

# Am I on the Right Track? What Can Predicted Query Performance Tell Us about the Search Behaviour of Agentic RAG

Fangzheng Tian<sup>†</sup>, Jinyuan Fang, Debasis Ganguly, Zaiqiao Meng and Craig Macdonald

University of Glasgow, Scotland, United Kingdom

## Abstract

Agentic Retrieval-Augmented Generation (RAG) is a new paradigm where the reasoning model decides when to invoke a retriever (as a “tool”) when answering a question. This paradigm, exemplified by recent research works such as Search-R1, enables the model to decide when to search and obtain external information. However, the queries generated by such Agentic RAG models and the role of the retriever in obtaining high-quality answers remain understudied. To this end, this initial study examines the applicability of query performance prediction (QPP) within the recent Agentic RAG models Search-R1 and R1-Searcher. We find that applying effective retrievers can achieve higher answer quality within a shorter reasoning process. Moreover, the QPP estimates of the generated queries, used as an approximation of their retrieval quality, are positively correlated with the quality of the final answer. Ultimately, our work is a step towards adaptive retrieval within Agentic RAG, where QPP is used to inform the model if the retrieved results are likely to be useful.

## Keywords

Retrieval Augmented Generation, Agentic RAG, Query Performance Prediction, Adaptive Search

## 1. Introduction

Retrieval-Augmented Generation (RAG) is a technique that incorporates retrieved external knowledge into the generative process of a Large Language Model (LLM) [1]. By doing so, RAG alleviates the hallucination problem often observed in LLMs [2] and compensates for their limitations in accessing up-to-date or domain-specific knowledge [3], which is essential for fulfilling downstream tasks such as Question Answering (QA). The basic *standard* form of the RAG pipeline involves a single retrieval stage, based solely on an input question. However, standard RAG has limitations in its flexibility and adaptability, as it applies a uniform retrieval strategy to all input questions, regardless of whether the LLM already possesses sufficient internal knowledge to answer them. In addition, it is not well-suited for tasks that require complex or multi-step information gathering [4]. As a result, the potential benefits of integrating retrieval and generation are not fully realised, which can lead to suboptimal performance on complex downstream tasks.

To address the limitation of single retrieval in standard RAG, an alternative approach is to delegate the LLM itself to determine when to invoke searches and what queries to search for. This paradigm is termed as *iterative* or *Agentic RAG*. Instead of relying on a static, single-step retrieval, the LLM in Agentic RAG can iteratively formulate and issue queries, and ingest the results of the search before continuing its reasoning process. This allows the reasoning process to adapt in light of the retrieved information. Agentic RAG models such as Search-R1 [5] and R1-Searcher [6] have shown improved performance in question answering tasks, such as Natural Questions (NQ) [7] and HotpotQA [8].

In the workflow of Agentic RAG, an LLM begins answering a question by reasoning. During the reasoning process, if external knowledge is needed, the LLM can automatically generate intermediate

---

SIGIR'25: IR-RAG Workshop, July 17, 2025, Padua, Italy

<sup>†</sup>Corresponding author.

✉ f.tian.1@research.gla.ac.uk (F. Tian); j.fang.2@research.gla.ac.uk (J. Fang); debasis.ganguly@glasgow.ac.uk (D. Ganguly); debasis.ganguly@glasgow.ac.uk (Z. Meng); craig.macdonald@glasgow.ac.uk (C. Macdonald)

🆔 0009-0000-3282-0220 (F. Tian); 0000-0002-6886-5882 (J. Fang); 0000-0003-0050-7138 (D. Ganguly); 0000-0001-5374-0318 (Z. Meng); 0000-0003-3143-279X (C. Macdonald)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

queries that interrupt generation and trigger the search engine (acting as an agent). The top-ranked retrieved documents are then appended to the generated tokens, serving as the context for reasoning in the next iteration. These reasoning-retrieval iterations continue until the LLM decides to yield the final answer after reasoning or reaching the output token limit. While existing Agentic RAG models have shown effective empirical results in question-answering tasks [5, 6], there remains a limited understanding of the properties of the generated queries and their retrieval results, as well as how the retrieval component contributes to enhancing answer quality. To further enhance the performance of Agentic RAG by improving the reasoning-retrieval process, it is important to analyse the generated queries and their likelihood of retrieving relevant information. This motivates us to study Agentic RAG’s retrieval component and its downstream effects on answer quality.

While the role of the retrieval component in standard RAG systems has been extensively studied, its role within Agentic RAG remains under-explored. In standard RAG, factors such as retriever effectiveness, the cutoff of top-retrieved documents and the order of documents when they are appended into RAG prompts can all influence the answer quality. However, Agentic RAG systems introduce additional complexity by involving multiple retrievals throughout the reasoning process. Therefore, in Agentic RAG, answer quality may be affected by the cumulative influence of all the retrievals conducted during the reasoning iterations, presenting new challenges for analysing it.

A significant challenge in evaluating the impact of retrieval quality within the Agentic RAG framework is the absence of ground truth relevance judgements for the intermediate generated queries in each reasoning-retrieval iteration. To tackle this challenge, we resort to Query Performance Prediction (QPP) as a possible estimator of the retrieval quality for generated queries. QPP methods aim to predict the effectiveness of the retrieval for a query without human-annotated relevance judgements, typically by analysing features of the query and the retrieved documents [9, 10, 11]. Given that both the generated queries and their corresponding retrieval results are available within the Agentic RAG (implemented by Py-Terrier-RAG [12]), QPP offers a viable approach for approximating retrieval quality and investigating the role of retrieval in the system.

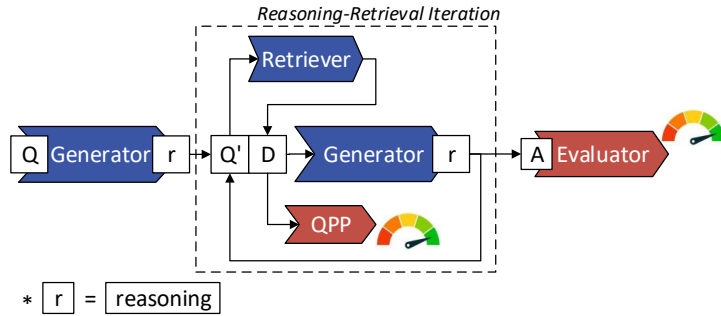
To this end, in this paper, we conduct experiments examining answer quality for two recent Agentic RAG models using three different retrieval models. For each retriever, we study how estimated query performance, estimated by several unsupervised QPP models, varies throughout the iterations in the reasoning process. Furthermore, we investigate how the estimated query performance correlates with answer quality.

In summary, this paper contributes a first study of the utility of QPP in the querying behaviour of two typical agentic RAG models, Search-R1 [5] and R1-Searcher [6]. Moreover, we investigate the impact of retriever effectiveness on the iterative reasoning-retrieval process in Agentic RAG. We conduct experiments on the 3610 test queries of the Natural Questions (NQ) QA dataset [7]. Based on our experiment results, we observe that (1) more effective retrieval models can reduce the length of Agentic RAG’s reasoning process and enhance the answer quality; (2) QPP estimation about the generated queries is a potential signal for predicting the quality of the final answers.

The remainder of the paper is organised as follows. In the next section, we review related work about RAG and the role of the retrieval component in it. We then provide a systematic description of the Agentic RAG system in Section 3. In Section 4, we state our research questions and introduce the setting of Agentic RAG pipelines and the experimented QPP methods. According to the experiment results, we answer the research questions in Section 5 and conclude the paper with perspectives on applying QPP in facilitating Agentic RAG in Section 6. Appendix 7 demonstrates a case study on two outputs - one with a correct answer and one with a wrong answer - to investigate the topical changes in the intermediate queries.

## 2. Related Work

We position our work within the existing literature, in terms of retrievers in RAG (Section 2.1), iterative RAG models (Section 2.2) and query performance prediction (Section 2.3).



**Figure 1:** A conceptual overview of an Agentic RAG. Given an input question  $Q$ , the model performs initial reasoning  $r$ , following which it iteratively generates queries  $Q'$ . For each such generated query, the retriever component incorporates the top-ranked documents into the ongoing reasoning process. This process continues until the model produces a final answer  $A$  after  $Iter$  iterations (the number of iterations required depends on the input question). A post-retrieval query performance prediction (QPP) model is applied to the intermediate retrieval results. An evaluator then assesses the quality of the generated answer. In this paper, we seek to find correlations between the QPP estimates and the answer quality.

## 2.1. The Role of Retriever in RAG

RAG models enhance an LLM’s generative process by providing access to external knowledge [3], which helps mitigate hallucinations [13] and improve overall performance. In RAG, particularly for the decoder-only LLMs [14, 15, 16], retrieved documents are directly concatenated with the input question [1] as part of the prompt. RAG has been applied to a number of downstream tasks, including QA [3, 17], fact verification [18, 19] and summarisation [20].

Unlike traditional information systems, where users consume the retrieval results, RAG systems pass the retrieval results directly to a generator (i.e. an LLM). In RAG systems, it is the LLM, rather than the user, that consumes the retrieved documents to generate the final response. Traditional IR metrics, such as nDCG, may not precisely measure the usefulness of retrieved documents in this context, due to the difference between human-defined relevance and the way LLMs interpret and leverage information [21]. Therefore, the quality of retrieved documents in RAG should be evaluated not by topical relevance but rather by their impact on the downstream task performance [21, 22, 23]. To better understand this relationship, recent studies have examined how different characteristics of the retrieved documents affect RAG performance, such as the number of documents [23], the position of relevant documents [24] and the relative position of the query [25]. Despite these findings, existing studies about the role of retrieval in RAG focus on single-retrieval RAG pipelines. There remains a **gap in understanding how retrieval influences performance in RAG pipelines that involve multiple retrievals**.

## 2.2. Iterative/Agentic RAG

In standard RAG models, external knowledge is retrieved and provided to the LLM only once [3]. However, a single retrieval might not be enough to answer some difficult queries [26, 27]. To address this issue, recent studies, such as IRCOT [28], KiRAG [29] and Search-O1 [30], have explored RAG frameworks that incorporate multiple (or iterative) retrievals. The number of retrievals can be determined dynamically using techniques such as question decomposition [4] or by iteratively detecting uncertain tokens in the generated answer [31]. The queries executed in these dynamic retrievals are often refined versions of the original query [32] or are generated to resolve ambiguities in the current answer [4].

More recent RAG frameworks integrate dynamic retrieval directly into the LLM’s reasoning process [5, 6, 30], letting the LLM itself autonomously conduct searches. This is a new paradigm known as Agentic RAG [33]. Search-O1 achieves this integration by prompt engineering [30], whereas more advanced models such as Search-R1 [5] and R1-Searcher [6] employ reinforcement learning to train the LLM to autonomously invoke retrievals during the reasoning process. These methods have demonstrated improved performance on QA benchmarks [7, 8]. In this paper, **we investigate the reasoning-retrieval**

### Prompt

Answer the given question. You must conduct reasoning inside `<think>` and `</think>` first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by `<search>` query `</search>` and it will return the top searched results between `<information>` and `</information>`. You can search as many times as your want. If you find no further external knowledge needed, you can directly provide the answer inside `<answer>` and `</answer>`, without detailed illustrations. For example, `<answer>` Beijing `</answer>`.

Question: **who will take the throne after the queen dies?**

### Output - Iteration 1

`<think>`I need to find out who will take the throne after the queen dies. I'll search for it.`</think>` `<search>` **who will take the throne after the queen dies** `</search>`  
`<information>`Doc 1(Title: "Succession to the British throne") Catholics are eligible .... Doc 2 .... Doc 3 ....`</information>`

### Output - Iteration 2

`<think>`I found out that after the queen dies, her heir apparent will take the throne. I need to find out who the heir apparent is for Queen Elizabeth II.`</think>` `<search>` **heir apparent to Queen Elizabeth II** `</search>`  
`<information>`Doc 1(Title: "Heir apparent") rank behind her brothers regardless of their ages .... Doc 2 .... Doc 3 .... `</information>`

### Output - Answer

`<think>`I found out that the heir apparent to Queen Elizabeth II is her eldest son, Charles, Prince of Wales. Now I can provide the answer.`</think>` `<answer>` **Charles, Prince of Wales** `</answer>`

**Figure 2:** Example prompt for Search-R1 and its output over two reasoning-retrieval iterations for a query from the Natural Questions dataset. In each iteration, the top three documents retrieved by E5 are used to inform the reasoning process. In this example, the generator is tasked with identifying the successor to the late Queen. Search-R1 performs two iterations with generated queries before producing the correct answer.

**iterations in RL-trained Agentic RAG models, focusing on the role of the retriever and its impact on answer quality.**

### 2.3. Query Performance Prediction

Query performance prediction (QPP) is a technique used to estimate the quality of retrieval results for an input query without human-assessed relevance judgements. QPP methods are generally categorised into **pre-retrieval** and **post-retrieval**, based on whether the prediction is made before or after the retrieval process.

Pre-retrieval QPP methods solely rely on the features of the query itself [34] – such as inverse document frequency (IDF) [35], query length – to infer the query’s specificity and predict the retrieval performance. Because they do not consider retrieval results, their predictions are independent of the retriever used and are generally less accurate [36].

In contrast, post-retrieval QPP methods can use the features of the retrieval results, including retrieval score [9, 37, 38, 39], dense embeddings [10, 40] and the document texts [41]. Among numerous post-retrieval QPP methods, supervised methods directly train models to predict specific target IR metrics [41, 42]. Unsupervised post-retrieval QPP models analyse the retrieval quality by analysing certain properties of the retrieval result, such as the distinctiveness of the top-retrieved document [39, 9], semantic concentration [40], coherence [43, 10], etc. **However, QPP methods have not been applied in examining the query generation process in Agentic RAG.** Given the lack of ground truth

for evaluating the queries generated during reasoning-retrieval iterations, QPP provides a promising technique for investigating the role of retrieval in Agentic RAG systems.

### 3. Agentic RAG

In this section, we describe how we incorporate a query performance prediction (QPP) component in an **Agentic RAG** workflow, as the LLM uses the retriever as an agent.

#### 3.1. Overview

Figure 1 shows the workflow of an Agentic RAG. In this system, an LLM-decoder, serving as the **generator**, automatically drives a **retriever** for searching information. Given an input question  $Q$ , the generator performs reasoning towards deriving an answer. During the reasoning process, if the generator perceives that additional external information is necessary, it generates a query  $Q'$ , enclosed within the tags `<search>` and `</search>`.<sup>1</sup> The retriever then executes this query to retrieve potentially relevant documents, which are incorporated into the ongoing reasoning process. This may prompt the generator to initiate further retrievals. Each cycle of this ‘reasoning’ and ‘retrieval’ constitutes an *iteration* in agentic RAG. This iterative process continues until the generator decides to yield the final answer  $A$ . The total number of reasoning-retrieval iterations undertaken to reach the final answer for the input query is termed as **length of the reasoning process**, denoted as  $Iter$  in this paper.

To exemplify the reasoning process of agentic RAG, Figure 2 presents a sample prompt and output from Search-R1, which is a commonly-used Agentic RAG model [5]. The top part of the figure shows the prompt that explicitly instructs the model to invoke searches when more knowledge is required for answering the input question. In this example, the generator invokes the retriever twice with progressively refined queries before yielding the final answer. In this example, the answer generated after two steps ( $Iter=2$ ) correctly answers the input question about the successor of the late Queen.

#### 3.2. Retrieval

When configuring a RAG model, various retrieval pipelines can be integrated with the generator. They can range from an efficient sparse retriever, such as BM25, to more complex retrieval-re-ranking pipelines. Using PyTerrier-style notation, we can denote any retrieval pipeline for RAG. For instance, a MonoT5 cross-encoder re-ranking the output of BM25 as:  $BM25 \% k \gg MonoT5$ <sup>2</sup>. Alternatively, a dense retrieval pipeline may be formulated with a query-encoder and an existing vector database of pre-encoded passage vectors (e.g.  $E5 \gg VectorIndex$ ).

This flexibility in configuring the retrieval pipeline allows us to easily instantiate Agentic RAG models with various retrievers<sup>3</sup>, which can result in different downstream RAG performances. Intuitively, the top- $k$  documents (where  $k$  is typically small in RAG) retrieved by more effective retrievers are more likely to contain relevant information. Such documents can better complement the LLM’s internal knowledge, potentially reducing the reasoning-retrieval iterations required to generate an accurate answer. However, since retrieval is only an intermediate step in the Agentic RAG, the degree to which the retrieval pipeline impacts answer quality remains an open question—one that we investigate in our research.

---

<sup>1</sup>For clarity, in this paper, we always refer to the user’s input as a question, and the search queries generated by the model as queries.

<sup>2</sup>In PyTerrier [44],  $a \gg b$  denotes a pipeline composition operation: the output of  $a$  is passed to  $b$ ;  $\%k$  denotes applying a rank cutoff of  $k$  per query.

<sup>3</sup>An example of the RAG pipeline with  $BM25 \gg MonoT5$  as the retriever can be found here: [https://github.com/terrierteam/pyterrier\\_rag/tree/main/examples](https://github.com/terrierteam/pyterrier_rag/tree/main/examples).

### 3.3. QPP of Generated Queries

To investigate the influence of the retrieval component in Agentic RAG models, we need to first estimate the quality of the retrieved documents. In this context, retrieval quality refers to how relevant the retrieved documents are to the generated sub-query, and how useful these documents will be to ‘guide’ the generator towards a high-quality (relevant and factually correct) answer.

Since the retrieval component receives queries that are dynamically generated by the LLM’s reasoning process, there is no relevance ground truth for explicitly evaluating retrieval quality. To address this problem, in this initial research, we investigate whether a Query Performance Predictor (QPP) estimate may be substituted as a reasonable approximation for the relevance or the usefulness of these documents retrieved during the reasoning phase.

In this work, we focus on post-retrieval QPP models, which estimate retrieval quality based on properties of the retrieved documents, such as their score distribution or embedded representations, instead of pre-retrieval ones, where the prediction depends only on the input query characteristics. A post-retrieval QPP model is a more appropriate choice for this task of estimating the usefulness of the retrieved documents in guiding the reasoning process towards the correct answer.

A QPP model can be integrated within the Agentic RAG pipeline to dynamically estimate the quality of retrievals in each reasoning-retrieval iteration. This component is depicted as the red box connected to the retrieval result in Figure 1.

Inclusion of a QPP model into the agentic RAG workflow is a step towards better understanding the quality of the intermediate queries generated during the reasoning-retrieval iterations. This is also a step towards developing an adaptive agentic RAG workflow that is capable of dynamically deciding to leverage the top-retrieved documents for modifying the reasoning process only if their estimated performance is high. In this initial work, we focus only on the analysis of the QPP estimates and their relation with the downstream answer quality, leaving the exploration of an adaptive QPP-based workflow for agentic RAG as future work.

Finally, it is worth mentioning that the pragmatic applicability of QPP methods depends on the retrieval model type used in RAG. For instance, it is not a pragmatic choice to apply dense QPP approaches, such as [40, 10], to retrievers using sparse representations (e.g., BM25). That is because if document embeddings are available, we may as well use them to improve the retriever itself, e.g., a bi-encoder retriever, instead of solely utilising them for QPP [10].

## 4. Experimental Setup

We now elicit our research questions (Section 4.1), our experimental setup in terms of retrievers and agentic rag models (Section 4.2 and query performance predictions (Section 4.3).

### 4.1. Research Questions

In this work, we focus on the iterative query generation process in two agentic RAG models, namely Search-R1 [5] and R1-Searcher [6], and specifically the quality of queries generated at each iteration. Indeed, if the effectiveness of the retriever used by the RAG system changes, the number of iterations required to reach a final answer may change. During their iterative reasoning–retrieval processes, the quality of the generated queries may shift - becoming more specific or more generic - which, in turn, can influence the final answer quality. Since human-assessed relevance judgments are not available for these dynamically generated queries, we estimate their effectiveness using query performance predictors. Based on this, we formulate our research questions as follows:

- **RQ-1:** What is the effect of different retriever configurations on answer quality and number of iterations?
- **RQ-2:** How does the estimated performance of generated queries vary as the number of iterations increases?

- **RQ-3:** How does the estimated performance of generated queries correlate with the quality of the generated answers?

## 4.2. Agentic RAG Pipeline Configurations

**Query set** We use the Natural Questions (NQ) test set [7], which contains a total of 3,610 questions. Each question is accompanied by gold (ground-truth) answers, as found from Wikipedia by manual annotation. These ground-truth answers are then used to evaluate the quality of the answers generated by Agentic RAG models. Following common practices, we use F1 and Exact Match % (EM) to measure answer quality [45].

**Document corpus** A snapshot of English Wikipedia from 2018, as used with the NQ dataset.

**Generator** We experiment with two recent Agentic RAG models trained with Reinforcement Learning (RL) from QWEN2.5-7B [16]:

- **Search-R1** [5]: This model is trained by single-stage outcome-based RL on NQ and HotpotQA training sets [8]<sup>4</sup>.
- **R1-Searcher** [6]: This model is trained via two-stage outcome-based RL on HotpotQA and 2Wiki-MultiHopQA [46]. Since NQ was not used in the training of the R1-Searcher model, our experiments for the R1-Searcher correspond to that of an out-of-domain evaluation.

Both Search-R1 and R1-Searcher have demonstrated improved answer quality over standard RAG formulations and other recent iterative RAG formulations such as IRCot [28].

**Retriever** We address both sparse/lexical retrieval, as well as neural re-ranking and dense retrieval:

1. **BM25:** A classic lexical retrieval model;
2. **MonoT5:** A retrieval-re-ranking pipeline with cross-encoder BM25>>MonoT5. The top-20 documents retrieved by BM25 are re-ranked by MonoT5 [47].
3. **E5:** The bi-encoder model E5 [48] for general purpose retrieval. It is commonly used as the retriever in RAG pipelines, including in the original papers for Search-R1 and R1-Searcher.

The top-3 documents retrieved by each retrieval model are incorporated into the reasoning process. These models span a range of retrieval strategies with varying effectiveness. In general, re-ranking BM25 results with MonoT5 improves relevance, as the cross-encoder can better model semantic similarity. However, the utility of MonoT5 may be hindered by the limited recall offered by BM25 (relevant documents not retrieved in the top 20 by BM25 cannot be retrieved by MonoT5). As a dense retrieval model, E5 may retrieve passages that do not match any of the terms in the queries; therefore, we expect that E5 should be more effective than MonoT5 (and BM25).

All of the above RAG and retrieval pipelines are implemented using PyTerrier [44]<sup>5</sup> and PyTerrier-RAG [12]<sup>6</sup>.

## 4.3. Query Performance Predictors

In our experiments, we apply QPP methods on the generated queries to estimate how relevant the retrieval results are to those generated queries, which may be related to the usefulness of those queries in answering the original question. Specifically, we test:

<sup>4</sup>[https://huggingface.co/PeterJinGo/SearchR1-nq\\_hotpotqa\\_train-qwen2.5-7b-em-ppo](https://huggingface.co/PeterJinGo/SearchR1-nq_hotpotqa_train-qwen2.5-7b-em-ppo)

<sup>5</sup><https://github.com/terrier-org/pyterrier>

<sup>6</sup>[https://github.com/terrierteam/pyterrier\\_rag](https://github.com/terrierteam/pyterrier_rag)

**Table 1**

Average quality (measured by EM & F1) of the answers generated by Search-R1 and R1-Searcher on the Natural Questions dataset, using various retrievers. We also report the average number of reasoning/querying iterations (denoted *Iter*).

Model	Search-R1			R1-Searcher			
	Retriever	BM25	MonoT5	E5	BM25	MonoT5	E5
EM		0.3391	0.3873	0.4838	0.2089	0.2307	0.3075
F1		0.4185	0.4709	0.5687	0.2586	0.2865	0.3736
<i>Iter</i>		2.52	2.19	2.00	2.20	2.03	1.99

**Table 2**

Spearman’s  $\rho$  correlation between number of iteration (*Iter*) and answer quality (F1).

Model	Search-R1			R1-Searcher			
	Retriever	BM25	MonoT5	E5	BM25	MonoT5	E5
$\rho(\text{Iter}, \text{F1})$		-0.2803	-0.2991	-0.3168	-0.1872	-0.1490	-0.1110

**Predictors based on Retrieval Scores** We experiment with

- **NQC** [9]: NQC is the classic QPP method based on the standard deviation of retrieval scores for retrieved documents. It is applicable to all three retrieval configurations. We use the retrieval scores of the top-100 retrieved documents in NQC, which is a reasonable setting to acquire accurate QPP estimates [10].
- **Max(Score)** [10]: For the neural retrievers, in which the retrieval scores can more accurately reflect the relevance between a query-document pair, there is a high likelihood that the score of the top-ranked document likely acts as an effective estimator of query performance. This predictor, denoted as **Max(Score)** [10], reflects the estimated upper bound of the relevance of the retrieved context, which is likely to be reliable due to a small cutoff on the retrieved lists, as is usually the case in Agentic RAG.

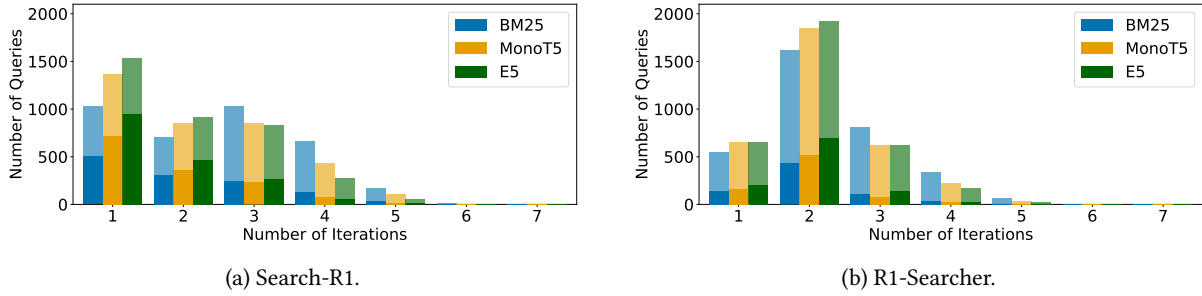
**Embedding-based Predictors** Additionally, for the bi-encoder model (E5), which uses dense embeddings of the retrieved documents, we test two recent embedding-based QPP models:

- **A-Pair-Ratio** [10]: A coherence-based predictor that calculates the ratio between the pairwise coherences of the top-ranked and last-ranked documents in the retrieval result. In particular, we compute this ratio among the first five and the last five documents in the top-50 retrieved documents, as described in [10].
- **Dense-QPP** [40]: A recent geometric-based predictor that calculates the volume of the minimum hypercube encompassing the embeddings of the top- $k$  retrieved documents and the query. We calculate the hypercube based on the top 3 documents, as per our RAG configuration.

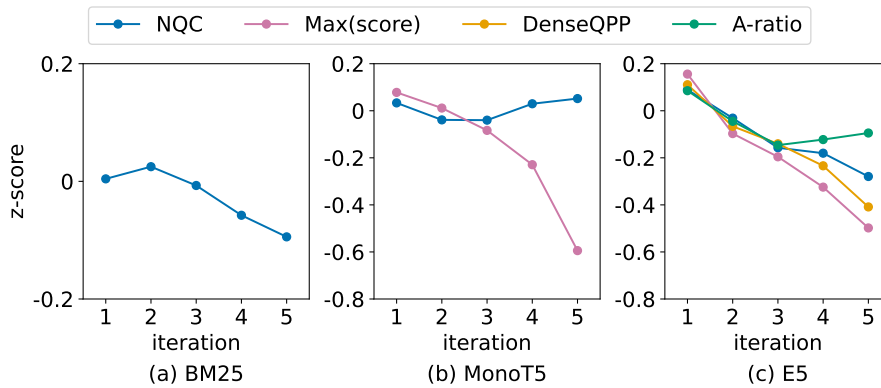
Finally, considering that there can be multiple intermediate generated queries for answering an input question, when evaluating the correlation between QPP estimates and the quality of generated answers, in this initial study, we only consider the QPP estimates for the first generated query.

## 5. Results

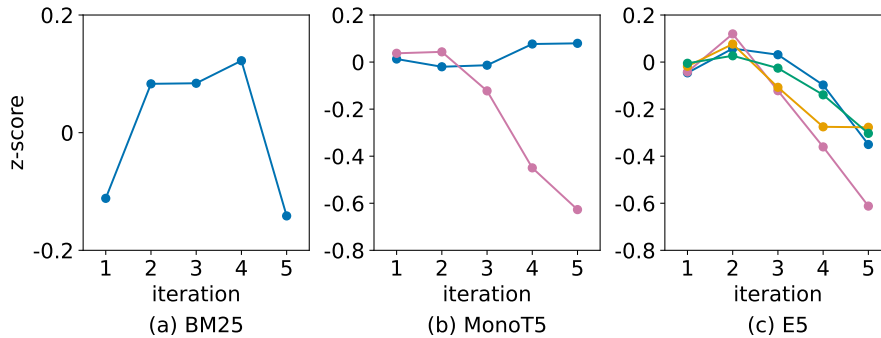
We now report findings for each of our research questions.



**Figure 3:** Distribution of the reasoning lengths (number of iterations) across 3610 NQ dataset questions for (a) Search-R1 and (b) R1-Searcher agentic RAG models. For both these models, we show the number of iterations for the three retrieval models experimented with. In each bar, the height of the part with a darker colour is the number of queries with a correct answer (EM=1). Notably, BM25 exhibits a longer tail with a higher number of reasoning iterations compared to that of the neural retrieval models, MonoT5 and E5.



**Figure 4:** Average QPP estimates for the generated queries in each reasoning-retrieval iteration for Search-R1. QPP estimates are normalised by z-score.



**Figure 5:** Average QPP estimates of generated queries in each reasoning-retrieval iteration for R1-Searcher, legend as per Figure 4.

### 5.1. RQ-1: Effect of different retriever configurations on answer quality and number of iterations

Table 1 presents the average answer quality (measured in terms of EM and F1), and also the number of retrieval iterations (*Iter*) obtained with Search-R1 and R1-Searcher models using three different retriever configurations. The results show that for both Search-R1 and R1-Searcher, more effective retrievers – MonoT5 and E5 – generally lead to fewer reasoning-retrieval iterations initiated by the LLM before the final answer is generated.

Additionally, the answers generated using results from more effective retrievers tend to be of higher

quality. For example, Searcher-R1 achieves an F1 score of 0.4185 with an average of 2.52 iterations when using BM25 as the retriever. Using the E5 retriever leads to an increase in F1 score, to 0.5687, and this, in fact, is obtained with a reduced average reasoning length of 2.0 iterations. These findings suggest that retrieval effectiveness influences both the length of the reasoning process and the quality of the generated answers.

Figure 3 provides a more detailed view of the number of iterations across the 3610 NQ test queries for both Search-R1 (a) and R1-Searcher (b). The dark part of each bar shows correct answers; the light part shows wrong answers. From the plots of Figure 3, we observe that, for both of the Agentic RAG models, using BM25 as the retriever exhibits longer tails, indicating an extended reasoning process compared to models that use E5 and MonoT5 as the retriever.

This observation raises a further question of whether the number of iterations (*Iter*) is correlated with average answer quality in Agentic RAG. Table 2 reports the Spearman’s  $\rho$  correlation between them within each RAG configuration. From the left side of the table, we observe a moderate negative correlation between *Iter* and F1 score for Search-R1 (i.e. more iterations implies lower likelihood of a good answer). For R1-Searcher, the correlations are weaker but remain negative, suggesting that shorter reasoning processes tend to lead to higher answer quality.

Another observation for Search-R1 is that a more effective retriever shows a stronger negative correlation, e.g., the correlation of both E5 and MonoT5 is lower than that of BM25. A plausible explanation for this is that more effective retrieval models return documents that are likely to be more relevant to the generated queries at each iteration, allowing the system to potentially generate better follow-up queries and reach the final answer with fewer iterations.

**To conclude for RQ-1**, for both the Search-R1 and R1-Searcher Agentic RAG models, adopting more effective retrievers can shorten the reasoning process and lead to final answers of higher quality. We now investigate whether QPP estimates of query performance can provide any indication of answer quality.

## 5.2. RQ-2: Predicted Generated Query Quality versus Iterations

Figures 4 and 5 show the average QPP values for queries generated in the first five iterations across the test set. Each figure contains three sub-plots, one for each retrieval model. Different QPP predictors are applied to each retriever, based on their applicability. Note that we normalise each predictor by z-score, to allow different predictors (with different ranges) to be more easily shown in a single plot. Finally, as noted earlier when analysing Figure 3, few questions have more than 5 iterations, so we show QPP values for iterations 1-5.

Overall, across both Figures 4 and 5, we observe a decreasing trend in QPP estimates as the reasoning process progresses through iterations. This suggests that the quality of retrieval, in terms of relevance to the generated queries, generally deteriorates over iterations. This overall downwards trend is consistent across the QPP methods evaluated on both Search-R1 and R1-Searcher pipelines. The only exception appears when NQC is applied to MonoT5, as shown in Figure 4(b) and Figure 5(b).

However, this exception is explainable, as NQC is not an accurate predictor for query performance in MonoT5. The score distribution produced by MonoT5, as argued by Datta et al. [11], does not align with the assumptions underlying the statistical QPP methods - such as NQC. As a result of this, the NQC predictor may not accurately reflect the relevance or the usefulness of the documents retrieved for the purpose of LLM reasoning.

Additionally, these decreasing trends in the QPP estimates are consistent for the better-performing QPP models, e.g., Max(Score) for Mono-T5. A likely explanation for this is that accurate QPP estimates correlate with the usefulness of the intermediate retrieved documents in answering the generated query. For instance, Max(Score) for Mono-T5 can be interpreted as an upper bound on the topical relevance between the query and the retrieved documents. If this score is low, the retrieved content is unlikely to satisfy the information need of the query – potentially prolonging the overall reasoning process.

**To conclude for RQ-2**, based on the predictions by the QPP methods employed, it can be observed that the average quality of queries generated by the Agentic RAG models tends to decrease as the

**Table 3**

Spearman’s  $\rho$  correlation between QPP estimation (for the query generated in the first iteration) and answer quality (measured by F1), for both models with various retrievers. A ‘-’ value in the table denotes that a predictor is not applicable for a given retriever (as discussed in Section 4.3). All observed correlations are significant.

QPP	Agentic RAG Models					
	Search-R1			R1-Searcher		
	BM25	MonoT5	E5	BM25	MonoT5	E5
NQC	.1297	.0524	.2394	.1205	.0096	.0515
Max(Score)	-	.2383	.2369	-	.1735	.0919
Dense-QPP	-	-	.1871	-	-	.0834
A-Pair-Ratio	-	-	.2497	-	-	.0297

number of reasoning iterations increases. The most likely reason for this is that the queries for which the answer generation converges in a small number of iterations are likely to be more central to the information need of the input question (and hence expected to be of higher quality as estimated by a QPP model). On the other hand, for questions that end up taking a larger number of iterations, the generated answers are not of high quality (recall the decreasing EM along iterations shown in Figure 3, and the negative correlations between the number of iterations and F1-score from Table 2). It is likely that the lower answer quality associated with an increase in the number of iterations is due to more noise being added to the generated text for reasoning itself by the low-quality retrievals. To better illustrate this phenomenon, we study the outputs of two example queries of Search-R1 in Appendix 7, which shows that low-quality retrievals for off-topic intermediate queries lead to a wrong answer with more iterations..

### 5.3. RQ-3: Correlation between Predicted Query Performance and Answer Quality

We now turn to examine whether the QPP values of the queries are correlated with the final answer quality. In particular, we postulate that with a high number of queries being answered in a single iteration (see Section 5.1), the retrieval quality of the first query is most likely to be indicative of the answer quality. Hence, we focus on the correlation between the QPP score of the first generated query and the F1 score of the final output answer. Table 3 reports the observed correlations. From the table, we see that all correlations are consistently positive, suggesting that a higher QPP estimate for the first generated query is associated with higher-quality answers. However, these correlations are weak, and less than what has been achievable for standard retrieval tasks (which can be as high as Spearman’s  $\rho > 0.5$ ) [11, 10]. This emphasises that QPPs are not entirely capturing final answer quality.

Comparing the two Agentic RAG models, the observed correlations are higher for Search-R1 than for R1-Searcher. Considering the answer quality of these two Agentic RAG frameworks in Table 1, QPP estimates may be better correlated with the quality of answers generated by more effective RAG systems, in that Search-R1 makes better use of the retrieved documents, and hence is more susceptible to poor retrieval, leading to higher correlations.

When BM25 is used as the retriever, both pipelines show a weak but positive correlation using the NQC predictor, around 0.12–0.13. When the retrieval results are re-ranked by MonoT5, the correlation for NQC decreases (as expected because NQC is not an effective QPP model for neural re-rankers [11]). In contrast, the QPP method Max(Score) achieves higher correlations, around 0.2. When E5 is used for retrieval, all QPP predictors in the Search-R1 pipeline show stronger correlations, particularly the coherence-based predictor A-ratio [10], achieving the highest correlation at 0.25.

These observations indicate that QPP can serve as a useful signal in predicting the answer quality in the setting of Agentic RAG. However, in this initial study, we focus on correlating the answer quality with the QPP about the first generated query; higher correlation may be observed with improved aggregation methods of the QPP estimates in the multi-iteration reasoning process [49].

**To conclude for RQ-3**, we found that QPP estimates of the queries generated in the first iteration of both Agentic RAG models are positively correlated with the quality of the final answer. Although the correlations are weak, they indicate promise worth further investigation.

## 6. Conclusions

In this work, we analysed the retrieval component in Agentic RAG models, using QPP as a tool to estimate the retrieval quality for queries generated by the Search-R1 and R1-Searcher models. Our experiments show that, for both these models, more effective retrieval models (e.g., E5) tend to yield higher-quality answers with fewer reasoning-retrieval iterations. Moreover, by evaluating QPP estimates of the generated queries at each iteration, we observe that retrieval quality tends to decrease over the course of the reasoning process. A case study in Appendix 7 suggests that the ambiguity of the original query may be a reason leading to this decrease in retrieval quality. Finally, we observe a weak positive correlation between the QPP estimations of the first generated query and the final answer quality. While the correlations are weak, it nonetheless provides a potentially useful signal in predicting the quality of generated answers. Overall, our results emphasise the importance of the first retrieval iteration in final answer quality.

In terms of limitations, this study focused on just two models: Search-R1 and R1-Searcher, while other Agentic RAG models, such as Search-O1 [30] may exhibit different behaviours. Furthermore, our experiments are conducted only on the NQ dataset, while Agentic RAG models may be better suited to multi-hop QA questions, such as HotpotQA [8]. We plan to address these limitations in a future extension.

This work opens the doors to using IR techniques, including QPP, to improve the searching behaviour of Agentic RAG models. For instance, QPP may be used to adaptively decide whether the retrieval results are useful enough to show to the reasoning model; alternatively, QPP estimates could serve as a reward signal during the training time to help the model learn to execute better queries. This goes beyond the reward signals used in the training of R1-Searcher, which only reward the model for making search requests and generating responses that are validly formatted [6]. We believe that, in the era of RAG, applying QPP to enhance the LLM’s ability to better utilise IR retrieval models is a promising research direction.

## 7. Appendices

In this Appendix, we study one negative example (7.1) and one positive example (7.2) output from Search-R1 with retriever E5. The purpose is to investigate the reasons leading to the degradation of average retrieval quality in later iterations. 7.3 presents a brief insight from the analysis of the examples.

### 7.1. Negative Example: Ambiguous Question Causes Drifted Sub-Queries and Low-Quality Answer

Figure 6 shows the studied negative example. In this example, the input question “chris cornell she’s going to change the world” asks for the name of the Chris Cornell song that contains the lyric “She’s going to change the world.” The query is formulated very ambiguously; consequently, it confuses the Agentic RAG model. The correct answer should be “Can’t Change Me”, but the RAG model instead yields Eric Clapton, a singer known for the song “Change the World”.

To answer this question, Search-R1 invokes the retriever (E5) three times. In the first retrieval, the intermediate query is identical to the input. From the reasoning preceding the search, we observe that the LLM treats “She’s going to change the world” as a song title rather than a lyric excerpt. In the second retrieval, the intermediate query omits “Chris Cornell”, one of the two key entities from the

original question. As a result, the topic drifts significantly. In the third search, the retrieved documents focus on a song titled “Changed the World”, which is only weakly related to the intermediate query, indicating a lower-quality retrieval.

The QPP estimates (using  $\text{Max}(\text{Score})$ ) for the three iterations are 0.8370, 0.7870 and 0.8511, which are lower than those in the positive examples discussed in the next section ( $\text{Max}(\text{Score}) > 0.9$ ). It is worth noting that retrieval quality does not necessarily decline monotonically. In this case, the ambiguity in the original question causes the LLM to misinterpret the user’s information need, ultimately leading to an incorrect answer.

## 7.2. Positive Example: Specific Question with High Quality Retrievals

Figure 7 shows the positive example under study. In this case, the input question: “when was the last time villanova won the ncaa basketball championship?” represents a very clear information need. The Search-R1 model uses two reasoning-retrieval iterations to answer this question. After the first retrieval, which reformulates the original query, the model initiates a second search — another reformulation of the same question. After two rounds, the model correctly outputs 2018 as the answer.

The QPP estimates (using  $\text{Max}(\text{Score})$ ) for these two iterations are 0.9263 and 0.9268, respectively. Compared to the negative example with three iterations, these high values indicate strong retrieval quality. Such high-quality retrievals enable the RAG model to arrive at the correct answer. Since questions requiring fewer iterations tend to rely on more effective intermediate retrievals, their average estimated query performance is generally higher in earlier iterations.

However, this example also reveals a persistent issue in Search-R1 and other agentic models: repetition of intermediate queries. With E5 as the retriever and a cutoff of 3 iterations, 262 out of 3610 queries generated identical sub-queries during reasoning. This indicates insufficient utilisation of retrieved documents in current Agentic models. Incorporating QPP or other signals into the LLM could help achieve more efficient use of retrieved information.

## 7.3. Insights from the Case Study

From the above analysis, we observe that questions requiring more iterations in their reasoning process tend to involve lower-quality intermediate retrievals, which do not positively contribute to the correct final answer. Conversely, reasoning trajectories that lead to correct answers generally involve fewer iterations, generally yielding higher-quality retrievals. When we compute the average (predicted) retrieval quality by iteration, the early iterations — populated by questions with shorter reasoning chains - exhibit higher mean quality than later iterations.

However, in the positive example (see Appendix 7.2), we still observe the current Agentic RAG models repeating identical queries in intermediate retrieval steps. This redundancy can reduce the efficiency of the system and underscores the need to improve the Agentic workflow further.

## Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4 and Grammarly in order to: Grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

## References

- [1] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, 2024. URL: <https://arxiv.org/abs/2312.10997>. arXiv: 2312.10997.

- [2] N. Dziri, S. Milton, M. Yu, O. Zaiane, S. Reddy, On the origin of hallucinations in conversational models: Is it the datasets or the models?, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 5271–5285. URL: <https://aclanthology.org/2022.naacl-main.387/>.
- [3] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020. URL: <https://arxiv.org/abs/2005.11401>.
- [4] Z. Jiang, F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, G. Neubig, Active retrieval augmented generation, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 7969–7992. doi:10.18653/v1/2023.emnlp-main.495.
- [5] B. Jin, H. Zeng, Z. Yue, J. Yoon, S. Arik, D. Wang, H. Zamani, J. Han, Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL: <https://arxiv.org/abs/2503.09516>. arXiv:2503.09516.
- [6] H. Song, J. Jiang, Y. Min, J. Chen, Z. Chen, W. X. Zhao, L. Fang, J.-R. Wen, R1-searcher: Incentivizing the search capability in llms via reinforcement learning, 2025. URL: <https://arxiv.org/abs/2503.05592>. arXiv:2503.05592.
- [7] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, S. Petrov, Natural questions: A benchmark for question answering research, Transactions of the Association for Computational Linguistics 7 (2019) 452–466. URL: <https://aclanthology.org/Q19-1026/>.
- [8] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, C. D. Manning, 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, Brussels, Belgium, 2018, pp. 2369–2380. URL: <https://aclanthology.org/D18-1259/>.
- [9] A. Shtok, O. Kurland, D. Carmel, F. Raiber, G. Markovits, Predicting query performance by query-drift estimation, ACM Trans. Inf. Syst. 30 (2012). doi:10.1145/2180868.2180873.
- [10] M. Vlachou, C. Macdonald, Coherence-based query performance measures for dense retrieval, in: Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 15–24. URL: <https://doi.org/10.1145/3664190.3672518>.
- [11] S. Datta, D. Ganguly, M. Mitra, D. Greene, A relative information gain-based query performance prediction framework with generated query variants, ACM Trans. Inf. Syst. 41 (2022). URL: <https://doi.org/10.1145/3545112>.
- [12] C. Macdonald, J. Fang, A. Parry, Z. Meng, Constructing and Evaluating Declarative RAG Pipelines in PyTerrier, in: Proceedings of SIGIR 2025, 2025.
- [13] S. Farquhar, J. Kossen, L. Kuhn, Y. Gal, Detecting hallucinations in large language models using semantic entropy, Nature (London) 630 (2024) 625–630.
- [14] A. Dubey, et al., The Llama 3 herd of models, 2024. URL: <https://arxiv.org/abs/2407.21783>. arXiv:2407.21783.
- [15] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.
- [16] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu,

- M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, Z. Qiu, Qwen2.5 technical report, 2025. URL: <https://arxiv.org/abs/2412.15115>. arXiv:2412.15115.
- [17] G. Izacard, E. Grave, Leveraging passage retrieval with generative models for open domain question answering, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 874–880. URL: <https://aclanthology.org/2021.eacl-main.74/>.
- [18] N. Roy, L. F. R. Ribeiro, R. Blloshmi, K. Small, Learning when to retrieve, what to rewrite, and how to respond in conversational QA, in: Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 10604–10625. URL: <https://aclanthology.org/2024.findings-emnlp.622/>.
- [19] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, E. Grave, Atlas: few-shot learning with retrieval augmented language models, *J. Mach. Learn. Res.* 24 (2023).
- [20] M. R. Parvez, W. Ahmad, S. Chakraborty, B. Ray, K.-W. Chang, Retrieval augmented code generation and summarization, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2719–2734. URL: <https://aclanthology.org/2021.findings-emnlp.232/>.
- [21] M. Alaofi, N. Arabzadeh, C. L. A. Clarke, M. Sanderson, Generative Information Retrieval Evaluation, 2024. URL: <https://arxiv.org/abs/2404.08137>. arXiv:2404.08137.
- [22] A. Salemi, H. Zamani, Evaluating retrieval quality in retrieval-augmented generation, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 2395–2400. URL: <https://doi.org/10.1145/3626772.3657957>.
- [23] F. Tian, D. Ganguly, C. Macdonald, Is relevance propagated from retriever to generator in rag?, in: Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, 2025.
- [24] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang, Lost in the middle: How language models use long contexts, *Transactions of the Association for Computational Linguistics* 12 (2024) 157–173. doi:10.1162/tacl\_a\_00638.
- [25] F. Cuconasu, G. Trappolini, F. Siciliano, S. Filice, C. Campagnano, Y. Maarek, N. Tonello, F. Silvestri, The Power of Noise: Redefining Retrieval for RAG Systems, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 719–729. URL: <https://doi.org/10.1145/3626772.3657834>.
- [26] B. He, N. Chen, X. He, L. Yan, Z. Wei, J. Luo, Z.-H. Ling, Retrieving, rethinking and revising: The chain-of-verification can improve retrieval augmented generation, in: Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 10371–10393. URL: <https://aclanthology.org/2024.findings-emnlp.607/>.
- [27] N. Varshney, M. Luo, C. Baral, Can open-domain qa reader utilize external knowledge efficiently like humans?, 2022. URL: <https://arxiv.org/abs/2211.12707>. arXiv:2211.12707.
- [28] H. Trivedi, N. Balasubramanian, T. Khot, A. Sabharwal, Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 10014–10037.
- [29] J. Fang, Z. Meng, C. Macdonald, Kirag: Knowledge-driven iterative retriever for enhancing retrieval-augmented generation, arXiv preprint arXiv:2502.18397 (2025).
- [30] X. Li, G. Dong, J. Jin, Y. Zhang, Y. Zhou, Y. Zhu, P. Zhang, Z. Dou, Search-o1: Agentic search-enhanced large reasoning models, 2025. URL: <https://arxiv.org/abs/2501.05366>. arXiv:2501.05366.
- [31] Z. Wang, A. Liu, H. Lin, J. Li, X. Ma, Y. Liang, Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation, 2024. arXiv:arXiv:2403.05313.
- [32] A. Joshi, S. M. Sarwar, S. Varshney, S. Nag, S. Agrawal, J. Naik, Reaper: Reasoning based retrieval

- planning for complex rag systems, in: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 4621–4628. URL: <https://doi.org/10.1145/3627673.3680087>.
- [33] A. Singh, A. Ehtesham, S. Kumar, T. T. Khoei, Agentic retrieval-augmented generation: A survey on agentic rag, 2025. URL: <https://arxiv.org/abs/2501.09136>. arXiv: 2501.09136.
- [34] C. Hauff, D. Hiemstra, F. de Jong, A survey of pre-retrieval query performance predictors, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, 2008, p. 1419–1420. URL: <https://doi.org/10.1145/1458082.1458311>.
- [35] B. He, I. Ounis, Inferring query performance using pre-retrieval predictors, in: String Processing and Information Retrieval, 2004, pp. 43–54.
- [36] C. Meng, N. Arabzadeh, M. Aliannejadi, M. de Rijke, Query performance prediction: From ad-hoc to conversational search, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 2583–2593. URL: <https://doi.org/10.1145/3539618.3591919>.
- [37] Y. Tao, S. Wu, Query performance prediction by considering score magnitude and variance together, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 1891–1894. URL: <https://doi.org/10.1145/2661829.2661906>.
- [38] A. Shtok, O. Kurland, D. Carmel, Using statistical decision theory and relevance models for query-performance prediction, in: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, 2010, p. 259–266. doi:10.1145/1835449.1835494.
- [39] S. Cronen-Townsend, Y. Zhou, W. B. Croft, Predicting query performance, SIGIR '02, Association for Computing Machinery, New York, NY, USA, 2002, p. 299–306. URL: <https://doi.org/10.1145/564376.564429>.
- [40] G. Faggioli, N. Ferro, C. I. Muntean, R. Perego, N. Tonello, A geometric framework for query performance prediction in conversational search, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 1355–1365. URL: <https://doi.org/10.1145/3539618.3591625>.
- [41] N. Arabzadeh, M. Khodabakhsh, E. Bagheri, Bert-qp: Contextualized pre-trained transformers for query performance prediction, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 2857–2861. URL: <https://doi.org/10.1145/3459637.3482063>.
- [42] S. Datta, S. MacAvaney, D. Ganguly, 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, SIGIR '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 2148–2153. URL: <https://doi.org/10.1145/3477495.3531821>.
- [43] N. Arabzadeh, A. Bigdeli, M. Zihayat, E. Bagheri, Query performance prediction through retrieval coherency, in: Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2021, p. 193–200. URL: [https://doi.org/10.1007/978-3-030-72240-1\\_15](https://doi.org/10.1007/978-3-030-72240-1_15).
- [44] C. Macdonald, N. Tonello, S. MacAvaney, I. Ounis, PyTerrier: Declarative experimentation in python from bm25 to dense retrieval, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21, 2021, p. 4526–4533. doi:10.1145/3459637.3482013.
- [45] A. Chen, G. Stanovsky, S. Singh, M. Gardner, Evaluating question answering evaluation, in: Proceedings of the 2nd Workshop on Machine Reading for Question Answering, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 119–124. URL: <https://aclanthology.org/D19-5817/>.
- [46] X. Ho, A.-K. Duong Nguyen, S. Sugawara, A. Aizawa, Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps, in: Proceedings of the 28th International

Conference on Computational Linguistics, 2020, pp. 6609–6625. URL: <https://aclanthology.org/2020.coling-main.580/>.

- [47] R. F. Nogueira, Z. Jiang, R. Pradeep, J. Lin, Document ranking with a pretrained sequence-to-sequence model, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of *Findings of ACL*, Association for Computational Linguistics, 2020, pp. 708–718.
- [48] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, F. Wei, Text embeddings by weakly-supervised contrastive pre-training, arXiv preprint arXiv:2212.03533 (2022).
- [49] M. Vlachou, Predicting Retrieval Failures in Conversational Recommendation Systems, Ph.d. thesis, University of Glasgow, Glasgow, UK, 2025.

## Output - Iteration 1

<think>I need to determine if Chris Cornell sang the song “She’s Going to Change the World”. I’ll search for it.</think> <search> **chris cornell She’s Going to Change the World** </search>

<information> **Doc 1(Title: “Chris Cornell”)** called “Chris Cornell”. According to lead singer Shirley Manson, she was obsessed with Cornell at the time of the release of Soundgarden’s hit song “Black Hole Sun”. A demo version of “Fix Me Now” has Manson singing Cornell’s name. The line “I’m looking California, and feeling Minnesota” from Soundgarden’s song “Outshined” (written by Cornell), inspired the title of Steven Baigelman’s 1996 film “Feeling Minnesota”, starring Keanu Reeves and Cameron Diaz. Cornell’s song “Preaching the End of the World” from his debut solo album “Euphoria Morning”, inspired the title of Lorene Scafaria’s 2012 film “Seeking a Friend for the End of

**Doc 2(Title: “Chris Cornell”)** the World”, starring Steve Carell and Keira Knightley. In 1985, Cornell started dating Susan Silver, the manager of Alice in Chains, Soundgarden and Screaming Trees, and they got married in 1990. They had a daughter, Lillian Jean, born in June 2000. He and Silver divorced in 2004. In December 2008, Cornell reported via his official website that he had finally won back his collection of 15 guitars after a four-year court battle with Silver. In 2004, he married Vicky Karayiannis, a Paris-based American publicist of Greek heritage. The couple had a daughter together, Toni, in September 2004, and a son,

**Doc 3(Title: “Chris Cornell”)** Chris Cornell Christopher John Cornell (né Boyle; July 20, 1964 – May 18, 2017) was an American musician, singer and songwriter. He was best known as the lead vocalist for the rock bands Soundgarden and Audioslave. Cornell was also known for his numerous solo works and soundtrack contributions since 1991, and as the founder and frontman for Temple of the Dog, the one-off tribute band dedicated to his late friend Andrew Wood. Cornell is considered one of the chief architects of the 1990s grunge movement, and is well known for his extensive catalog as a songwriter, his nearly four-octave vocal </information>

## Output - Iteration 2

<think>I found out that Chris Cornell sang for the rock bands Soundgarden and Audioslave. Now I need to check if he sang the song “She’s Going to Change the World”. I’ll search for it.</think> <search> **She’s Going to Change the World** </search>

<information> **Doc 1(Title: “How to Change the World”)** It was very difficult to start a private organization at first, since Hungary was a communist country. So she started with a small group of disabled individuals and taught them simple skills that they could use to work a job, and that grew larger and changed over the years. It eventually became a functional and beautiful community where disabled individuals could come and take up a residency, learn a skill and make a living, and learn how to live independently if possible. There were many times where she failed, or her efforts seemed to be going nowhere. But she never

**Doc 2(Title: “How to Change the World”)** method, by stern discipline, by rigid attention to detail, by ceaseless labor, by the fixed determination of an indomitable will.” She knew she had the ability to make an extremely beneficial impact, and even though it reversed society’s norm, she succeeded in fulfilling her dreams. Florence Nightingale will forever be remembered as a great social entrepreneur for her heroic work in the field of nursing. Nightingale’s “indomitable will” may have been her most instrumental trait in accomplishing her radical ideas and plans. Beginning with strong resistance from her parents, and continuing to the point of defying army officers, her calling

**Doc 3(Title: “How to Change the World”)** was a constant struggle against society’s standards. Like Nightingale, every social entrepreneur will face opposition at some point in their career. It comes with the very definition of an entrepreneur. Their job is to defy cultural norms to find revolutionary, though often seemingly insane, ideas to solve the problems of life. “It takes concentrated focus, practical creativity, and a long-term source of energy to advance a system change and to ensure that the change becomes well rooted in institutions and cultures.” Social entrepreneurs must have the courage and indomitable will of Florence Nightingale to succeed in a world so resistant </information>

### Output - Iteration 3

<think>I found out that “She’s Going to Change the World” is a song, but I still need to check if Chris Cornell sang it. I’ll search for it.</think> <search> **who sang She’s Going to Change the World** </search>

<information>**Doc 1(Title: “Change the World”)** Change the World “Change the World” is a song written by Tommy Sims, Gordon Kennedy, and Wayne Kirkpatrick whose best-known version was recorded by the British recording artist Eric Clapton for the soundtrack of the 1996 film, “Phenomenon”. The track was produced by R&B record producer Kenneth “Babyface” Edmonds. The single release, Clapton recorded for Reprise and Warner Bros. Records, reached the Top 40 in twenty countries and topped the charts in Canada as well as “Billboard” magazine’s Adult Contemporary and Adult Top 40 charts in the United States. The single was prized with eight awards, among them three Grammy

**Doc 2(Title: “Gordon Kennedy (musician)”)** 20 years of friendship, Kennedy remains Frampton’s collaborator and performs with him often on tour. As of 2017, Kennedy has composed for or recorded with Bonnie Raitt (five songs, including “Gypsy in Me”), Reba McEntire, Tim McGraw, Wynonna, Charlie Daniels, Trisha Yearwood, George Strait, Faith Hill (“It Will Be Me”), Carrie Underwood (“The More Boys I Meet”), and others. The enduring worldwide hit song, “Change the World”, earned Kennedy a Grammy for “Song of the Year” in 1996. The song was first recorded in 1995 by Wynonna Judd and later Eric Clapton through the release of the John Travolta film,

**Doc 3(Title: “Change the World”)** the line “If I could change the world...”, leading to the second link, which is repeated three times, before the song’s outro, which is identical to the song’s intro and interlude. Before Clapton’s hit version was released, the song was recorded by country artist Wynonna Judd for her February 1996 album, “Revelations” (MCA Records). Her three-minute, nineteen-second take on the song is more of a neo-traditionalist country music track. Judd did not release her version as a single. Judd’s interpretation was also released on the compilation album “Best of America, Volume 2” for Curb Records on June 3, 2003. Eric </information>

### Output - Answer

<think>I found out that the song “She’s Going to Change the World” was sung by Eric Clapton. Now I can provide the answer.</think> <answer> **Eric Clapton** </answer>

**Figure 6:** The output from Search-R1 for the negative example “chris cornell she’s going to change the world”. In this example, the intermediate queries gradually drift away from the original information need, eventually leading to the wrong answer.

### Output - Iteration 1

<think>I need to find the last time Villanova won the NCAA basketball championship. I'll search for it.</think> <search> **when was the last time Villanova won the NCAA basketball championship** </search>

<information>**Doc 1(Title: "Villanova Wildcats men's basketball")** a team for the game. Their 2016 NCAA Championship, is referred to as "The Perfect Ending" and is the only NCAA Men's Championship game to be won on a buzzer beater, as Kris Jenkins drained a shot as time expired. They made the Final Four in 1939, 1971, 1985, 2009, 2016 and 2018; their six Final Four appearances are 13th most all-time. As of 2018, they have an NCAA Tournament record of 64–36 (). Villanova has defeated six No. 1 seeds in the NCAA tournament (Michigan and Georgetown in 1985, Pittsburgh in 2009, and Kansas and North Carolina in 2016,

**Doc 2(Title: "Villanova Wildcats men's basketball")** and Kansas in 2018), which is sixth most all-time. The Villanova Wildcats have appeared in the NCAA Tournament 37 times, the eighth highest total in NCAA history. They have won the Big East regular season championship eight times, most recently winning four straight from 2014 to 2017. They won the Big East Tournament in 1995, 2015, 2017, and 2018. Villanova entered the 2016–2017 season with an all-time winning percentage of (.648), placing the Wildcats tied for 13th among all NCAA Division I basketball programs. Through 2018, Villanova has 1,753 wins, which is 23rd among Division I men's basketball teams. Villanova

**Doc 3(Title: "Villanova Wildcats men's basketball")** and were the national champions in 1985, 2016, and 2018. Villanova has won as the underdog (based on Tournament seeding) 16 times, more than any other program. Villanova is one of only two programs (the other being Ohio State) that has played in the NCAA Tournament in every decade since the 1930s. "The NCAA began seeding the tournament with the 1979 edition." The Wildcats have appeared in the National Invitation Tournament (NIT) 17 times. Their combined record is 24–17. They were NIT Champions in 1994. The Wildcats appeared in the only National Campus Basketball Tournament. Their record is 0–1. </information>

### Output - Iteration 2

<think>I found out that Villanova won the NCAA basketball championship in 1985, 2016, and 2018. Now I need to find the last time they won the championship.</think> <search> **last time Villanova won the NCAA basketball championship** </search>

<information>**Doc 1(Title: "Villanova Wildcats men's basketball")** a team for the game. Their 2016 NCAA Championship, is referred to as "The Perfect Ending" and is the only NCAA Men's Championship game to be won on a buzzer beater, as Kris Jenkins drained a shot as time expired. They made the Final Four in 1939, 1971, 1985, 2009, 2016 and 2018; their six Final Four appearances are 13th most all-time. As of 2018, they have an NCAA Tournament record of 64–36 (). Villanova has defeated six No. 1 seeds in the NCAA tournament (Michigan and Georgetown in 1985, Pittsburgh in 2009, and Kansas and North Carolina in 2016,

**Doc 2(Title: "Villanova Wildcats men's basketball")** and Kansas in 2018), which is sixth most all-time. The Villanova Wildcats have appeared in the NCAA Tournament 37 times, the eighth highest total in NCAA history. They have won the Big East regular season championship eight times, most recently winning four straight from 2014 to 2017. They won the Big East Tournament in 1995, 2015, 2017, and 2018. Villanova entered the 2016–2017 season with an all-time winning percentage of (.648), placing the Wildcats tied for 13th among all NCAA Division I basketball programs. Through 2018, Villanova has 1,753 wins, which is 23rd among Division I men's basketball teams. Villanova

**Doc 3(Title: "Villanova Wildcats men's basketball")** and were the national champions in 1985, 2016, and 2018. Villanova has won as the underdog (based on Tournament seeding) 16 times, more than any other program. Villanova is one of only two programs (the other being Ohio State) that has played in the NCAA Tournament in every decade since the 1930s. "The NCAA began seeding the tournament with the 1979 edition." The Wildcats have appeared in the National Invitation Tournament (NIT) 17 times. Their combined record is 24–17. They were NIT Champions in 1994. The Wildcats appeared in the only National Campus Basketball Tournament. Their record is 0–1. </information>

### Output - Answer

<think>I found out that Villanova won the NCAA basketball championship in 1985, 2016, and 2018. The last time Villanova won the championship was in 2018.</think> <answer> **2018** </answer>

**Figure 7:** The output from Search-R1 for the positive example "when was the last time villanova won the ncaa basketball championship". In this example, through multiple retrievals based on reformulations of the original query, Search-R1 model reaches the correct answer.