics proposed in previous studies [14, 45]. These predictors assess the degree to which the retrieved passages fulfill the information requirements of the question. The second component relied on the presence of named entities within the passages that have the potential to answer the question. By considering the occurrence of relevant named entities, the framework aims to gauge the likelihood that the retrieved passages contain the answers sought by the question.

Hashemi *et al.* [15] study the problem of performance prediction for non-factoid questions and develop a neural model that estimates the performance based on three components: (1) the scores assigned to candidate answers by the QA system, (2) query performance, estimated using pre-retrieval QPP models, and (3) the content of top k retrieved answers.

Roitman *et al.* [50] examined the decision-making process of accepting the best answer from a search engine for a user query. They explored the utility of different feature sets in this context, including learning-to-rank features such as the answer's

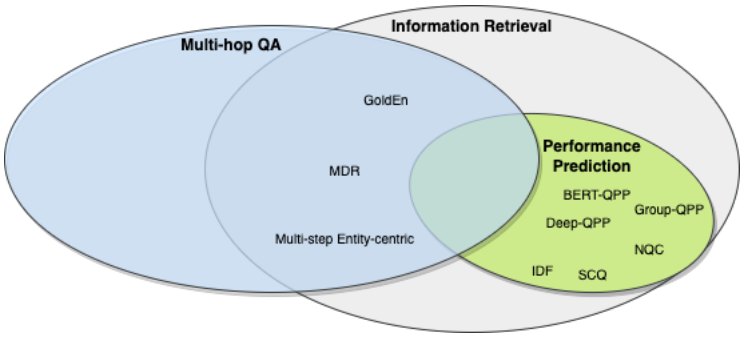


Figure 2.1: Comparative placement of our study in performance prediction and multi-hop QA domains

TF-IDF score. They also incorporated query performance prediction (QPP) methods, such as AvgIDF and WIG, to assess the quality of the retrieved answers. Additionally, they employed answer-level QPP measures (e.g., ColScoreDiff [51] and MaxPsg1 [52]) to further evaluate the relevance and effectiveness of candidate answers. Their investigation aims to gain insights into the decision-making criteria for accepting the best answer from a search engine based on these various feature sets.

Considering all the post-retrieval methods proposed for ad-hoc retrieval and question answering, an interesting question is if these approaches can be applied to the multi-hop question task. The retrieval step holds a critical role in this task, as it involves retrieving relevant information from a large corpus to answer complex, multi-step questions. By estimating the difficulty of these questions beforehand, we can make informed decisions on parameter settings and choose optimal strategies to achieve the best results while maintaining efficiency. Query performance prediction in the context of multi-hop questions can guide the design and optimization of retrieval systems, leading to more effective and efficient solutions for this challenging task.

As Figure 2.1 shows, our work is positioned at the intersection of multi-hop QA and performance prediction. We aim to leverage the concept presented by [31], utilizing question entities for performance prediction and analyzing its effectiveness across various retriever types.

Chapter 3

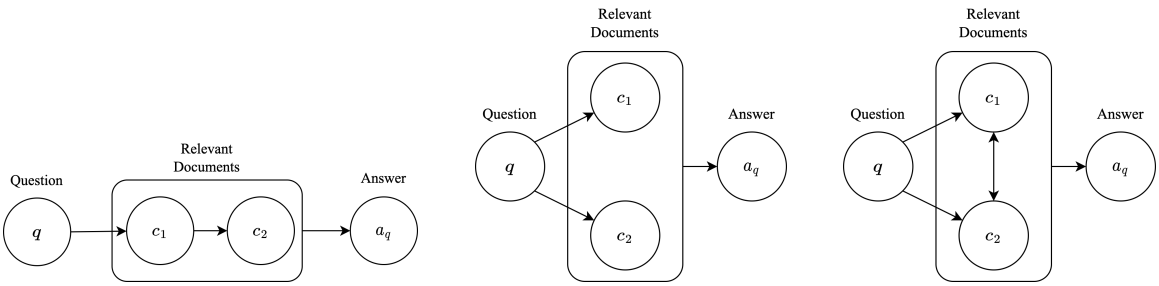
Pre-retrieval QPP

Our approach to estimate the performance of a multi-hop question is based on estimating the performance of its hops and combining the estimates. In this section, we present retrieval paths as steps a QA system must go through to gather evidence for a multi-hop question. We then analyze those paths, in terms of the difficulty of retrieving evidence under each path and present our approach for estimating the performance.

3.1 Retrieval Paths

For a multi-hop question, one must collect evidence from two or more relevant documents to be able to answer the question. In open domain settings, those relevant documents are retrieved from a large collection of documents. A challenge that is unique to multi-hop questions is that the question may not have enough information to retrieve all relevant documents. Consider a question that must gather information from two supporting documents d_1 and d_2. The relationship between those documents and the question can fall into the following cases:

- There is enough information in the question that allows both supporting documents d_1 and d_2 to be retrieved. These supporting documents may or may not be closely related.
- The question has enough information to allow only one supporting document



(a) Bridge (b) Comparison (c) Mixed

Figure 3.1: Retrieval path types

d_1 to be retrieved; the other supporting document can be retrieved with the information in both d_1 and the question.

The *relatedness* between the question and the supporting documents and between two documents may be established syntactically, based on common terms and phrases. The relatedness may also be defined at the semantic level, for example, based on the embedding of the question and those of the documents. The former is the approach taken by a sparse retriever whereas the latter is employed in dense retrievers. For the simplicity of our analysis, we assume the relatedness is defined syntactically.

Consider a graph where each node denotes a document or a question and each edge indicates the relatedness between two documents. The steps a retriever takes to reach from a question to its answer can be described as a set of paths, all starting from the question and ending with an answer. Figure 3.1 shows the set of retrieval paths that can be formed between a question and its two supporting documents. The path in Figure 3.1a, referred to as a *bridge retrieval* path, describes a scenario where the second document cannot be easily retrieved without retrieving the first document. This may describe, for example, a question about an entity that is not explicitly mentioned in the question, but the question provides enough context to retrieve document d_1 where the entity name is given. The path in Figure 3.1b dubbed as *comparison retrieval* path, represent a scenario where both supporting documents can be retrieved with the information given in the question, but the reasoning for an

answer needs both documents. For example, the question may ask if two persons, each described in a separate document, have the same nationality. It should be noted that, despite the naming, the reasoning can take forms other than a comparison such as an aggregation function. For example, each document may give financial data about a company and the question may ask for the total assets of a parent company that owns two companies. For some questions, the retrieval path type may not be known or easy to detect. This can happen when the two documents are closely related and and they are also closely related to the question text, as depicted in Figure 3.1c. This is called *mixed retrieval* path meaning that the retriever may consider it as *bridge* or *comparison*. Retrieval paths may be generalized to n-hop questions, with n nodes representing the supporting documents and the edges describing possible steps a retriever can take.

3.2 Retrieval Paths in HotpotQA

To study the prevalence of retrieval paths, we construct those paths for questions in HotpotQA, one of the largest multi-hop QA datasets that is public. For each question in the dataset, the question and its two supporting documents form the nodes of the graph. An edge is added between d_1 and d_2, deeming them as relevant, if there is a common term between the two documents and the probability of finding that term in an arbitrary document is low (i.e. below a threshold P_{thr}).

Similarly, edges are added between q and d_1 and between q and d_2 if a relevance can be established. The value of P_{thr} may be determined experimentally based on the corpus statistics. If P_{thr} is close to 0, most of the retrieval paths will form incomplete graphs. On the other hand, when P_{thr} is close to 1, there will be edges for more general terms or even stop words and the number of mixed types will increase.

With P_{thr} is set to 0.001, our study of multi-hop questions in the training set of HotpotQA reveals that about 20% of the questions demonstrate a bridge retrieval path, whereas the number of questions that show a comparison retrieval path is

Ret. path	Example
Bridge	**Question**: What year was the actor that co-starred with Sidney Poitier in *Little Nikita* born? **Context 1:** Little Nikita is a 1988 American cult drama film directed by Richard Benjamin and starring River *Phoenix* and Sidney Poitier. The film marks the first collaboration between *Phoenix* and Poitier (the second being Sneakers in 1992). **Context 2:** River Jude *Phoenix* (born River Jude Bottom; August 23, **1970**, October 31, 1993) was an American actor, musician, and activist. He was the older brother of *Rain Phoenix*, Joaquin *Phoenix*, Liberty *Phoenix*, and Summer *Phoenix*. **Answer**: **1970**
Comparison	**Question**: Were *Stanley Kubrick* and *Elio Petri* from different countries? **Context 1**: *Elio Petri* (29 January 1929 2013 10 November 1982) was an Italian political filmmaker. **Context 2**: *Stanley Kubrick* (July 26, 1928 March 7, 1999) was an American film director, screenwriter, producer, cinematographer, editor, and photographer. He is ... extensive set designs, and evocative use of music. **Answer**: **yes**

Table 3.1: Two examples of retrieval paths from the HotpotQA dataset. The named entities mentioned in both the question and the contexts, shown in red, may assist the retriever in finding the supporting documents. The common entities between the two contexts, shown in violet entities, may also help.

around 14%. A majority 63% of questions show a mixed retrieval path with enough overlapping terms between the questions and their supporting documents. In the development set, our findings indicate that approximately 19% of the questions exhibit a bridge retrieval path, around 15% show a comparison retrieval path, and 63%, display a mixed retrieval path. Clearly, these are some rough estimates, based on our threshold setting of relatedness (P_{thr}). Table 3.1 gives an example of each retrieval path. For less than 3% of questions in both training and development sets, no retrieval path could be detected due to the lack of more specific common terms. For example, the question *"Who released the 2012 record of Red?"* forms an incomplete graph, because *Red*, the name of an album, is also commonly used as a color.

Our analysis of the dataset also reveals that some questions can be answered with only one supporting document, and they are not really multi-hop. The retrieval paths for these questions show a strong relatedness edge between the question and one supporting document and the answer also appears in the same document. While the second supporting document is related to the question, it is not required for extracting the answer. For example, the question "Who was known by his stage name Aladin and helped organizations improve their performance as a consultant?" can be answered by the document with title "Eenasul Fateh" and text "Eenasul Fateh also known by his stage name Aladin, is a bangladeshi-british cultural practitioner, magician, live artist and former international management consultant" in the content. The second document given for this question is "Management consulting" which is not necessary to answer the question.

3.3 Difficulty Estimation based on Retrieval Paths

The difficulty of a question is often tied to its ambiguity with respect to the collection being searched, which may be estimated using a clarity score [40] or a coherence score [38]. However, such scorings ignore the multi-hop structure of questions and the complex relationships that hold between documents retrieved for each hop. In our

approach, each multi-hop question is assigned a retrieval path, and the difficulty of the question can be measured by the cost of retrieving the context documents along the path. The cost here refers to the number of additional documents retrieved.

Let $P(c|q)$ denote the probability of reaching from question q a context document c that is needed to answer the question. The smaller the probability, the larger the number of documents the retriever has to retrieve before finding c. When this probability is one, there is enough evidence to reach the context without incurring additional costs. The expected number of documents to be retrieved, or the cost, may be denoted by $1/P(c|q)$. We use the terms contexts and supporting documents interchangeably in this paper.

Now consider a question q associated with a 2-hop *bridge* retrieval path and context documents c_1 and c_2, as shown in Figure 3.1a. The probability of retrieving both contexts can be written as

$$P_{ret} = P(c_1|q) \times P(c_2|q, c_1) \tag{3.1}$$

where $P(c_1|q)$ is the probability of reaching c_1 from q and $P(c_2|q, c_1)$ is the probability of reaching c_2 from q and c_1. Here c_1 denotes a context that is directly reachable from q but c_2 can be reached only after retrieving c_1. For a *comparison* retrieval path, both contexts can be retrieved independently, and the probability of retrieving both contexts can be expressed as

$$P_{ret} = P(c_1|q) \times P(c_2|q). \tag{3.2}$$

For a *mixed* retrieval path, the retriever has three options: (1) retrieve c_1 first and c_2 next, (2) retrieve c_2 first and c_1 next, and (3) retrieve both c_1 and c_2 independently. It is reasonable to assume that the retriever will take the path with the highest probability (or the least cost), i.e.

$$\begin{aligned} P_{ret} = \max\{&P(c_1|q) \times P(c_2|q), \\ &P(c_1|q) \times P(c_2|q, c_1), \\ &P(c_2|q) \times P(c_1|q, c_2)\}. \end{aligned} \tag{3.3}$$

It is possible that a given question does not follow any of the aforementioned retrieval paths, for example, when the question does not provide enough evidence to efficiently retrieve any of its contexts. These are rare cases though, examples of which were reported for the HotpotQA dataset in the previous section. We consider these questions difficult, with $P_{ret} \approx 0$.

Finally, estimating the difficulty of a question is hinged on estimating the model parameters, as addressed next.

3.4 Estimating the Model Parameters

Under a pre-retrieval setting, our probabilities can be estimated based on the question and maybe the corpus statistics. This means we may not have enough information about some of the hops (e.g., the 2nd-hop document in a bridge question).

3.4.1 Estimating the probabilities

Suppose the retrieval path of a question is known, and the goal is to estimate the probabilities of reaching the hops on the path[1]. A sparse retriever will use the terms of the question to find the context documents, but selecting those terms for each hop of a multi-hop question is not straightforward. Unlike a single-hop retriever that uses all question terms in the retrieval, a multi-hop retriever may use named entities that are mentioned and their relationships to guide the search [7]. On the same basis, named entities are good candidates for retrieving the supporting documents at each hop.

Sometimes the question has long phrases (e.g., the title of a song) that appear as a whole in context documents, and those phrases may not be detected as named entities. Yadegari *et al.* [53] study those phrases, referred to as frozen phrases, and show that identifying them can improve the retrieval models in open-domain QA. Thus frozen phrases may also be considered.

[1] In the next subsection, we discuss how retrieval paths can be predicted.

We utilize publicly available code for extracting named entities[2] and frozen phrases[3]. However, named entities or frozen phrases may not appear verbatim in the supporting documents, and this will be a problem in calculating the probabilities. For example, consider the question "Which singer is in the duo Sugarland, Jennifer Nettles or Roger Taylor?" from the HotpotQA dataset. In the supporting documents, "Roger Meddows Taylor" appears instead of "Roger Taylor." To deal with this problem, we extract unigrams, bigrams, and trigrams[4] from named entities. We only extract unigrams from frozen phrases because named entities are already extracted by the named entity extraction module, and the remaining terms in frozen phrases may not be consecutive. With this strategy, it is more likely that some n-grams will appear in supporting documents. Let NG_q denote all those n-grams of query q.

Now consider questions that follow a comparison retrieval path. A hypothesis is that such questions are expected to mention two or more entities (see the example given in Table 3.1) and each hop closely relates to one of those entities. Based on this hypothesis, a 2-hop retriever may extract two unique named entities of the question and retrieve the relevant documents of each named entity. Our probabilities are also estimated based on those unique named entities and frozen phrases. In particular, we select the two most specific n-grams $n_1, n_2 \in NG_q$ to represent the two contexts of q, and estimate $P(c_1|q) = P(c_1|n_1) = \frac{1}{N(n_1)}$ and $P(c_2|q) = P(c_2|n_2) = \frac{1}{N(n_2)}$, where $N(n) \neq 0$ denotes the number of documents that mention n-gram n.

For a bridge question, the probabilities may be estimated similarly, with the exception that only the first context c_1 can be reached from the question. In particular, the probability of reaching the first context may be estimated under a *Max* scheme, i.e.

$$P(c_1|q) = max_{t \in NG_q, N(t)>0} \frac{1}{N(t)}. \tag{3.4}$$

This is an optimistic estimation that assumes the most specific n-gram (i.e., the

[2]https://huggingface.co/Jean-Baptiste/roberta-large-ner-english
[3]https://github.com/Aashena/Frozen-Phrases
[4]These are consecutive terms forming bigrams and trigrams.

one with the highest probability) appears in the first supporting document. Since a pre-retrieval method does not have any additional information about the retrieved documents in the second hop, the probability of reaching the second hop in Equations 3.1 and 3.3 can be set to a constant (i.e., $P(c_2|q, c_1) = P_{hop_2}$).

3.4.2 Detecting the retrieval path of questions

To use our performance prediction models in Eq. 3.1 and 3.2, one must know the retrieval path of questions. Generally, detecting the retrieval path of a question without detailed information about the supporting documents is not easy. However, the structure of a question and the relationships between the entities mentioned often provide clues on the retrieval path the question may take. Based on those features, a classifier may be trained to detect the retrieval path type of a question.

3.5 Experimental Evaluation

3.5.1 Datasets

HotpotQA [7], one of the largest public benchmark for multi-hop QA, includes 113k Wikipedia-based question-answer pairs. The dataset is broken down to 90k train set, 7.4k dev set, 7.4k test-distractor where 2 gold paragraphs are mixed with 8 distractors (closed-domain) and 7.4k test-fullwiki where the relevant paragraphs include the first paragraph of all Wikipedia articles (open-domain). The train set is also broken down to 18k train-easy (mostly single-hop), 56.8k train-medium (multi-hop) and 15.7k train-hard (multi-hop hard) questions. The train-easy set is detected by labelling the turkers who tended to type single-hop questions. The train-medium class includes questions that could be answered with high confidence using a QA system built upon Clark and Gardner [54] with some of the SOTA techniques added [7]. Questions in the train and dev sets are also tagged as either bridge or comparison.
WikiPassageQA [55], is the largest non-factoid open-domain QA dataset including 4k questions created from 863 Wikipedia documents. The dataset consists of train,

dev and test sets including 3332, 417, and 416 questions. Each question can be answered with multiple passages from one long document.

WikiQA [56] is an open-domain dataset including 3k single-hop questions created from Bing query logs and it broken into train, dev and test sets including 2118, 296, and 633 questions. In WikiQA, each question can be answered with a Wikipedia document.

3.5.2 Evaluation Metrics

Correlation with the average precision

The quality of our query difficulty estimation may be gauged in terms of *the correlation* between our model performance estimates and the actual performance of the retrievers, as commonly done in ad-hoc and QA retrieval [11, 16, 44]. We report this metric in terms of Pearson's correlation (P-ρ), Spearman's correlation (S-ρ), and Kendall's correlation (K-τ). Significant test results at p-values 0.01 and 0.001 against the null hypothesis that the distributions are uncorrelated are also reported.

As a measure of the performance of a retriever on a question, we use the average precision in retrieving documents that are needed to answer the question. However, unlike ad-hoc retrieval where there is one list of retrieved documents, multi-hop QA retrieval involves retrieving supporting documents for multiple hops. To aggregate these results into a single list, we interleave the documents from different hops, with the first document from the first hop, followed by the first document from the second hop, etc. This strategy, which is also used in Xiong *et al.* [18], evenly combines the results for different hops and is expected to describe the behaviour of retrievers.

Pairwise difficulty estimation accuracy

Pairwise difficulty estimation, which determines which one of two questions is more challenging to evaluate, is a more modern performance metric utilized in recent post-retrieval studies [17]. This metric is more intuitive compared to correlation, which

can be challenging due to the the disparity between the distribution of the estimated scores and the distribution of the actual evaluation metric, such as average precision. We compare the difficulty of question pairs by utilizing the estimated scores obtained from pre-retrieval models. These scores indicate whether q_1 is more difficult than q_2. Furthermore, by considering the positions of supporting documents in the list of retrieved documents, we can say that q_1 is more challenging than q_2 if the number of retrieved documents to cover both supporting documents is greater for q_1. Hence, the accuracy of a model can be estimated based on the labels it assigns to question pairs and the actual labels of those pairs during evaluation.

Paragraph exact match and recall

Question difficulty classes may be defined and each question may be assigned to a class based on a performance prediction model. The actual performance of retrievers on those classes can show how good the prediction model has performed. Our question classes include *easy*, *hard* and *extra hard*. The actual performance of a QA system on each class is measured in terms of the fraction of questions whose supporting documents are all retrieved and the fraction of questions that at least one of their supporting documents is found. These performance measures are referred to in the literature [18, 25] as *Paragraph Exact Match (PEM)* and *Paragraph Recall (PR)* respectively.

Answer exact match and F1-score

For the end-to-end performance of a QA system, we use Exact Match (EM) and the F1 score, following the prior work on QA evaluation[6, 7, 18, 25]. The former measures if a returned answer exactly matches the ground truth and the latter combines the precision and recall in terms of the number of common words between a predicted answer and the ground truth.

QPP Baseline	Bridge		Comparison		Mixed	
	MDR	GoldEn	MDR	GoldEn	MDR	GoldEn
SCS [41]	49.93	50.86	51.34	54.06	53.71	54.81
maxSCQ [37]	53.94	54.03	**53.85**	56.83	54.67	55.26
avgSCQ [37]	52.46	52.05	51.67	57.15	55.46	56.01
maxIDF [40]	53.81	53.83	53.78	57.03	54.61	55.20
avgIDF [40]	52.44	51.99	51.52	56.99	55.41	55.91
maxIEF [43]	52.11	51.19	50.56	50.92	51.08	50.50
avgIEF [43]	50.23	50.21	50.37	50.07	50.21	50.15
multHP (ours)	**58.82**	**58.90**	52.50	**57.73**	**57.39**	**58.06**

Table 3.2: Pairwise difficulty estimation accuracy compared to pre-retrieval QPP baselines

Question Type	**QPP Baseline**	**MDR**			**GoldEn**		
		P-ρ	S-ρ	K-τ	P-ρ	S-ρ	K-τ
Bridge	maxSCQ [37]	0.1418	0.1263	0.0907	0.1477	0.1479	0.1081
	maxIDF [40]	0.1405	0.1259	0.0904	0.1431	0.1450	0.1061
	maxIEF [43]	0.0266	0.0304	0.0220	0.0164	0.0067	0.0050
	multHP (ours)	**0.2342**	**0.2480**	**0.1849**	**0.2858**	**0.3088**	**0.2369**
Comparison	maxSCQ [37]	**0.0866**	0.1051	0.0829	0.1794	0.1977	0.1548
	maxIDF [40]	0.0783	0.0941	0.0742	**0.1923**	0.2050	0.1604
	maxIEF [43]	-0.0195	-0.0076	-0.0055	0.0342	0.0070	0.0057
	multHP (ours)	0.0460	**0.1139**	**0.0894**	0.1130	**0.2597**	**0.2024**

Table 3.3: Correlation between the difficulty prediction of pre-retrieval models and the actual retriever performance, in terms of average precision, of MDR and GoldEn on HotpotQA (results are statistically significance at p-value < 0.001)

3.5.3 Code

Our implementation includes the following components.

- Multi-hop question analysis, encompassing all functions related to generating retrieval paths and categorizing them based on their question types.
- Pre-retrieval performance predictor, comprising functions that process each question by calculating the probability of its successful retrieval.
- Post-retrieval training component, including all necessary classes and functions for training a neural network.

We make our code publicly available on GitHub [5].

3.5.4 Retrieval Models and QPP Baselines

In the absence of prior research on QPP in multi-hop QA settings, we use the following pre-retrieval methods commonly used in ad-hoc retrieval tasks as baselines for comparison: (1) Inverse Document Frequency (IDF) [40, 57], which predicts query performance by considering the specificity of the question terms, with higher values indicating an easier question to answer; (2) Simplified Clarity Score (SCS) [38], which estimates query performance by taking into account both the query length and the specificity by computing the divergence between the simplified query language model and the collection language model; (3) Collection Query Similarity (SCQ) [37], which predicts the performance based on the similarity between the query and the collection documents; (4) Inverse Edge Frequency (IEF) [43], which estimates question specificity within an embedding space, taking into account the number of close neighbors associated with each term. These predictors are aggregated over question terms using max and average functions, resulting in maxIEF, avgIEF, maxIDF, avgIDF, maxSCQ, avgSCQ, and SCS.

[5] https://github.com/MhmDSmdi/performance-prediction-for-multihop-QA.git

With the performance of open-domain QA systems typically bounded by the retrievers [28, 58], query difficulty may be quantified in terms of the performance of the retrievers. We utilize two multi-hop and one single-hop QA retrievers in our evaluation. **GoldEn** [6] is a sparse model built on top of DrQA [2]. We used the authors public code [6] and instructions to train the retriever on the HotpotQA dataset. **MDR** [18] is a dense model that retrieves the relevant documents based on the inner product score between a question and documents embedding vectors. In our work, we used the public code [7] and retriever checkpoint provided by the authors. We utilized the same hyperparameters as reported in the original study, with the exception of the number of retrieved documents, which we set to 5 per hop. We also used DrQA [2] as a single-hop retriever.

3.5.5 Compared to Pre-retrieval QPP Baselines

Pairwise Difficulty Comparison

Table 3.2 shows the accuracy of correctly predicting the pairwise difficulty of questions. The results are categorized based on the question types introduced in Section 3.3. There are 5,918 Bridge questions (17,508,403 question pairs), and our metric outperforms all of the baselines with over 4.87% improvement (corresponding to more than 852k question pairs) in both retrievers. In Comparison questions, our multHP slightly improves the accuracy, while our estimate for MDR's performance does not exceed the baselines. Based on our analysis, MDR performs quite well in retrieving both supportive documents, and our predicted scores for questions may not accurately estimate the retriever's performance. In addition to Bridge and Comparison questions, we leverage Equation 3.3 to demonstrate the effectiveness of our proposed formulations even when the question type is not specified. In the Mixed type, our multHP outperformed all of the baselines with improvements of 1.93% and 2.05% in terms of accuracy for the MDR and GoldEn models, respectively.

[6] https://github.com/qipeng/golden-retriever
[7] https://github.com/facebookresearch/multihop_dense_retrieval

Multi-hop Pointwise Correlation

Table 3.3 shows the correlation between the predicted query performance scores and the actual performance of the retrievers. The results are broken down to question types, in terms of comparison or bridge, since the calculations are slightly different and the retrieval path of a question can be easily predicted. Our predicted scores showed a significant correlation with the actual performance of the both retrievers for bridge questions. However, the Pearson correlation for comparison questions was lower for MDR [18] simply because MDR performs quite well in comparison questions due to the fact that both entities are mentioned in the questions. This results in both supporting documents being placed in the low ranks of the retrieved results for either one hop or both, and an estimation solely based on a random document selection model may not correlate with the actual average precision. Ultimately, in comparison questions, neither our estimated scores nor those of our baseline methods show strong correlation with the actual performance of the MDR retriever. Our analysis of five retrieved documents showed that the average ranks of the two supporting documents that appeared in the 1st hop were 1.71 and 1.72 for MDR, and those ranks were 1.63 and 4.52 for GoldEn. The difference in the average ranking of the second document in the hop 1 indicates that MDR performs exceptionally well on these questions. Besides, considering the disparity in scale between our estimated scores and the average precision, particularly for comparison questions, Spearman and Kendall coefficients may serve as better indicators of performance due to their rank-based nature.

Comparing the results of syntactic metrics such as multHP and SCQ with the Inverse Edge Frequency (IEF) semantic metric [43] shows that the mere semantics of terms may not be a good estimator of specificity. One of the main challenges of embedding-based metrics is out-of-vocabulary terms, wwhich include many named entities. Most multi-hop questions are factoid questions that include named entities, and these named entities may not map well to the embedding space. Furthermore,

QPP Model	WikiPassageQA			WikiQA		
	P-ρ	S-ρ	K-τ	P-ρ	S-ρ	K-τ
maxSCQ [37]	0.1107	0.1363 $*$	0.0980 $*$	-0.0015	-0.0230	-0.0168
maxIDF [40]	0.1096	0.1287 $*$	0.0929 $*$	-0.0402	-0.0432	-0.0321
maxIEF [43]	0.1164	0.0823	0.0621	-0.0816	-0.0801	-0.0602
multHP (ours)	**0.1889** †	**0.2507** †	**0.1833** †	**0.1030** $*$	**0.1280** $*$	**0.0981** $*$

Table 3.4: Correlation between the difficulty prediction of QPP models and the actual retriever performance, in terms of average precision, of DrQA on WikiPassageQA and WikiQA datasets ($*$ and † denote the correlations with p-value less than 0.01 and 0.001 respectively)

IEF merely leverages the embedding space and ignores the corpus statistics, which are good predictors for retrieval performance prediction.

The same results was observed using GoldEn [6], a sparse retriever. As shown in Table 3.3, the correlation between our estimated difficulty score and the actual performance of the GoldEn retriever was much more pronounced, compared to the baselines, for bridge questions. However, for comparison questions, our approach did not perform better than the baselines. Our error analysis shows that the GoldEn retriever could not find the supporting documents for questions that we estimated as easy questions with a fairly high score. For instance, consider the question "Which pizza shop opened first, Toppers Pizza or America's Incredible Pizza Company?". Based on our probabilities, $P(c_1|Toppers\ Pizza) = 0.33$ and $P(c_2|America's\ Incredible) = 1$, but the GoldEn retriever failed to retrieve the supporting document "America's Incredible Pizza Company". In this particular case, GoldEn query generator emitted two queries, "pizza shop opened first, Toppers Pizza" and "Pizza," to retrieve the supporting documents. These queries failed to retrieve the second entity. Also in some cases, the GoldEn retriever could extract supporting documents for questions that we estimated as difficult to retrieve. Consider the comparison question "Hayden is a singer-songwriter from Canada, but where does Buck-Tick hail

from?". Our approach correctly extracted the two entities, computed the probabilities $P(c_1|Buck-Tick) = 0.0164$ and $P(c_2|Hayden) = 0.0011$, and estimated the $P_{ret} \approx 1.8 \times 10^{-5}$. However, Golden retriever extracted both supporting documents easily at top of the result set using the two queries "Hayden is a singer-songwriter from Canada" and "Buck-Tick". In this case, we underestimated the probability of finding a relevant document by considering only one entity per query while the retriever leveraged all information in the query, such as "singer-songwriter" and "Canada" in this example.

Single-hop Pointwise Correlation

To show how our model performs on single-hop questions, we evaluated our approach on the test sets of WikiPassageQA [55] and WikiQA [56], two open domain QA datasets using DrQA retriever, following the setting as explained in Section 3.5.5. We used the setting of our bridge questions where the estimation is done based on the first hop. From Table 3.4, we can observe that while the correlations are not very strong, our estimates show a stronger correlation with the actual performance of the system, compared to the baselines, and the results are statistically significant. This is mostly because of our term selection strategies and the use of named entities and frozen phrases for our performance prediction.

3.5.6 Performance Across Difficulty Classes

Retriever performance results

In another experiment, we wanted to illustrate the performance drop in different difficulty classes. To this aim, we categorized the questions of HotpotQA's dev set into different difficulty classes and evaluated the performance of the retrievers on those class. A similar categorization is done in the work of Mothe *et al.* [59]. To set threshold scores for categorizing questions into different difficulty classes, we used the percentile-based strategy [59]. For bridge questions, we set $P_{hop_2} = 0.125$ to

Class	Model									
	DrQA (k=100)		GoldEn (k=5)				MDR(k=1)			
	PEM	PR	PEM	PR	EM_{joint}	$F1_{joint}$	PEM	PR	EM_{joint}	$F1_{joint}$
Easy	33.07	84.70	67.01	95.78	20.68	43.87	71.50	89.97	36.86	59.66
Hard	27.49	77.27	60.31	88.63	20.25	39.31	63.94	83.59	35.62	55.68
Ex. Hard	22.69	71.45	51.61	80.09	16.95	35.27	56.71	73.65	31.49	48.06

Table 3.5: Retrieval performance (in terms of PEM and PR) and end-to-end performance (in terms of EM and F1) of three models across three difficulty classes, predicted using the *Max* scheme, showcasing performance degradation for more challenging questions

calibrate the score difference between bridge and comparison questions. We named these difficulty classes extra hard (1st quartile), hard (2nd quartile), easy (3rd and 4th quartiles). We merged 3rd and 4th quartiles into one class because we observed that there was no noticeable difficulty gap between these two sets.

Table 3.5 shows the performance of three retrievers, with their default settings, on different question classes in terms of PEM and PR under *Max* scheme. We can see more than 10%, 16%, and 15% performance drop between easy and extra hard classes of DrQA, GoldEn, and MDR respectively in terms of PEM. Since both supporting documents are required to answer a multi-hop question, by comparing the results of the retrievers on question classes, we can conclude that the number of questions that cannot be answered in hard and extra hard classes are considerably larger than the number of such questions in the easy class.

End-to-end performance results

Our analysis so far shows that our QPP model is quite effective in predicting the performance of multi-hop QA retrievers. In this section, we want to evaluate how this translates to predicting the end-to-end performance of our QA systems on different difficulty classes. To evaluate the performance in both document retrieval and answer extraction phases, we used the standard answer and supporting facts given in the

dataset, and calculated Exact Match (EM) and F1 score on questions with different classes of difficulty. Tables 3.5 shows the results of GoldEn and MDR models, in an end-to-end setting, in terms of joint EM and F1 score for answer extraction and supporting sentences prediction. We can observe a declining performance between easy and extra hard classes. It should be noted that our difficulty score estimation is merely based on the retrieval paths, and some questions that are deemed difficult to answer, using the QPP model, may not pose much challenge to the QA system in finding the supporting documents.

3.5.7 Two Use Cases

Query performance prediction may serve multiple purposes, including providing valuable feedback to users, enhancing content quality, and optimizing resource allocation for retrieval processes. In this study, our focus is twofold: initially, we employ performance prediction to annotate the HotpotQA test set, a task particularly challenging due to the absence of labels. Subsequently, we introduce an innovative adaptive retriever that boasts both reduced runtime and improved performance when compared to static retrievers. Importantly, all the applications highlighted above find their utility within the post-retrieval paradigm, showcasing the adaptable nature of performance prediction across various scenarios.

Dataset annotation

Annotating datasets such as HotpotQA can help in evaluating the models, and our performance prediction can be used in the annotation process. In particular, detecting question difficulty prior to the retrieval phase may allow, for example, the models to be evaluated on more difficult subsets. One may also choose simpler models for easy questions and more complex models to answer difficult questions. The train and dev sets in HotpotQA have annotations for the question difficulty and the type of retrieval path, but those annotations are not given for the test set. The test set of HotpotQA

does not have question types (e.g., comparison or bridge). For our experiments on the test set, we trained a type detection model, as discussed in Section 3.4.2. To train the model, we used 80% of the HotpotQA train set for training and the other 20% as the development set because the test set does not have retrieval path types of questions. For training, we utilized the [CLS] embedding created by RoBERTa$_{base}$ followed by a hidden layer with size 128 to extract high-level features. Finally, the last layer is a softmax layer to predict the probability of each type (i.e., bridge or comparison). We used *relu* activation function and set learning rate $5e-5$, batch size 64, and epoch 3. We evaluated this model using the actual dev set of HotpotQA, and its performance was 99.63%, 94.64%, and 97.07% in terms of precision, recall, and F1-score respectively.

Adaptive retrievers

Improving the performance of models is possible by detecting the difficulty levels of questions and maybe allocating additional resources for more difficult questions. Figure 3.2 shows how retrieving additional documents improves the end-to-end performance of MDR model. We can observe that increasing the number of retrieved documents have most positive impact on the extra hard set compared to the easy set.

To take another step forward, we built an adaptive retriever that retrieved more passages for questions that were detected to be difficult. In particular, the retriever fed ck documents to the reader where c was set based on the difficulty class and k varied from 1 to 20. Figure 3.3 shows the results with c set to 1 for easy, 4 for hard and 4 and 5 for extra hard. Given a limited running time budget, a higher performance, in terms of F1-score, is achieved using the adaptive retriever compared to a constant one that retrieves the same number of documents for all questions irrespective of their difficulty classes. This means the adaptive retriever saves time on easy questions by retrieving fewer documents and spends the saved time on retrieving more documents for difficult questions, which improve the performance of the model. We can also

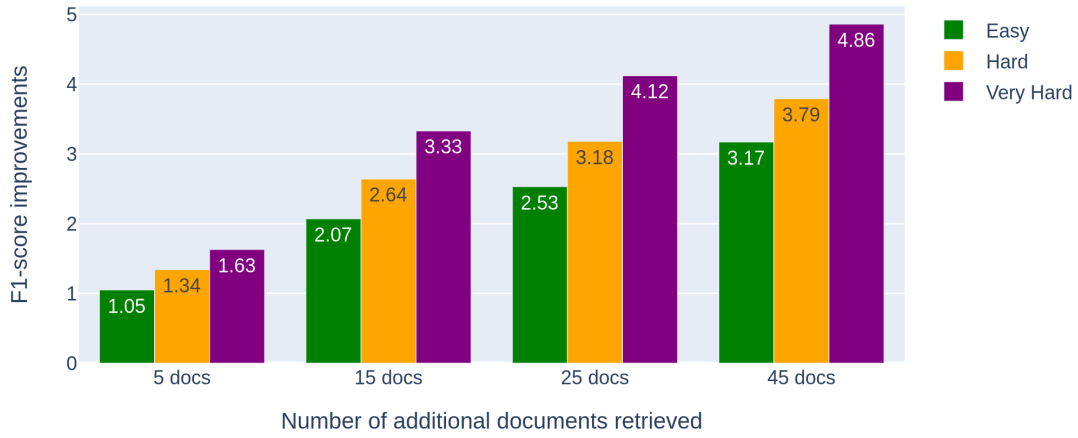


Figure 3.2: Improvements in the end-to-end performance of MDR [18], in terms of F1-score, across different difficulty classes and varying the number of additional document retrieved, showing larger improvements for more difficult classes

see a sharp increase in the performance of the constant retriever as k is increased from 1 to 3, indicating that all questions including easy ones benefit from larger k values. However, for $k \geq 5$, easy questions do not benefit as much as hard questions. One possible explanation is that top-ranking documents tend to contain relevant information for easier questions more often than for harder ones.

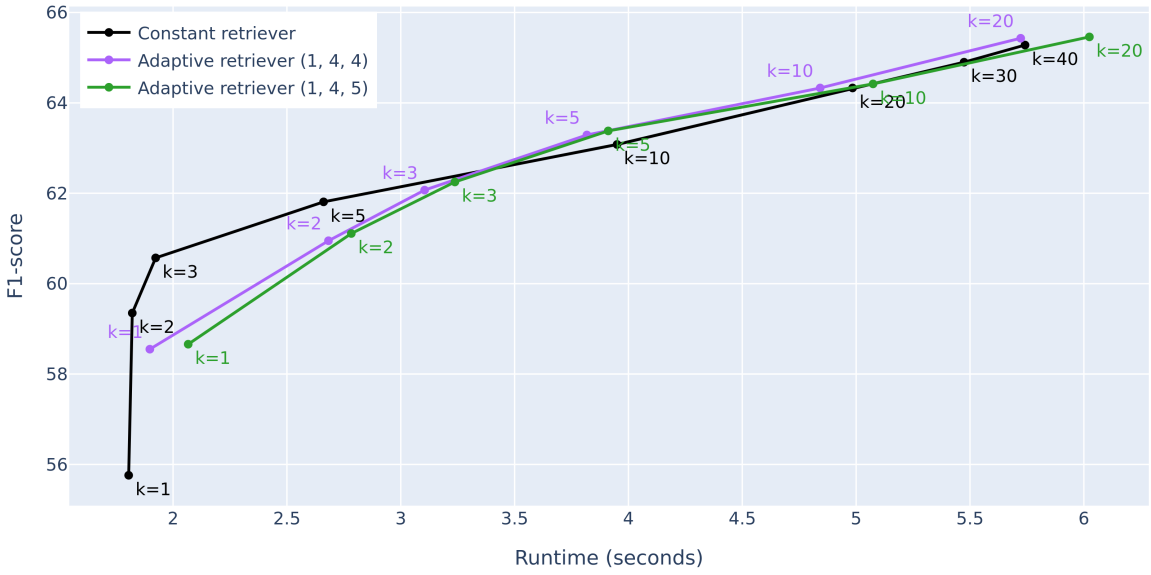


Figure 3.3: Performance, in terms of F1-score, of MDR [18] with the adaptive retriever compared to a constant retriever while k varied, showing that the adaptive retriever achieves a higher performance under the same budget

Chapter 4

Post-retrieval QPP

4.1 Motivation

In the previous chapter, we defined the task of performance prediction from the pre-retrieval viewpoint. However, we believe that relying only on syntactic relatedness between question and documents, as introduced in Section 3.1, is not sufficient for estimating performance precisely. Also, in multi-hop QA, the primary factor that determines the difficulty of bridge questions is missing information, which can be found within the retrieved documents. As we ignored this factor in our proposed formulation in Section 3.4, we aim to tackle these challenges and conduct further research on potential improvements. Our goal is to develop a post-retrieval performance prediction method for multi-hop QA.

Recent studies on performance prediction in ad-hoc retrieval [16, 17] have demonstrated that considering semantic interactions between the query and the top retrieved documents can enhance the accuracy of performance estimation for the model. They formulate the performance prediction as a supervised-task and leverage semantic features to estimate a real-number as the model's performance. Similar to this line of work, we aim to look at the problem from a post-retrieval viewpoint and propose a model that leverages both semantic features and retrieval scores.

Unlike our pre-retrieval method (Chapter 3), this approach would not be constrained by a fixed number of hops. Taking into account the iterative nature of

multi-hop QA, multi-hop retrievers perform a retrieval step and reasoning in each iteration to extract missing information and update the question. Thus, we consider the performance prediction for a 2-hop question, which involves one full cycle of retrieval and reasoning, as the fundamental building block of our method. Building upon the retrieval paths introduced in Chapter 3, we define two main steps, bridge-step and comparison-step. Using multiple bridge and comparisons steps, we can construct complex n-hop questions for $n > 2$, following the approach of Ho *et al.* [8]. Therefore, our proposed method handles performance prediction for n-hop questions by utilizing a step prediction model. However, our evaluation is limited to 2-hop questions, closely following our pre-retrieval work in Chapter 3. Further study and evaluation of the work for n-hop questions where $n > 2$ is left to the future work.

4.2 Methodology

For a given 2-hop question q, our objective is to predict the performance of the retriever by considering the results of the top-k supporting documents D_q obtained from a fixed retrieval system $\mathcal{M}$. This prediction involves estimating a real value that reflects the retriever's effectiveness in handling the multi-hop question based on the information retrieved from the top documents.

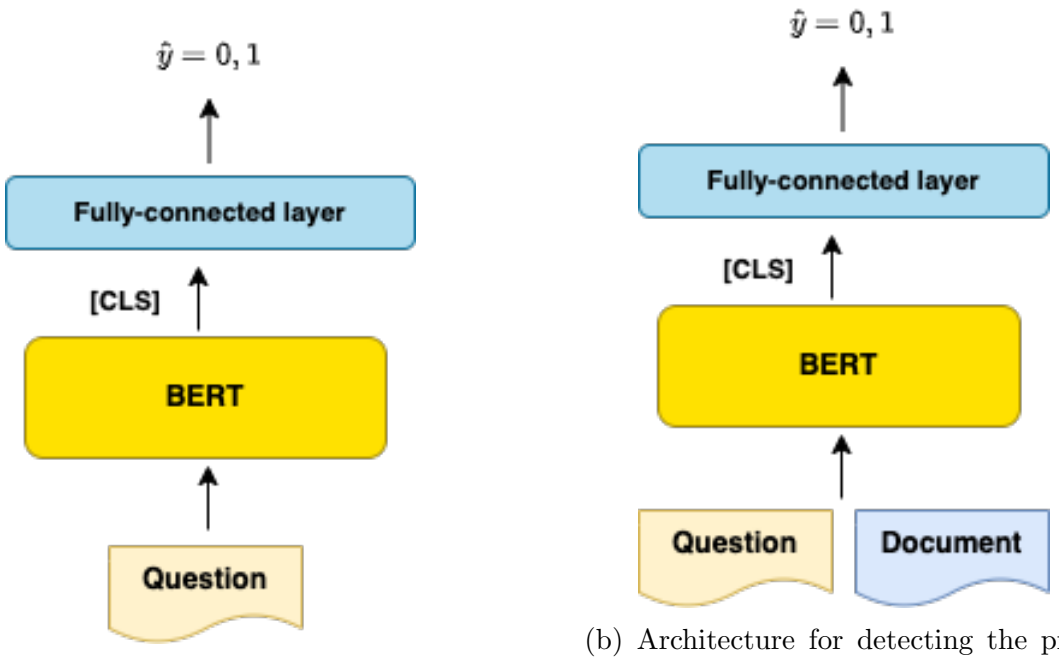
In the literature of supervised performance prediction for ad-hoc retrieval, feature extraction plays a pivotal role in achieving an accurate model, with many proposed methods aiming to capture the semantic interactions [17, 48]. However, the main distinction between ad-hoc and multi-hop retrieval lies in the presence of multiple sets of documents rather than a single list, necessitating the capture of semantic interactions between them. To address this, we propose a unified model that leverages various types of features commonly used in the literature of performance prediction. The following subsections delve into the details of extracted features and the architecture our unified model.

4.2.1 Feature Extraction

We highlight the significance of feature extraction in supervised post-retrieval methods for question difficulty estimation, building upon previous studies [15, 17, 43]. We present our approach, which focuses on extracting three sets of features to capture various aspects of the multi-hop question answering process: (1) Specificity-based, (2) Score-based, and (3) Question-dependent features.

Specificity-based

These features capture the relationship between a given question and the retrieved documents at each hop. By considering the specificity of the information retrieved and its relevance to the question, we gain insights into the alignment of the retrieved documents with the question's information need. Specificity-based features help us measure the degree of connection and coherence between the question and the supporting documents. Motivated by previous studies in performance prediction [15], we employ the BERT [23] encoder to extract semantic features from the given question and the documents retrieved at different hops. To capture the interactions between the question and each retrieved document set, we adopt pair-encoding using BERT. This approach enables us to generate a joint representation that considers the specific context provided by each document, and we can effectively capture the semantic relationships between them. It is important to note that in the multi-hop reasoning process, retrievers often update the question at each hop based on the newly retrieved information. In our framework, we account for this dynamic nature by using the updated question representation, which is obtained by simply concatenating the top-1 document with the question. A similar approach is taken by Xiong *et al.* [18]. This ensures that our model considers the evolving question context, enhancing its ability to estimate question difficulty accurately.



(a) Type detection architecture

(b) Architecture for detecting the presence of bridge entities

Figure 4.1: Question-dependent feature extractor models

Score-based

Previous studies [14, 15] have demonstrated that considering the distribution of retrieval scores associated with the retrieved documents can serve as a reliable estimator of retrieval performance. By analyzing the scores, we can assess the relevance and importance of the retrieved documents for each hop, so they provide a quantitative measure of the strength of evidence and the confidence level associated with the retrieved information. In our method, we used the retrieval scores corresponding to retrieved documents in 1st and 2nd Hop called Hop1 and Hop2 scores in Figure 4.2 as additional features and concatenate them to the extracted semantic features to give the model more information about the retrieved documents.

Question-dependent

Additionally, we integrate features that are specifically related to the multi-hop question, the question type, and the probability of finding a bridge entity for a bridge

question. The former helps the model differentiate between score prediction for bridge and comparison questions, while the latter addresses the distinct challenges and demands of multi-hop reasoning. The HotpotQA dataset includes question types and bridge entities in both the train and dev sets. We train a dedicated model using the HotpotQA training set to predict whether the given question is bridge or comparison (Figure 4.1a). We employ BERT to extract contextualized representations of the question, particularly the [CLS] token and feed it into a fully-connected layer with sigmoid activation. Similarly, for estimating whether a bridge entity appears in the given document, we develop another model with the same architecture (Figure 4.1b). To capture the semantic interaction between the question and its corresponding top-1 retrieved results, we utilize BERT pair-encoding and feed the representation vector into a fully-connected layer with sigmoid activation to predict the presence of the bridge entity in the document.

By combining the three sets of extracted features, we create a comprehensive feature representation for each question. This representation captures the specific aspects of the multi-hop question answering task, including the relationship between the question and the retrieved documents, the distribution of retrieval scores, and the task-specific characteristics. These features serve as valuable inputs for the estimation of probabilities required for bridge reasoning and comparison steps.

4.2.2 Unified Model Architecture

The architecture of our unified model is illustrated in Figure 4.2. For extracting semantic interaction between the question and the documents retrieved for each hop, we utilize the pooling-output (i.e., $\phi_{CLS}^{BERT} \subset R^d$) of BERT, allowing us to obtain a comprehensive representation that incorporates the overall meaning and contextual information. Considering the iterative nature of multi-hop retrieval, the first representation vector is fed into a fully-connected layer, i.e. $\phi_{fc}(\phi_{CLS}^{BERT}(q, d_k^{hop1}))$, where $\phi_{fc} : R^d \rightarrow R^{d_{fc}}$, q is the original question, d_k^{hop1} is k-th document in the 1st hop. Pro-

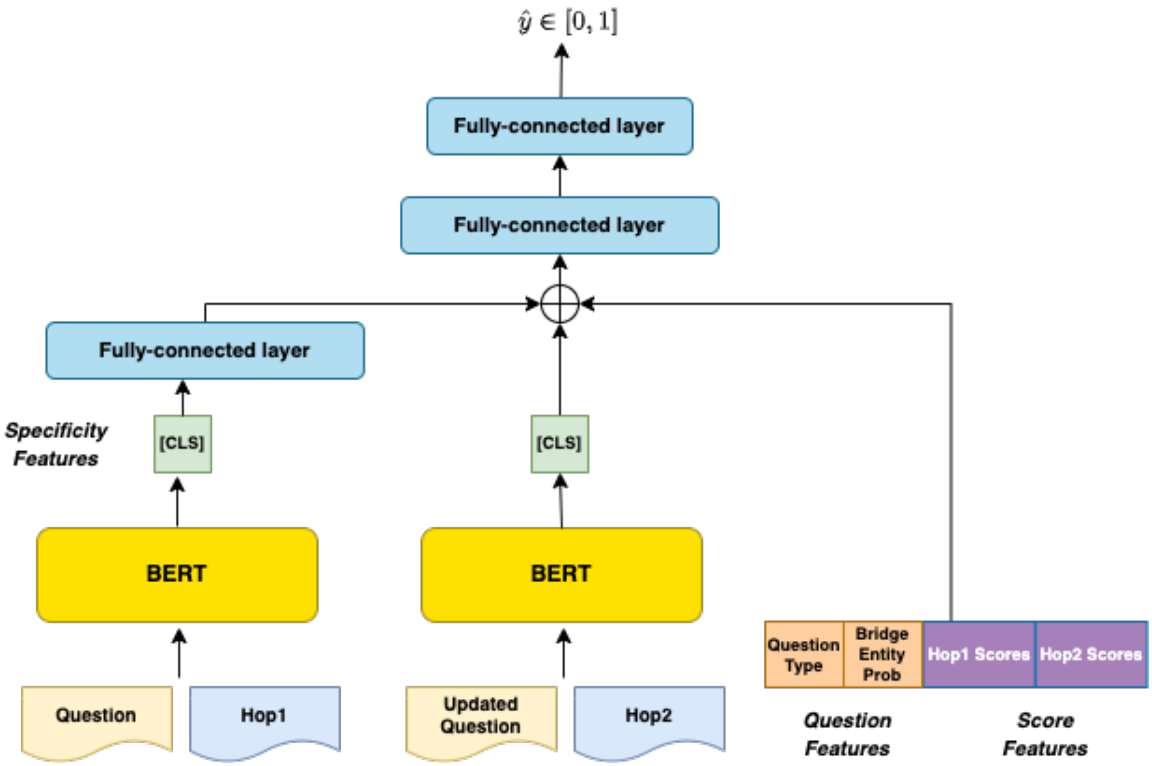


Figure 4.2: The architecture od our Unified Model

jecting the representation into a different feature space can potentially improve the model's ability to capture relevant patterns and introduce non-linearity, thereby enhancing its ability to handle added complexity. However, we refrained from adding an additional fully-connected layer for the question and hop2 representation to prevent overcomplicating the model and potential overfitting.

Secondly, we update the question by concatenating the top-1 document from the 1st hop and feed it into BERT to extract contextualized representations. Finally, we concatenate all the extracted specificity features, question-dependent features, and retrieval scores to form a vector, which is then passed into the classification layer. The classification layer is a fully-connected layer with sigmoid activation, constraining the output between 0 and 1.

To train the unified model, we mix all question types together and give the model more freedom to make decisions about learning how to extract features based on the

question types. Similar to the prior works [43], we train the model by minimizing the binary cross entropy of the output of the sigmoid function and the ground truth label.

4.3 Experimental Evaluation

4.3.1 Datasets

The HotpotQA dataset serves as the foundation for our experiments. In the full-wiki setting, it encompasses a large collection of 2-hop questions that necessitate multiple retrieval steps and reasoning to achieve accurate answers. More detailed information about HotpotQA is provided in Section 3.5.2.

4.3.2 Evaluation Metrics

To assess the performance of our supervised QPP method, we employ the evaluation metrics described in Section 3.5.2. In addition to measuring the correlation with average precision, we also evaluate the performance of our QPP method by correlating it with the number of documents required to complete the task. We believe that easy questions may necessitate fewer document retrievals compared to difficult questions. Given that this number corresponds to the final index of the supporting documents (considering both hops), we referred to it as reciprocal rank (RR) in our study. This metric primarily focuses on the total number of documents needed to gather all the necessary information for answering the question accurately.

4.3.3 Settings

We used the BERT pre-trained model [23] with 12 layers and attention heads with $d = 768$ dimensions for extracting contextualized features from the input text. For all of our experiments, we trained all models for 3 epochs and set the batch size to 8, learning rate to $2e-5$, maximum length to 256, and hidden-layers size to 128. To prevent overfitting of our model on the training set, we add dropout with $p = 0.25$

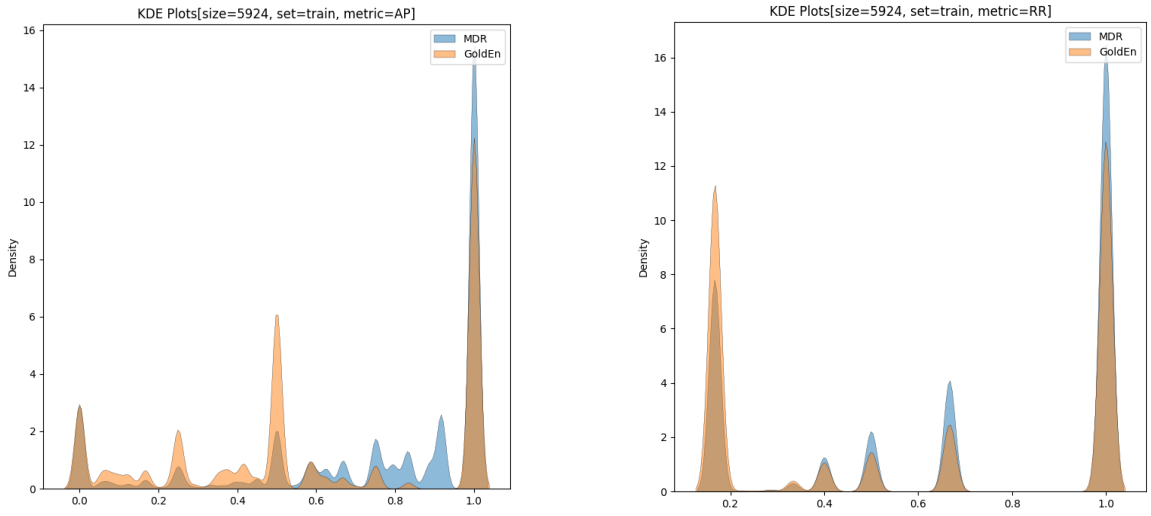
Retriever	QPP Baseline	AP			RR		
		P	S	K	P	S	K
GoldEn [6]	BERT-QPP [43]	0.7878	0.7843	0.6210	<u>0.6855</u>	**0.6968**	<u>0.5522</u>
	Unified Model	**0.8198**	**0.8266**	**0.6535**	**0.6993**	<u>0.6946</u>	**0.553**
	– Bridge Prob.	<u>0.8078</u>	<u>0.8139</u>	<u>0.6438</u>	0.6853	0.6909	0.5449
	– Doc. Semantic	0.4301	0.4265	0.3115	0.3869	0.4022	0.3118
MDR [18]	BERT-QPP [43]	0.7809	0.6752	0.5196	0.6081	<u>0.5824</u>	<u>0.4519</u>
	Unified Model	**0.8074**	**0.6927**	**0.5328**	**0.629**	**0.5936**	**0.4576**
	– Bridge Prob.	<u>0.7875</u>	<u>0.682</u>	<u>0.5242</u>	<u>0.614</u>	0.5791	0.4486
	– Doc. Semantic	0.4924	0.4570	0.3423	0.3905	0.3980	0.3058

Table 4.1: Performance of our proposed method compared to previous post-retrieval approaches using the GoldEn retriever. **Bold** and <u>underline</u> indicate the 1st and 2nd best performance, respectively.

to the hidden layers. For each hop, we use the top-1 document due to transformer input length limitations. Additionally, our aim is to minimize the retrieval time during inference. Retrieving more documents for post-retrieval performance prediction would essentially be akin to performing retrieval, and it would diminish the effectiveness of performance prediction. Moreover, BERT-QPP only supports **one** document, while for a 2-hop question, we have two supporting documents. To ensure a fair comparison, we concatenate both documents as one context for their model.

4.3.4 Results & Discussions

Our main results for evaluating our post-retrieval method reported in Table 4.1. We used GoldEn and MDR as fixed retrievers and trained our unified model using their retrieval results, considering both average precision and reciprocal rank. Observing the correlation coefficients for average precision, we found that our Unified Model consistently outperformed the previous state-of-the-art model in ad-hoc retrieval on both retrievers. However, the results in terms of reciprocal rank were not as stable.



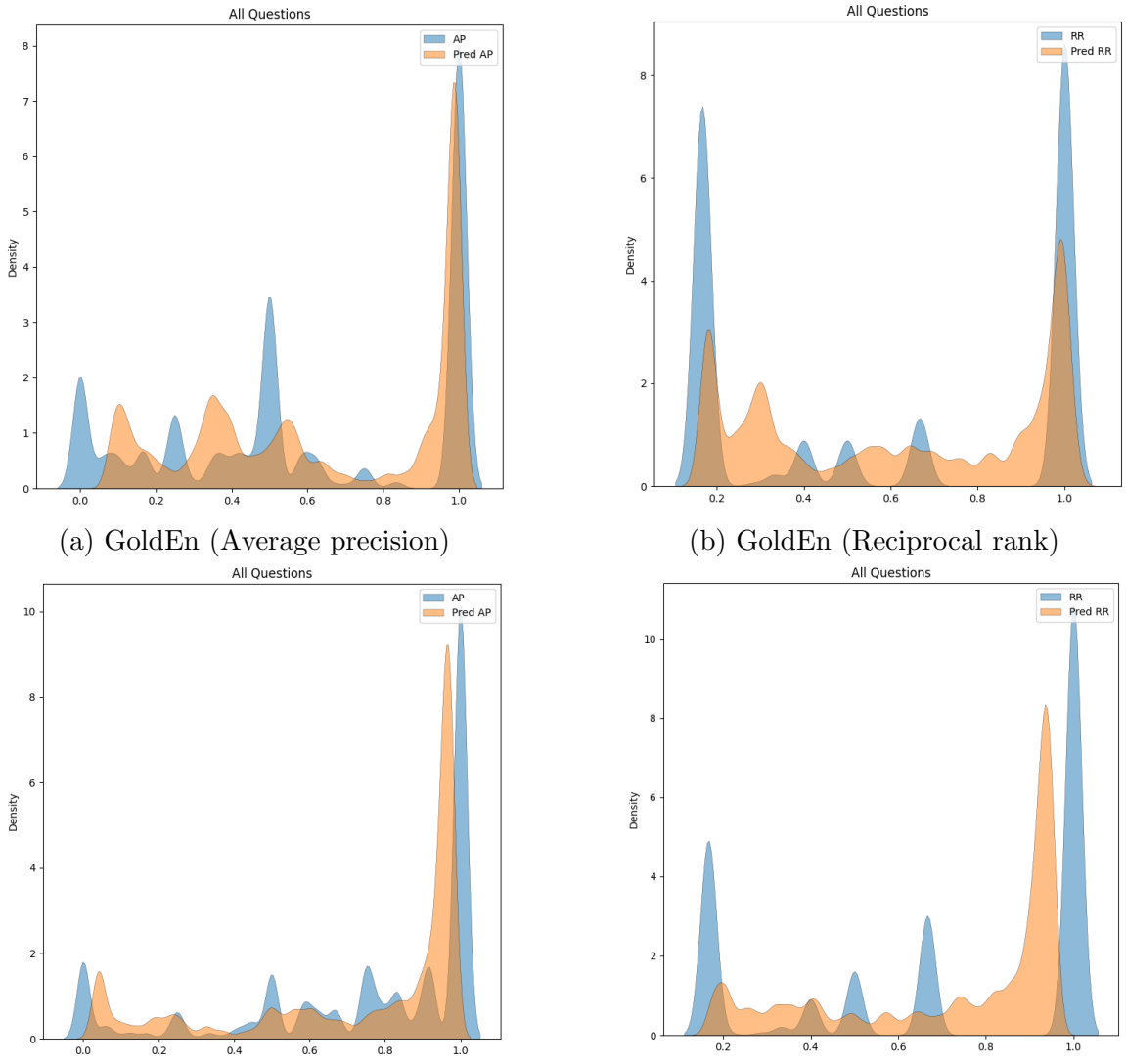
(a) Average precision (b) Reciprocal rank

Figure 4.3: Kernel Density Estimation (KDE) plots of actual scores for GoldEn and MDR

The Unified Model demonstrated superiority in Pearson and Kendall coefficients, while in Spearman correlation, it matched the performance of BERT-QPP.

Since bridge probability is a crucial feature specifically related to multi-hop QA, we conducted an evaluation to measure its effectiveness. Removing the bridge probability as an additional feature resulted in a decrease in performance across all correlation coefficients and evaluation metrics. This finding indicates that the bridge probability feature significantly assists the model in making better predictions. Moreover, when we remove all features related to the retrieved documents, which includes both semantic features and retrieval scores, there is a drastic decrease in performance for all correlation coefficients. Consequently, the model's performance becomes equivalent to that of a pre-retrieval model trained using supervised learning. This observation underscores the critical importance of document features for accurate performance estimation in the post-retrieval paradigm.

In addition to unexpected fluctuations in the correlation values for Reciprocal Rank (RR), we observe a performance gap between Average Precision (AP) and reciprocal rank. This suggests that both Unified Model and BERT-QPP may not excel at



(a) GoldEn (Average precision)

(b) GoldEn (Reciprocal rank)

(c) MDR (Average precision)

(d) MDR (Reciprocal rank)

Figure 4.4: Kernel Density Estimation (KDE) plots of predicted scores using the Unified Model and actual scores in different settings

predicting RR performance as effectively as they do for AP. Our analysis has led to some hypotheses to explain this phenomenon.

Figure 4.3 illustrate the Kernel Density Estimation (KDE) of actual score distribution. We can observe that the average precision score distribution differs significantly from the reciprocal rank distribution for both retrievers. While both scores are imbalanced, the reciprocal rank is more imbalanced as it depends on the number of retrieved documents and there are some datapoints for which no sample exists in

the dataset. For instance, no question requires the retriever to retrieve 8 documents, whereas there are several questions that only require 2 or more than 10 documents to obtain all the necessary information. This imbalance may have an adverse effect on training our model, causing the model to predict a value for which there is no corresponding datapoint in the dataset (orange areas that do not have any overlap with blue ones). Moreover, this issue with the actual score distribution causes the model not to train well, as evident from the correlation scores reported in Figure 4.4.

Chapter 5

Conclusions & Future Work

5.1 Conclusions

The main goal of this project was introducing the task of query performance prediction for multi-hop questions. In Chapter 1, we covered the main concepts of MHQA and QPP and discussed about our motivation for applying QPP in MHQA. In Chapter 2, we reviewed recent studies with a focus on applying QPP to ad-hoc retrieval and question answering, different methods proposed for multi-hop retrieval, and different QPP approaches that were categorized into pre-retrieval and post-retrival paradigms.

To take the first step of defining the task of performance prediction in multi-hop retrieval, we presented a pre-retrieval method to estimate a difficulty score of a multi-hop question based on the clues in the question (Chapter 3). We analyzed multi-hop questions in the HotpotQA dataset and proposed retrieval paths based on overlapping terms between the question and its supporting documents. Our experimental evaluation showed significant correlations between the performance of the retrievers used in our evaluation and our estimated difficulty scores, and those correlations were much higher than those obtained by our QPP baselines from the literature. The same trend was observed for the end-to-end models with the performance considerably dropped for the questions that were deemed difficult by our model. Determining the difficulty of a multi-hop question using a pre-retrieval method can assist the retrievers to have a better chance of retrieving all required documents to answer the question.

In the QPP literature, it has been shown that post-retrieval models are more accurate than pre-retrieval methods since they have more information about the retrieval process. To take another step further, we presented a generalized post-retrieval framework specifically designed for multi-hop QA. In Chapter 4, our generalized framework for any-hop retrieval is reviewed and we introduced distinct models for bridge and comparison questions. Based on our experimental results, there is strong correlation between our estimated performance prediction and the retrievers actual performance.

5.2 Future Work

As a potential avenue for future research, our models can be further enhanced by exploring more refined retrieval path types and fine-tuning the corresponding parameter settings. By considering a wider range of path types and optimizing the associated parameters, we can improve the accuracy and effectiveness of our QPP framework in estimating question difficulty and predicting retrieval performance.

Additionally, an interesting direction for future work involves analyzing the impact of our difficulty score estimation on downstream tasks. By evaluating the performance of downstream tasks, such as answer generation or passage ranking, using our estimated difficulty scores, we can gain insights into the relationship between question difficulty and the performance of subsequent tasks. This analysis can contribute to the development of more efficient and effective systems for multi-hop question answering.

Furthermore, it is worth exploring the connection between our approach and predicting optimal parameter settings for retrievers. By investigating how our difficulty score estimation correlates with the optimal number of retrieved documents, we can provide valuable insights into the parameter settings that yield the best retrieval performance. This research direction can contribute to improving the efficiency and effectiveness of retrievers in multi-hop question answering scenarios.

Bibliography

[1] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, *et al.*, "Okapi at trec-3," *Nist Special Publication Sp*, vol. 109, 1995.

[2] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to answer open-domain questions," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1870–1879.

[3] S. Wang *et al.*, "R^3: Reinforced ranker-reader for open-domain question answering," in *AAAI*, 2018.

[4] Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang, "Multi-passage bert: A globally normalized bert model for open-domain 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)*, 2019, pp. 5878–5882.

[5] W. Yang *et al.*, "End-to-end open-domain question answering with bertserini," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 72–77.

[6] P. Qi, X. Lin, L. Mehr, Z. Wang, and C. D. Manning, "Answering complex open-domain questions through iterative query generation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2590–2602. DOI: 10.18653/v1/D19-1261. [Online]. Available: https://aclanthology.org/D19-1261.

[7] Z. Yang *et al.*, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018, pp. 2369–2380.

[8] X. Ho, A.-K. Duong Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020, pp. 6609–6625.

[9] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "MuSiQue: Multihop questions via single-hop question composition," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 539–554, 2022.

[10] D. Carmel and E. Yom-Tov, "Estimating the query difficulty for information retrieval," *Synthesis Lectures on Information Concepts, Retrieval, and Services*, vol. 2, no. 1, pp. 1–89, 2010.

[11] C. Hauff, V. Murdock, and R. Baeza-Yates, "Improved query difficulty prediction for the web," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 439–448.

[12] J. Mothe and L. Tanguy, "Linguistic features to predict query difficulty," in *ACM Conference on Research and Development in Information Retrieval, SIGIR, Predicting query difficulty-methods and applications workshop*, 2005, pp. 7–10.

[13] J. Pérez-Iglesias and L. Araujo, "Standard deviation as a query hardness estimator," in *International Symposium on String Processing and Information Retrieval*, Springer, 2010, pp. 207–212.

[14] A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits, "Predicting query performance by query-drift estimation," *ACM Transactions on Information Systems (TOIS)*, vol. 30, no. 2, pp. 1–35, 2012.

[15] 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*, ser. ICTIR '19, Santa Clara, CA, USA: Association for Computing Machinery, 2019, 55–58, ISBN: 9781450368810.

[16] 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*, 2021, pp. 2857–2861.

[17] 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*, 2022, pp. 201–209.

[18] W. Xiong *et al.*, "Answering complex open-domain questions with multi-hop dense retrieval," *International Conference on Learning Representations*, 2021.

[19] J. Welbl, P. Stenetorp, and S. Riedel, "Constructing datasets for multi-hop reading comprehension across documents," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 287–302, 2018.

[20] V. Mavi, A. Jangra, and A. Jatowt, "A survey on multi-hop question answering and generation," *arXiv preprint arXiv:2204.09140*, 2022.

[21] W. Xiong *et al.*, "Simple yet effective bridge reasoning for open-domain multi-hop question answering," *arXiv preprint arXiv:1909.07597*, 2019.

[22] Y. Zhang, P. Nie, A. Ramamurthy, and L. Song, "Answering any-hop open-domain questions with iterative document reranking," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 481–490.

[23] 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)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[24] V. Karpukhin *et al.*, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.

[25] X. Zhang *et al.*, "Answer complex questions: Path ranker is all you need," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '21, New York, NY, USA: Association for Computing Machinery, 2021, ISBN: 9781450380379.

[26] C. Wu *et al.*, "Triple-fact retriever: An explainable reasoning retrieval model for multi-hop qa problem," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, IEEE, 2022, pp. 1206–1218.

[27] Y. Feldman and R. El-Yaniv, "Multi-hop paragraph retrieval for open-domain question answering," *arXiv preprint arXiv:1906.06606*, 2019.

[28] Y. Nie, S. Wang, and M. Bansal, "Revealing the importance of semantic retrieval for machine reading at scale," in *Proceedings of the EMNLP-IJCNLP Conference*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2553–2566. DOI: 10.18653/v1/D19-1258. [Online]. Available: https://aclanthology.org/D19-1258.

[29] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, "Learning to retrieve reasoning paths over wikipedia graph for question answering," in *International Conference on Learning Representations*, 2020.

[30] Y. Zhu, L. Pang, Y. Lan, H. Shen, and X. Cheng, "Adaptive information seeking for open-domain question answering," *arXiv preprint arXiv:2109.06747*, 2021.

[31] R. Das *et al.*, "Multi-step entity-centric information retrieval for multi-hop question answering," in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019, pp. 113–118.

[32] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang, "Cognitive graph for multi-hop reading comprehension at scale," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019.

[33] N. Shao, Y. Cui, T. Liu, S. Wang, and G. Hu, "Memory augmented sequential paragraph retrieval for multi-hop question answering," *arXiv preprint arXiv:2102.03741*, 2021.

[34] C. Zhao, C. Xiong, C. Rosset, X. Song, P. Bennett, and S. Tiwary, "Transformer-xh: Multi-evidence reasoning with extra hop attention," in *International Conference on Learning Representations*, 2020.

[35] S. Li *et al.*, "Hopretriever: Retrieve hops over wikipedia to answer complex questions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 13 279–13 287.

[36] O. Kurland, A. Shtok, S. Hummel, F. Raiber, D. Carmel, and O. Rom, "Back to the roots: A probabilistic framework for query-performance prediction," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 823–832.

[37] Y. Zhao, F. Scholer, and Y. Tsegay, "Effective pre-retrieval query performance prediction using similarity and variability evidence," in *Advances in Information Retrieval*, C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. W. White, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 52–64, ISBN: 978-3-540-78646-7.

[38] J. He, M. Larson, and M. d. Rijke, "Using coherence-based measures to predict query difficulty," in *European conference on information retrieval*, Springer, 2008, pp. 689–694.

[39] C. Hauff, "Predicting the effectiveness of queries and retrieval systems," in *SIGIR Forum*, vol. 44, 2010, p. 88.

[40] 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*, 2002, pp. 299–306.

[41] B. He and I. Ounis, "Inferring query performance using pre-retrieval predictors," in *String Processing and Information Retrieval: 11th International Conference, SPIRE 2004, Padova, Italy, October 5-8, 2004. Proceedings 11*, Springer, 2004, pp. 43–54.

[42] D. Roy, D. Ganguly, M. Mitra, and G. J. Jones, "Estimating gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction," *Information processing & management*, vol. 56, no. 3, pp. 1026–1045, 2019.

[43] N. Arabzadeh, F. Zarrinkalam, J. Jovanovic, F. Al-Obeidat, and E. Bagheri, "Neural embedding-based specificity metrics for pre-retrieval query performance prediction," *Information Processing & Management*, vol. 57, no. 4, p. 102 248, 2020.

[44] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg, "What makes a query difficult?" In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 390–397.

[45] 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*, 2007, pp. 543–550.

[46] H. Zamani, W. B. Croft, and J. S. Culpepper, "Neural query performance prediction using weak supervision from multiple signals," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 105–114.

[47] X. Chen, B. He, and L. Sun, "Groupwise query performance prediction with bert," in *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II*, Springer, 2022, pp. 64–74.

[48] 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*, 2022, pp. 2148–2153.

[49] E. Krikon, D. Carmel, and O. Kurland, "Predicting the performance of passage retrieval for question answering," in *Proceedings of the 21st ACM International Conference on Information and Knowledge management*, 2012, pp. 2451–2454.

[50] 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*, 2019, pp. 43–46.

[51] H. Roitman, S. Erera, O. Sar-Shalom, and B. Weiner, "Enhanced mean retrieval score estimation for query performance prediction," in *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, 2017, pp. 35–42.

[52] H. Roitman, "An extended query performance prediction framework utilizing passage-level information," in *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, 2018, pp. 35–42.

[53] M. Yadegari, E. Kamalloo, and D. Rafiei, "Detecting frozen phrases in open-domain question answering," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1990–1996.

[54] C. Clark and M. Gardner, "Simple and effective multi-paragraph reading comprehension," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 845–855.

[55] D. Cohen, L. Yang, and W. B. Croft, "Wikipassageqa: A benchmark collection for research on non-factoid answer passage retrieval," in *The 41st International ACM SIGIR Conference on Research; Development in Information Retrieval*, ser. SIGIR '18, Ann Arbor, MI, USA: Association for Computing Machinery, 2018, 1165–1168, ISBN: 9781450356572.

[56] Y. Yang, W.-t. Yih, and C. Meek, "WikiQA: A challenge dataset for open-domain question answering," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 2013–2018.

[57] F. Scholer, H. E. Williams, and A. Turpin, "Query association surrogates for web search," *Journal of the American Society for Information Science and Technology*, vol. 55, no. 7, pp. 637–650, 2004.

[58] J. Lee, S. Yun, H. Kim, M. Ko, and J. Kang, "Ranking paragraphs for improving answer recall in open-domain question answering," in *Proceedings of EMNLP*, 2018, pp. 565–569.

[59] J. Mothe, L. Laporte, and A.-G. Chifu, "Predicting query difficulty in ir: Impact of difficulty definition," in *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, 2019, pp. 1–6. DOI: 10.1109/KSE.2019.8919433.