# A Replicability Study of Metacognitive Retrieval-Augmented Generation

Gabriel Iturra-Bocaz
gabriel.e.iturrabocaz@uis.no
University of Stavanger
Stavanger, Norway

Petra Galuščáková
petra.galuscakova@uis.no
University of Stavanger
Stavanger, Norway

## Abstract

Recently, Retrieval Augmented Generation (RAG) has shifted focus to multi-retrieval approaches to tackle complex tasks such as multi-hop question answering. However, these systems struggle to decide when to stop searching once enough information has been gathered. To address this, Zhou et al. [52] introduced Metacognitive Retrieval Augmented Generation (MetaRAG), a framework inspired by metacognition that enables Large Language Models to critique and refine their reasoning. In this replicability paper, we replicate MetaRAG following its original experimental setup and extend it in two directions: (i) by evaluating the effect of PointWise and ListWise rerankers, and (ii) by comparing with SIM-RAG, which employs a lightweight critic model to stop retrieval. Our results confirm MetaRAG's relative improvements over standard RAG and reasoning-based baselines, but also reveal lower absolute scores than reported, reflecting challenges with closed-source LLM updates, missing implementation details, and unreleased prompts. We show that MetaRAG is partially replicated, gains substantially from reranking, and is more robust than SIM-RAG when extended with additional retrieval features.

## CCS Concepts

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

## Keywords

Retrieval Augmented Generation, Metacognition, Large Language Models

**ACM Reference Format:**
Gabriel Iturra-Bocaz and Petra Galuščáková. 2018. A Replicability Study of Metacognitive Retrieval-Augmented Generation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX


*Conference acronym 'XX, Woodstock, NY*

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

Metacognition is the ability to reflect on and critique one's own cognitive processes [19]. It enables individuals to notice gaps in understanding, judge whether additional information is needed, and adjust their strategies to reach more reliable conclusions. This self-awareness is fundamental in tasks that demand several reasoning steps across multiple pieces of information, allowing errors to be detected and planning strategies to be refined [19, 34]. Such ability is important because it helps to maintain reasoning that stays focused on the question rather than drifting to irrelevant details, and that remains consistent when integrating new or conflicting information [7]. However, although modern LLMs employ implicit control mechanisms such as routing or model selection, they typically generate responses without explicit self-reflection or awareness of their own knowledge limitations [23, 37].

Inspired by the concept of metacognition, Zhou et al. [52] proposed MetaRAG, a framework that enables RAG systems to monitor and critique their own reasoning before further retrieval is required. While other reasoning-oriented approaches such as CoT [44], ReAct [47], or Self-Ask [30] focus on guiding the reasoning process, they do not explicitly implement metacognitive control. This distinction remains important because multi-stage retrieval systems must regulate additional retrieval to avoid unnecessary overhead, latency, and cost, especially in large-scale and industrial settings. In this replicability paper, we replicate MetaRAG's results under a comparable setup and further investigate how aspects such as document ordering relate to metacognitive reasoning in LLMs through reranker models.

In this work, we formulate and address the following research questions:

- **RQ1**: How replicable are the results of MetaRAG?

While the original MetaRAG paper reports strong results on multi-hop Question-Answering (QA) tasks [25], its replicability remains uncertain [20]. For example, the code and prompts for the baseline implementations are not fully available, and the underlying LLMs have evolved since the study was published[1]. Moreover, some aspects of the original setup remain unclear, such as how sparse and dense retrieval are combined in the implementation and described in the paper [52]. Because MetaRAG represents an important step toward integrating metacognitive control into RAG systems, we aim to re-examine how it performs under current experimental conditions, including updated LLM backends, retrieval implementations, and incomplete or underspecified setup details.

- **RQ2**: Can reranker models improve the performance of MetaRAG?

[1] https://platform.openai.com/docs/deprecations#2023-11-06-chat-model-updates

In the human reasoning process for answering complex questions, we first prioritize information and then reflect on it [27]. Similarly, while MetaRAG critiques its own reasoning, it does not explicitly address the quality or ordering of retrieved documents. Since reranker models help reduce noise and control order [6, 26, 49], we investigate whether integrating them into MetaRAG enhances its overall reasoning and answer quality.

- **RQ3**: How does MetaRAG compare to other metacognitive RAG frameworks, such as SIM-RAG?

While MetaRAG investigates metacognition in multi-hop QA, the comparative impact of different metacognitive mechanisms on retrieval and reasoning is still not well understood. For example, Yang et al. [45] argue that multi-round retrieval RAG systems do not know when they have gathered enough information to answer a question. They propose SIM-RAG, a framework that adds a lightweight critic module which continuously checks if enough information has been retrieved, helping the system decide when to stop searching and start reasoning. We therefore compare MetaRAG with SIM-RAG to better understand how different forms of metacognition control influence retrieval stopping and reasoning performance.

**Main Contributions.** Our main contributions are summarized as follows: (1) we conduct a replicability study of the MetaRAG framework, focusing on evaluating its performance on QA tasks and comparing it with other baselines, (2) we extend MetaRAG with rerankers, finding performance gains by reducing noise and mitigating the effects of document reordering, and (3) we compare MetaRAG with SIM-RAG and show that MetaRAG, especially when combined with rerankers, performs more robustly. To support further research, we release our code[2] and experimental setup, including baselines.

## 2 Related Work

RAG has become a central paradigm for enhancing LLMs by injecting external information, which extends their internal knowledge and addresses common issues such as hallucinations and unverifiable answers [10, 21]. Early RAG systems employed single-retrieval strategies [8, 10, 21, 51] to answer simple queries, such as factual questions, but these approaches fall short for multi-hop QA tasks that require several reasoning steps to produce high-quality answers. To overcome these challenges, research has shifted towards multi-retrieval frameworks [17, 18, 40, 47], in which the retrieval stage is invoked iteratively during reasoning or generation. Generally, these approaches can be broadly divided into passive retrieval, decompositional calls, and dynamic schemes controlled by LLMs.

*Passive Retrieval.* In these approaches, retrieval is triggered on a fixed schedule (e.g., after every sentence or token count), regardless of whether the system needs to retrieve information at that moment. For example, Khandelwal et al. [17] enhance traditional language models by consulting a datastore of past contexts to recall similar contexts for informing the next predictions. Similarly, Trivedi et al. [40] introduced IR-CoT, which performs retrieval after every Chain of Thought (CoT) reasoning step.

*Decompositional Queries.* In these systems, complex queries are divided into sub-questions, each triggering a dedicated retrieval process, the results of which are later aggregated. Khot et al. [18] proposed that rather than solving a complex problem all at once, the original question should be decomposed into a sequence of simpler sub-queries, with each sub-question performing retrieval independently.

*Retrieval controlled by LLMs.* In these frameworks, retrieval is guided by the reasoning steps of LLMs. For example, ReAct [47] combines CoT reasoning to develop a series of thoughts that influence actions in an environment, including retrieval, to obtain more information at each specific step in the reasoning process. Furthermore, the synergy of these "reasoning" frameworks and iterative retrieval has enabled RAG systems to address more complex tasks, such as multi-hop QA.

Recent work on multi-hop QA has focused on improving reliability and control in RAG particularly in the presence of noisy retrieval and over-searching. Prompt-based strategies such as Self-Consistency [43] and Context Repetition (CoRe) [48], as well as fine-tuning approaches with external critics, aim to guide the reasoning process beyond a single retrieve–generate step. Among these approaches, SIM-RAG [45] introduces a lightweight classifier trained on synthetic data to determine whether additional retrieval is required. This design enables adaptive retrieval decisions but provides only an implicit form of self-monitoring, as the underlying reasoning errors or evidence deficiencies are not explicitly diagnosed.

MetaRAG [52] explicitly models metacognition through a structured *monitor–evaluate–plan* loop, allowing the system to identify reasoning issues, assess evidence sufficiency, and decide whether to continue retrieval. This explicit formulation places MetaRAG within a broader line of agentic and multi-round retrieval frameworks [1, 37, 41, 47], and makes it a particularly relevant case for studying the replicability of metacognitive control mechanisms under modern IR systems.

## 3 Metacognitive RAG Frameworks

To contextualize our study, we first describe the general structure of metacognitive RAG systems [52]. These frameworks extend the standard *retrieve-then-generate* [21] paradigm by introducing a metacognitive loop that enables the model to reflect on its own reasoning process before finalizing an answer.

Figure 1 illustrates this general metacognitive architecture, which operates across two complementary spaces:

- **Cognitive Space (CS).** This space corresponds to the standard RAG workflow. A *Retriever Module* fetches relevant documents from an external knowledge base given a question, which are then combined with the question and passed to *CS Reasoner Module*. This module performs the initial reasoning step, interpreting the evidence and generating a candidate answer along with its underlying rationale.
- **Metacognitive Space (MS).** In this space, a *Critic Module* examines the candidate answer produced by *CS Reasoner Module*, assessing its coherence, confidence, and completeness. If the rationale is insufficient, the critic invokes *MS Reasoner Module* to refine the reasoning process, which may involve rewriting the question, requesting additional

[2] https://anonymous.4open.science/r/sigir2026-replicability-4B48/

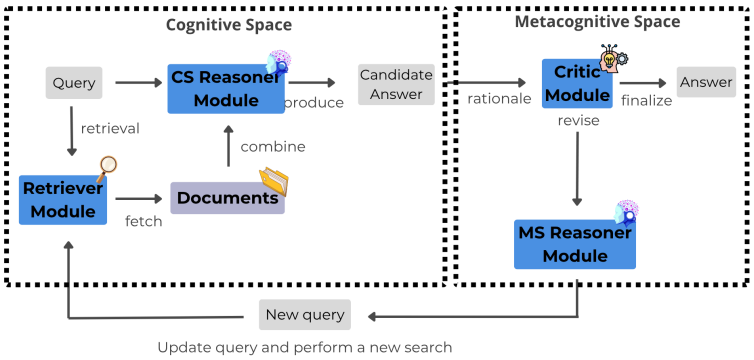


**Figure 1: General architecture of metacognitive RAG systems. In the Cognitive Space (CS), the *Retriever Module* fetches relevant documents based on the question, which are then integrated by the *CS Reasoner Module* to produce a candidate answer. In the Metacognitive Space (MS), the *Critic Module* evaluates this rationale and, if necessary, delegates to the *MS Reasoner Module* to refine the reasoning or reformulate a new question for additional retrieval.**

retrieval, or revising the generated answer before finalization.

This general pipeline serves as the foundation for the specific frameworks analyzed in this study.

### 3.1 MetaRAG

MetaRAG implements metacognitive control through an explicit *monitor–evaluate–plan* loop in the MS, which determines whether the current answer should be accepted, refined, or supported by additional retrieval.

- **Monitoring.** Given a question $q$ and retrieved passages $D_q$, the CS produces an answer $y$, while an "expert" QA model generates a reference answer $y'$. A similarity score between $y$ and $y'$ is computed, and metacognitive evaluation is triggered only when this score falls below a predefined threshold; otherwise, the answer is accepted directly.
- **Evaluation.** When it is activated, the Critic module diagnoses why the current answer may be unreliable. Internal knowledge sufficiency is assessed by prompting the LLM, while external knowledge sufficiency is evaluated using an NLI model that verifies whether the retrieved passages entail the required information. In addition, the Critic Module checks for common reasoning errors such as incomplete or inconsistent use of evidence.
- **Planning.** Based on the evaluation outcome, MetaRAG selects the next action. If knowledge is insufficient, the system reformulates the question by explicitly targeting missing information and retrieves additional documents. If knowledge is sufficient but the reasoning is flawed, the system revises unsupported reasoning steps or adjusts the reliance on internal versus external knowledge. This loop repeats until the answer is accepted or a maximum number of iterations is reached.

### 3.2 SIM-RAG

Instead of employing an explicit monitor–evaluate–plan loop as in MetaRAG, SIM-RAG [45] introduces a lightweight critic module that continuously judges whether enough evidence has been retrieved to answer a question. This critic is trained to signal when retrieval should stop, thereby preventing unnecessary or noisy retrieval rounds. While MetaRAG emphasizes explicit self-reflection and reasoning refinement, SIM-RAG focuses on efficient retrieval control through implicit metacognitive signals. In this study, we do not aim to replicate SIM-RAG, but include it as a comparative metacognitive framework to better contextualize the strengths and weaknesses of MetaRAG under similar experimental conditions.

## 4 Experimental setting

### 4.1 Datasets and Evaluation Metrics

Following the original paper, we conduct experiments on two multi-hop QA collections: HotpotQA [46] and 2WikiMultiHopQA [39]. HotpotQA contains over 100K Wikipedia-based questions that often require reasoning across two or more passages, while 2WikiMultiHopQA provides 193K questions explicitly designed to enforce multi-hop reasoning across multiple documents. Both datasets are open-domain and rely on the Wikipedia corpus.

To remain consistent with the original MetaRAG configuration, we evaluate all methods on 500 examples sampled from the development (dev) set of each collection. These samples are publicly available in our repository[3]. This evaluation protocol follows common practice in prior RAG works [11, 12, 22, 45, 52], where full-dataset evaluation is often infeasible due to computational constraints. Since the original MetaRAG code does not release the exact development subsets used, we randomly sample 500 examples from each dev set. We further adopt the same evaluation metrics as the original paper to assess the performance of MetaRAG and all other frameworks in this paper. At the answer level, we use exact match (EM) [2], which counts a prediction as correct only if

[3]https://anonymous.4open.science/r/sigir2026-replicability-4B48/data/README.md

it matches the gold answer string exactly, with no extra or missing words. At the token level, we use token-level F1, Precision (Prec.), and Recall (Rec.).

## 4.2 Baselines and SIM-RAG

The original MetaRAG paper does not release the code or prompts for its baseline methods, nor does it describe how these baselines are adapted for a fair comparison with MetaRAG. In particular, it remains unclear how each reasoning frameworks [35] is integrated with the retrieval component. For example, ReAct [47] interleaves reasoning steps with retrieval actions, but the MetaRAG paper does not specify how this action interface interacts with the index or how it is managed across iterations. Similarly, the prompting setup used in these baselines is not documented, even though performance in reasoning-oriented methods strongly depends on prompt design [35]. To ensure a consistent and transparent comparison, we re-implement all baselines in LangChain[4], explicitly defining how each method interacts with the retriever and adapting their prompts based on the descriptions in their respective papers. All baseline methods are evaluated using a fixed few-shot prompting setup to reduce prompt sensitivity.

The baselines we include are: *Standard Prompting*, where the LLM generates answers without retrieval; *Standard RAG* [21], which concatenates a fixed set of retrieved passages with the question; *CoT* [44], which guides the model to produce explicit reasoning steps; *ReAct* [47], which interleaves reasoning steps with external actions; *Self-Ask* [30], which decomposes complex queries into sub-questions; *IR-CoT* [40], which interleaves retrieval with CoT reasoning to incrementally gather evidence during multi-hop inference; *FLARE* [12], which adaptively triggers retrieval based on the model's uncertainty during generation; and *Reflexion* [37], which enables iterative self-critique and refinement of answers. In addition to the baselines, we include *SIM-RAG* [45] not as a baseline but as a comparative framework, since it represents a RAG system that explicitly integrates metacognitive principles. SIM-RAG was introduced at SIGIR 2025, and we use the official implementation released by the authors, with minor adaptations to ensure a fair comparison with our MetaRAG setup.

## 4.3 Experimental Setup

We follow the experimental protocol described in the MetaRAG paper, with some adaptations to ensure clarity and replicability. This section details the LLMs used, retrieval and metacognitive configurations, reranking strategies, and the comparison with SIM-RAG.

*4.3.1 Language Models.* We use `gpt-3.5-turbo-16k`[5] (GPT-3.5) as the LLM, accessed through the OpenAI API[6]. The same LLM serves as the Reasoner model in both the CS and the MS. This corresponds to the closed-source model adopted in the original MetaRAG paper. Additionally, to assess generalization to open-source LLMs, we use Llama-3.3:70B[7] (Llama3.3) as the Reasoner model across both spaces. All Llama3.3 experiments are run on a server equipped with four NVIDIA A100 SXM4 GPUs (40 GB VRAM each) and 514 GB of system memory, enabling efficient inference in a quantized setup. Both LLMs are configured with a temperature of 0.

*4.3.2 Retrieval Setup.* The original MetaRAG paper describes a hybrid retrieval setup that combines BM25 [33] as a sparse retriever with E5 [42] as a dense retriever. However, neither the paper nor the released codebase provides sufficient implementation details to fully replicate how these two retrieval signals are combined. To remain faithful to the intended hybrid design while ensuring replicability, we implement sparse–dense fusion from scratch using Reciprocal Rank Fusion (RRF) [4]. Sparse retrieval is performed using BM25 implemented in Lucene via Pyserini[8], while dense retrieval relies on E5 embeddings indexed with FAISS[9]. Both retrievers are built over the same Wikipedia dump [16], which serves as the shared document collection for indexing. For each question, we independently retrieve the top 100 documents from the sparse and dense retrievers and merge the resulting ranked lists using RRF. After fusion, we retain the top five passages, which are passed to the answer generation and downstream reasoning components.

While the original SIM-RAG [45] implementation employs a custom retriever and corpus built with ElasticSearch[10], we replace this component with our Pyserini–FAISS hybrid retrieval pipeline. This modification guarantees that both MetaRAG and SIM-RAG operate under identical retrieval conditions.

*4.3.3 Metacognitive Space Configuration.* In the MS, we replicate the configuration described in the MetaRAG paper. A fine-tuned T5-large[11] model is used as the expert for the monitoring stage. The judgment threshold is set to 0.4, which triggers the evaluation phase and allows a maximum of five iterations per question. For the evaluation and planning stages from the MetaRAG framework, we employ the same NLI T5-XXL[12] model used in the original study. This configuration enables MetaRAG to assess evidence sufficiency, detect reasoning errors, and decide whether additional retrieval or reasoning refinement is required.

*4.3.4 Reranking Strategies.* To analyze the impact of reranking on MetaRAG, we evaluate two reranking configurations: *PointWise* and *ListWise* rerankers. Reranking is applied after each retrieval round, and the top five reranked documents are concatenated and passed to the LLM for answer generation. In the PointWise setting, each document–question pair is scored independently by the reranker, and the top-$k$ passages are selected based on their individual relevance scores. We evaluate three widely used PointWise rerankers: *MiniLM*[13], *BGE*[14] [3], and *ModernBERT*[15] [24, 50]. In contrast, the ListWise configuration jointly evaluates the full set of retrieved passages, allowing the model to consider inter-document relationships during ranking. For this setting, we employ *RankGPT* [38], *Zephyr* [29], and *Vicuna* [36], implemented through the RankLLM[16] library

[4] https://www.langchain.com/
[5] https://platform.openai.com/docs/models
[6] https://platform.openai.com/docs/overview
[7] https://ollama.com/library/llama3.3:70b
[8] https://github.com/castorini/pyserini
[9] https://github.com/facebookresearch/faiss
[10] https://github.com/elastic/elasticsearch
[11] https://huggingface.co/gaussalgo/T5-LM-Large_Canard-HotpotQA-rephrase
[12] https://huggingface.co/google/t5_xxl_true_nli_mixture
[13] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L12-v2
[14] https://huggingface.co/BAAI/bge-reranker-v2-m3
[15] https://huggingface.co/Alibaba-NLP/gte-reranker-modernbert-base
[16] https://github.com/castorini/rank_llm

[36], which provides ListWise reranking interfaces for LLMs. In our setup, RankGPT uses `gpt4o_mini`[17] as the underlying LLM for ListWise reranking. We understand that ListWise reranking introduces additional computational cost compared to PointWise or retrieval-only setups, particularly for RankGPT, which relies on a proprietary LLM. However, since our goal is to analyze the impact of reranking on MetaRAG rather than optimize for efficiency, we do not prioritize computational cost in this setup.

*4.3.5 SIM-RAG Critic Model.* For the SIM-RAG critic, we use *SIM-RAG-Llama3-2B*[18], a 2B-parameter Llama 3[19] model adapted for general-purpose inference. It is fine tuned on several multi-hop QA benchmarks, including TriviaQA [14], HotpotQA [46], 2WikiMultiHopQA [9], PopQA [31], and Musique [40].

## 5 Results and Discussion

This section presents our replication results for MetaRAG and discusses how they differ from the original study. We then examine diagnostic signals, threshold sensitivity, the impact of rerankers, and a comparison between MetaRAG and SIM-RAG in terms of accuracy and efficiency.

### 5.1 Replicability of MetaRAG

To answer **RQ1**, we first investigate to what extent the results of MetaRAG can be replicated under a setup that closely follows the configuration reported in the original paper. Our replicated results show that, although absolute performance is generally lower than that reported in the original study, the relative trends are preserved. In particular, MetaRAG consistently outperforms all baseline methods across both HotpotQA and 2WikiMultiHopQA, including self-critique approaches such as Reflexion, as it is depicted in Table 1. This indicates that the core monitor–evaluate–plan mechanism proposed in MetaRAG remains effective under replicated conditions. However, we observe some differences in absolute performance, particularly on 2WikiMultiHopQA, where the gap between the original and replicated results is larger than on HotpotQA. Additionally, several baselines also achieve higher scores than originally reported, which may reflect changes in the underlying GPT-3.5 model over time due to deprecations and improvements[20], introducing additional variability that hinders replicability with proprietary LLMs. We observe that Self-Ask performs worse than Standard RAG in our replicated GPT-3.5 experiments, which differs from the results reported in prior work. Rather than indicating an implementation error, the discrepancy may stem from the fact that the Self-Ask baseline prompt used in the original MetaRAG experiments is not publicly released. Consequently, we attempt to approximate the prompt based on the description provided in the Self-Ask paper [30]. Since LLMs are known to be highly sensitive to even small changes in prompt wording and formatting [20, 32, 35], such differences can influence how sub-questions are generated and how retrieval is triggered, ultimately affecting final answer quality.

[17] https://platform.openai.com/docs/models/gpt-4o-mini
[18] https://huggingface.co/dyang39/SIM-RAG-Llama3-2B
[19] https://huggingface.co/andrijdavid/Llama3-2B-Base
[20] https://platform.openai.com/docs/deprecations#2023-11-06-chat-model-updates

### 5.2 Diagnostic Analysis

The difference between the original and replicated results, particularly on 2WikiMultiHopQA, motivates a deeper diagnostic analysis. We examine MetaRAG's evaluation signals on HotpotQA and 2WikiMultiHopQA, including: (i) an LLM-based assessment of whether a question can be answered using internal knowledge alone, (ii) an auxiliary LLM-based judgment of whether the retrieved documents are sufficient, and (iii) the NLI critic used to verify whether the retrieved documents entail the generated answer. Table 2 highlights clear differences between the two datasets. While HotpotQA contains a noticeable portion of questions answerable using internal knowledge, 2WikiMultiHopQA is almost entirely retrieval-dependent. Moreover, when retrieval is required, the LLM more often judges the retrieved documents sufficient on HotpotQA than on 2WikiMultiHopQA, indicating that the initial top-$k$ retrieval documents more frequently fails to provide clearly sufficient evidence on 2WikiMultiHopQA.

In contrast, the NLI critic validates a larger fraction of answers than the LLM's sufficiency judgment on both datasets. This suggests that relevant evidence is frequently present in the retrieved context but not easily recognized as sufficient prior to multi-hop reasoning by LLMs. The effect is particularly pronounced on 2WikiMultiHopQA, where many questions are not clearly solvable with the current evidence, so the system hesitates between stopping and continuing the reasoning process.

### 5.3 Sensitivity to the Metacognitive Threshold

Our diagnostic analysis in Table 2 shows that nearly all questions in both datasets require external retrieval, with a notably larger gap on 2WikiMultiHopQA. Since MetaRAG relies on a confidence threshold to decide when to transition into the MS, this parameter controls how often additional retrieval and refinement are triggered. Given the high proportion of retrieval-dependent cases, this control mechanism may substantially affect performance. To examine its impact, we vary the threshold on the same 500-question subset of 2WikiMultiHopQA using GPT-3.5, while keeping all other components fixed. The results are reported in Table 3. We find that MetaRAG is highly sensitive to this parameter, as lower thresholds reduce transitions into the MS and additional retrieval calls, while improving answer quality. In the original MetaRAG paper, the threshold is set to 0.4, which yields the best results under their setup. In contrast, our replication, answer quality (e.g., EM/F1) peaks at a threshold of 0.2, whereas further lowering the threshold to 0.1 results in decreased performance, even though it reduces retrieval and generation calls. One possible explanation is that GPT-3.5 has evolved[21] since the original study, potentially strengthening its internal knowledge. As a result, the original threshold configuration may no longer be optimal, and repeatedly triggering metacognitive control may provide limited additional benefit.

Notably, this analysis does not contradict our earlier diagnostic findings in Table 2. Although the LLM frequently judges the retrieved context as insufficient, the NLI critic validates a much larger portion of answers, indicating that the necessary evidence is often already present. The bottleneck therefore appears to lie in how evidence is connected across hops rather than in its availability.

[21] https://platform.openai.com/docs/deprecations#2023-11-06-chat-model-updates

| Data | Method | Retr. | Multi. | Critic | Original | | | | Replicated | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | EM | F1 | Prec. | Rec. | EM | F1 | Prec. | Rec. |
| HotpotQA | Standard Prompting | × | × | × | 20.0 | 25.8 | 26.4 | 28.9 | 26.4* | 36.6* | 39.4* | 35.2* |
| | Chain of Thought | × | × | × | 22.4 | 34.2 | 33.9 | 46.0 | 26.6* | 36.7* | 39.3* | 38.3* |
| | Standard RAG | ✓ | × | × | 24.6 | 33.0 | 34.1 | 34.5 | 30.8* | 42.0* | 45.1* | 42.2* |
| | ReAct | ✓ | ✓ | × | 24.8 | 41.7 | 42.6 | 44.7 | 31.8* | 42.4* | 46.8* | 41.3* |
| | Self-Ask | ✓ | ✓ | × | 28.2 | 43.1 | 43.4 | 44.8 | 28.6* | 39.2* | 41.2* | 43.1* |
| | FLARE | ✓ | ✓ | × | 29.2 | 42.4 | 42.8 | 43.0 | 33.6* | 44.7* | 48.4* | 44.0* |
| | IRCoT | ✓ | ✓ | × | 31.4 | 40.3 | 41.6 | 41.2 | 32.8* | 47.4* | 51.7* | 43.5* |
| | Reflexion | ✓ | ✓ | ✓ | 30.0 | 43.4 | 43.2 | 44.3 | 30.8* | 41.3* | 43.5* | 41.7* |
| | MetaRAG | ✓ | ✓ | ✓ | **37.8** | **49.9** | **52.1** | **50.9** | **34.2** | **48.2** | **49.5** | **48.6** |
| 2WikiMultiHopQA | Standard Prompting | × | × | × | 21.6 | 25.7 | 24.5 | 31.8 | 25.6* | 29.4* | 29.9* | 29.4* |
| | Chain of Thought | × | × | × | 27.6 | 37.4 | 35.8 | 44.3 | 25.8* | 31.2* | 31.7* | 32.0* |
| | Standard RAG | ✓ | × | × | 18.8 | 25.3 | 25.6 | 26.2 | 28.9* | 33.5* | 34.5* | 33.6* |
| | ReAct | ✓ | ✓ | × | 21.0 | 28.0 | 27.6 | 30.0 | 31.0* | 38.8* | 38.6 | 39.3* |
| | Self-Ask | ✓ | ✓ | × | 28.6 | 37.5 | 36.5 | 42.8 | 27.2* | 34.6* | 35.8* | 34.9* |
| | FLARE | ✓ | ✓ | × | 28.2 | 39.8 | 40.0 | 40.8 | 29.8* | **40.4** | 41.2* | **40.8** |
| | IRCoT | ✓ | ✓ | × | 30.8 | 42.6 | 42.3 | 40.9 | 30.6* | 39.9 | **43.1** | 38.2* |
| | Reflexion | ✓ | ✓ | ✓ | 31.8 | 41.7 | 40.6 | 44.2 | 31.2* | 36.8* | 35.9* | 37.8* |
| | MetaRAG | ✓ | ✓ | ✓ | **42.8** | **50.8** | **50.7** | **52.2** | **33.4** | 38.9 | 37.1 | 39.4 |

**Table 1: Evaluation results with retrieval (Retr.), multi-round retrieval (Multi.), and critic (Critic). Baselines include Standard Prompting, Standard RAG, CoT, ReAct, Self-Ask, FLARE, IRCoT, and Reflexion. Original (reported in MetaRAG) vs. Replicated (ours) results. Best scores are in bold. An asterisk (*) denotes a statistically significant difference from MetaRAG ($p < 0.05$).**

| Evaluation Condition | HotpotQA | 2WikiMultiHopQA |
|---|---|---|
| *All questions (LLM Judge)* | | |
| Answered using internal knowledge | 20.4% | 1.2% |
| Requires external retrieval | 79.6% | 98.8% |
| *Conditional on cases that require external retrieval (LLM Judge)* | | |
| Answered using retrieved documents | 63.3% | 33.6% |
| Rejected despite retrieved documents | 36.7% | 66.4% |
| *Conditional on cases that require external retrieval (NLI critic)* | | |
| Validated by critic model | 86.7% | 72.9% |
| Rejected by critic model | 13.3% | 27.1% |

**Table 2: Diagnostic breakdown of MetaRAG evaluation signals on 500-question subsets. The first block is computed over all 500 questions. The second and third blocks focus on questions the LLM marked as requiring retrieval: the second reports the LLM's own assessment of retrieval sufficiency, and the third shows whether the NLI critic determines that the retrieved documents entail the generated answer (entailment = validate, otherwise = reject).**

Therefore, increasing retrieval iterations alone does not necessarily improve reasoning, and may instead complicate the detection of relevant information.

For all subsequent experiments, we fix the confidence threshold at 0.4 in order to isolate the effects of reranking improvements over retrieval-only set up.

| Threshold | EM | F1 | MS trans./q | Retr./q | Iters./q |
|---|---|---|---|---|---|
| 0.4 | 33.4 | 38.9 | 3.2 | 2.8 | 4.2 |
| 0.3 | 37.8 | 45.6 | 2.5 | 2.1 | 3.5 |
| 0.2 | 38.1 | 46.2 | 1.5 | 1.2 | 2.5 |
| 0.1 | 35.4 | 42.6 | 1.3 | 1.1 | 2.2 |

**Table 3: Sensitivity to the confidence threshold on 2WikiMultiHopQA (500 samples; GPT-3.5; max_iter=5; top-$k$=5). The parameter max_iter denotes the maximum number of MetaRAG reasoning iterations permitted per question, whereas top-$k$ defines the number of documents retrieved at each retriever call. The columns MS trans./q, Retr./q, and Iters./q denote the average metacognitive transitions, retrieval calls, and reasoning iterations per question, respectively. Results are reported as Exact Match (EM) and F1.**

## 5.4 Enhancing MetaRAG with Rerankers

To answer **RQ2**, we examine whether reranker models improve MetaRAG's performance by promoting more relevant documents within the top-$k$ context used for reasoning and critique. As shown in Table 4, reranking generally improves performance on both collections; however, its effectiveness strongly depends on the specific reranker configuration. Overall, BGE delivers the strongest gains among PointWise methods, while RankGPT performs best among ListWise approaches. Moreover, reranking yields larger improvements with Llama3.3 than with GPT-3.5, suggesting that reranking can improve document ordering, although its effectiveness may rely on the LLM's ability to connect reasoning hops across the reordered evidence and generate an accurate answer from it.

Reranking is not uniformly beneficial and, in certain cases, performs worse than the retrieval-only setup. For example, MiniLM reduces performance on HotpotQA when paired with Llama3.3 and similarly underperforms on both collections when used with GPT-3.5. This pattern is consistent with limitations of PointWise reranking in multi-hop settings. PointWise rerankers evaluate each document independently with respect to the question, which may be suboptimal when answering requires combining evidence across multiple passages. MiniLM is trained on the MS MARCO passage collection [28], which primarily contains single-hop question–passage pairs. As a result, it may prioritize documents that are individually similar to the question rather than those that collectively support multi-hop reasoning across multiple sources. In contrast, BGE is trained on a larger and more diverse mixture of retrieval and QA-oriented datasets [3], which enables it to better capture deeper semantic relevance. Similarly, RankGPT's strong performance among ListWise approaches may stem from its ability to consider all candidate documents jointly when producing a ranking. Unlike PointWise methods, ListWise rerankers model interactions between passages, which is particularly advantageous in multi-hop question answering where evidence is combined across several documents.

## 5.5 Comparison with SIM-RAG

*5.5.1 Performance Comparison.* In **RQ3**, we compare MetaRAG against SIM-RAG to study its performance relative to other metacognitive frameworks. We evaluate both under identical retrieval and reranking conditions, as shown in Table 5. Our findings show that MetaRAG performs significantly better than SIM-RAG when supported by reranker models. For this comparison, we select the best PointWise and ListWise settings from the previous experiment, namely BGE and RankGPT, respectively. The results reveal a clear pattern across both GPT3.5 and Llama3.3: while SIM-RAG performs reasonably well in retrieval-only settings, its performance drops once rerankers are introduced, especially with Llama3.3, though the trend is also present for GPT3.5. This may be because SIM-RAG's fine-tuned critic is tailored to a fixed retrieval setup, and rerankers modify the input distribution, disrupting the conditions under which it operates effectively. In contrast, MetaRAG conducts self-critique through prompting rather than fine-tuning, making it more flexible and less sensitive to such distributional changes, and thus better able to adapt to varying retrieval conditions. Paired t-tests are conducted between MetaRAG and SIM-RAG. Several settings show statistically significant improvements in favor of MetaRAG ($p < 0.05$), as indicated by the asterisks in Table 5, while other configurations are favor SIM-RAG.

*5.5.2 Efficiency and Computational Cost.* Beyond accuracy, we compare MetaRAG and SIM-RAG in terms of computational cost under the retrieval-only setup. As shown in Table 6 (a) and Table 6 (b), MetaRAG incurs substantially higher overhead across both LLMs. It requires more model calls per question and exhibits significantly higher token usage and latency than SIM-RAG, resulting in a markedly higher overall cost in the GPT-3.5 setting. Similar trends are observed with Llama3.3, where MetaRAG's iterative control mechanism leads to increased request counts and slower inference. Overall, the results highlight a clear efficiency gap between the two frameworks. However, while MetaRAG requires considerably greater computational resources to reach high accuracy, it remains training-free and broadly applicable across datasets, whereas SIM-RAG achieves faster and more resource-efficient inference but depends on dataset-specific critic fine-tuning, potentially constraining its cross-domain generalization despite its efficiency gains

## 6 Replicability Challenges

We identify several challenges that explain the discrepancies between our results and those reported in the original paper.

First, the use of closed-source LLMs, such as those from the OpenAI family, hinders replicability because these models are continuously updated to provide better answers for users and, consequently, do not produce identical outputs over time [20]. For instance, the model GPT3.5 undergoes several updates after its release[22], which makes exact numerical replication impossible even when following the same methodology as the original authors.

Second, the original paper describes a hybrid retrieval strategy combining sparse (BM25) and dense (E5) methods, but it does not specify how these signals are integrated, nor is this in any detail

[22]https://platform.openai.com/docs/deprecations#2023-11-06-chat-model-updates

| LLM | Setting | Reranker | HotpotQA | | | | 2WikiMultiHopQA | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | EM | F1 | Prec. | Rec. | EM | F1 | Prec. | Rec. |
| GPT-3.5 | Ret | – | 34.2 | 48.2 | 49.5 | 48.6 | 33.4 | 38.9 | 37.1 | 39.4 |
| | Ret+PointWise | MiniLM | 34.1 | 48.4 | 50.1 | 47.3 | 31.9 | 36.8 | 37.5* | 36.3 |
| | | BGE | 38.8* | 51.3* | 53.3* | 51.5* | 36.3* | 42.1* | 43.1* | 42.4* |
| | | ModernBERT | 35.7* | 48.7* | 52.4* | 49.6* | 30.9 | 35.1 | 36.7 | 36.4 |
| | Ret+ListWise | RankGPT | **40.1*** | **53.5*** | **55.5*** | **54.2*** | **38.9*** | **44.3*** | **45.4*** | **43.6*** |
| | | Zephyr | 36.3* | 46.0 | 47.2 | 49.6* | 32.0 | 34.0 | 32.9 | 33.8 |
| | | Vicuna | 35.8* | 48.6* | 51.4* | 46.5 | 33.9* | 40.8* | 40.1* | 41.9* |
| Llama3.3 | Ret | – | 41.5 | 54.1 | 57.9 | 55.6 | 29.0 | 33.3 | 31.3 | 32.9 |
| | Ret+PointWise | MiniLM | 40.4 | 51.3 | 56.1 | 52.1 | 33.7* | 40.1* | 41.5* | 42.1* |
| | | BGE | 43.6* | 56.1* | 61.3* | 56.1 | 35.5* | 41.3* | 42.1* | 42.9* |
| | | ModernBERT | 41.8 | 54.7* | 59.4* | 53.9 | 33.2* | 36.9* | 37.9* | 36.8* |
| | Ret+ListWise | RankGPT | **46.7*** | **59.3*** | **63.1*** | **57.1*** | **38.9*** | **45.5*** | **48.6*** | **46.7*** |
| | | Zephyr | 44.4* | 58.8* | 62.1* | 56.3 | 33.7* | 37.9* | 35.4* | 35.2* |
| | | Vicuna | 41.2 | 53.5 | 58.9 | 51.4 | 36.2* | 40.8* | 39.2* | 38.1* |

**Table 4: Comparison of MetaRAG performance under retrieval-only (Ret), PointWise, and ListWise reranking settings across different LLMs and datasets. Results are reported as EM, F1, Precision (Prec.), and Recall (Rec.). Best scores are in bold. An asterisk (*) denotes a statistically significant improvement ($p < 0.05$) over the corresponding retrieval-only baseline within the same LLM.**

| Method | LLM | Setting | Reranker | HotpotQA | | | | 2WikiMultiHopQA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | EM | F1 | Prec. | Rec. | EM | F1 | Prec. | Rec. |
| MetaRAG | GPT-3.5 | Ret | – | 34.2 | 48.2 | 49.5 | 48.6 | 33.4* | 38.9* | 37.1* | 39.4* |
| | | Ret+PointWise | BGE | 38.8 | 51.3* | 53.3* | 51.5* | **36.3** | **42.1** | **43.1** | **42.4** |
| | | Ret+ListWise | RankGPT | **40.1** | **53.5*** | **55.5*** | **54.2*** | 35.9* | 41.3* | 42.4* | 40.6* |
| | Llama3.3 | Ret | – | 41.5 | 54.1* | 57.9* | 55.6* | 29.0 | 33.3 | 31.3 | 32.9 |
| | | Ret+PointWise | BGE | 43.6* | 56.1* | 61.3* | 56.1* | 35.5* | 41.3* | 42.1* | 42.9* |
| | | Ret+ListWise | RankGPT | **46.7*** | **59.3*** | **63.1*** | **57.1*** | **38.9*** | **45.5*** | **48.6*** | **46.7*** |
| SIM-RAG | GPT-3.5 | Ret | – | **42.2** | **53.1** | **55.4** | **54.9** | 30.8 | 34.4 | 35.3 | 33.7 |
| | | Ret+PointWise | BGE | 41.6 | 44.3 | 46.3 | 45.5 | **36.3** | **45.6** | **49.2** | **47.3** |
| | | Ret+ListWise | RankGPT | 41.2 | 47.5 | 49.5 | 50.1 | 23.5 | 25.3 | 29.4 | 27.4 |
| | Llama3.3 | Ret | – | **43.1** | **60.9** | **61.3** | **58.4** | **38.3** | **43.3** | **45.6** | **43.2** |
| | | Ret+PointWise | BGE | 21.4 | 25.0 | 26.4 | 26.3 | 14.5 | 15.1 | 15.6 | 14.9 |
| | | Ret+ListWise | RankGPT | 18.7 | 22.3 | 23.8 | 22.5 | 12.8 | 16.5 | 19.3 | 18.9 |

**Table 5: Comparison of MetaRAG and SIM-RAG performance across retrieval-only (Ret), PointWise, and ListWise reranking settings. The *Reranker* column specifies the reranking model used. Results are reported as Exact Match (EM), F1, Precision (Prec.), and Recall (Rec.). Best results are in bold. An asterisk (*) indicates MetaRAG significantly outperforms SIM-RAG under the same LLM and setting ($p < 0.05$).**

reflected in the released code. In our implementation, we explicitly construct both components: we use BM25 via Pyserini[23] and build a dense retriever by encoding the Wikipedia dump with E5 embeddings and indexing them using FAISS. The two retrieval outputs are fused using RRF. Since these implementation details are absent

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

**(a) GPT-3.5**

| Method | Req./question | Tok./req. | Lat./req. (s) | Cost (USD) |
|---|---|---|---|---|
| RAG | 1.0 | 870.0 | 0.5 | 1.3 |
| MetaRAG | 14.85 | 30,064.5 | 47.85 | 37.66 |
| SIM-RAG | 6.5 | 14,118.4 | 7.2 | 21.3 |

**(b) Llama3.3**

| Method | Req./question | Tok./req. | Lat./req. (s) | Cost (USD) |
|---|---|---|---|---|
| RAG | 1.0 | 871.7 | 1.7 | – |
| MetaRAG | 18.9 | 26,004.6 | 121.4 | – |
| SIM-RAG | 14.1 | 31,107.5 | 103.8 | – |

**Table 6: Computational cost on HotpotQA (500 questions) under the retrieval-only setup. (a) GPT-3.5 and (b) Llama3.3 report the average number of model requests per question (Req./question), average tokens per request (Tok./req.), average latency per request in seconds (Lat./req.), and total monetary cost (USD) for processing the full dataset.**

from the original paper and code, differences in the hybrid retrieval setup may partly explain the lower absolute performance observed in our results.

Third, neither the prompts nor the baseline implementations are released. As a consequence, we re-implement both from scratch, which hinders replicability and makes it difficult to obtain the same scores for each baseline. Although our implementations closely follow the descriptions provided in the original paper, even small changes to prompts can significantly affect LLM outputs and evaluation metric results.

Fourth, the original paper reports results based on a subset of 500 QA pairs from the development split of each collection, but this subset is not shared in the released code. To replicate the experiments, we randomly sample 500 QA pairs from the respective development sets of HotpotQA and 2WikiMultiHopQA collections. This difference in sample selection likely contributes to the score variations between our results and those reported in the original MetaRAG paper.

Fifth, the original paper lacks an explanation of how the index for fetching passages from Wikipedia is constructed. The authors state that they use ElasticSearch for indexing, whereas our approach relies on Pyserini, which provides a more transparent and easily replicable interface to Lucene. Although both systems are built on the same underlying Lucene engine, differences in configuration and implementation details can influence which documents are retrieved and, consequently, affect the generated answers [15].

Sixth, the released code contains evidence of reranker usage, specifically the model `intfloat/simlm-msmarco-reranker`[24]. However, this reranker is neither mentioned nor discussed in the original paper, and its role within the MetaRAG pipeline remains unclear. Since the paper refers only broadly to hybrid retrieval, it is uncertain how this reranker is applied and how it impacts the reported results. This ambiguity motivates our further exploration of different retrieval and reranking setups to better understand their influence on metacognitive RAG frameworks such as MetaRAG and SIM-RAG.

Finally, we attempted to contact the authors on three occasions via email to clarify these discrepancies and better understand the released code. Unfortunately, we did not receive a response, which further limited our ability to resolve the identified replicability issues.

[24] https://github.com/ignorejjj/MetaRAG/blob/bd85d13cb3500afc119e178500ea9dbaee4d99e5/config.py#L18

## 7 Conclusions and Future Work

In this work, we conduct a replicability study of MetaRAG for multi-hop QA. Our results confirm its main qualitative claim: MetaRAG consistently outperforms standard prompting, Standard RAG, and strong reasoning baselines such as CoT, ReAct, Self-Ask, IRCoT, FLARE, and Reflexion on both HotpotQA and 2WikiMultiHopQA. However, we observe lower absolute scores than those reported in the original paper, particularly on 2WikiMultiHopQA, highlighting the sensitivity of RAG systems to evolving closed-source LLMs and underspecified implementation details. Our diagnostic analysis shows that retrieval is required for most questions, yet the LLM often judges the top-k evidence insufficient even when an NLI critic later validates the answer. This suggests that performance limitations are more related to evidence ordering and multi-hop reasoning than to missing information in the retrieved documents. We also find that MetaRAG is sensitive to its judgment threshold, where lower values reduce metacognitive transitions and extra retrieval while improving accuracy in our replicated setting.

We further extend MetaRAG with reranker models and observe consistent gains in both PointWise and ListWise settings, especially when paired with stronger LLM backends. In contrast, SIM-RAG, while competitive in retrieval-only configurations, does not consistently benefit from reranking and in s settings degrades subseveraltantially when the retrieved evidence is reordered or altered. This suggests that SIM-RAG's fine-tuned critic is more sensitive to retrieval modifications, whereas MetaRAG's prompt-based metacognitive control is more adaptable across configurations. Overall, our findings indicate that MetaRAG's effectiveness depends strongly on key design choices, including the judgment threshold, retrieval strategy, reranking method, and underlying LLM.

As future work, we aim to extend metacognitive RAG to additional knowledge-intensive tasks, such as biomedical question answering and mathematical reasoning, where balancing retrieval and reasoning is particularly critical. We also plan to compare metacognitive RAG frameworks with alternative approaches that learn retrieval and search behaviors via reinforcement learning, such as Search-R1 [13], as well as with other agent-based RAG systems. Moreover, we will broaden our evaluation beyond final answer accuracy by assessing the quality of retrieved evidence and the correctness and coherence of intermediate reasoning steps [5].

## References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. (2024).

[2] Hossein Bahak, Farzaneh Taheri, Zahra Zojaji, and Arefeh Kazemi. 2023. Evaluating chatgpt as a question answering system: A comprehensive analysis and comparison with existing models. *arXiv preprint arXiv:2312.07592* (2023).

[3] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216 [cs.CL]

[4] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 758–759.

[5] Shaima Ahmad Freja, Ferhat Ozgur Catak, Betul Yurdem, and Chunming Rong. 2026. EvalQReason: A Framework for Step-Level Reasoning Evaluation in Large Language Models. *arXiv preprint arXiv:2602.02295* (2026).

[6] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, Rerank, Generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2701–2715.

[7] Yasushi Gotoh. 2016. Development of Critical Thinking with Metacognitive Regulation. *International association for development of the information society* (2016).

[8] Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient Nearest Neighbor Language Models. In *Conference on Empirical Methods in Natural Language Processing*.

[9] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*. 6609–6625.

[10] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *EACL 2021-16th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 874–880.

[11] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403* (2024).

[12] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 7969–7992.

[13] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516* (2025).

[14] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551* (2017).

[15] Chris Kamphuis, Arjen P De Vries, Leonid Boytsov, and Jimmy Lin. 2020. Which BM25 do you mean? A large-scale reproducibility study of scoring variants. In *European Conference on Information Retrieval*. Springer, 28–34.

[16] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering.. In *EMNLP (1)*. 6769–6781.

[17] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. [n. d.]. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations*.

[18] Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. [n. d.]. Decomposed Prompting: A Modular Approach for Solving Complex Tasks. In *The Eleventh International Conference on Learning Representations*.

[19] Emily R Lai. 2011. Metacognition: A literature review. (2011).

[20] Md Tahmid Rahman Laskar, Sawsan Alqahtani, M Saiful Bari, Mizanur Rahman, Mohammad Abdullah Matin Khan, Haidar Khan, Israt Jahan, Amran Bhuiyan, Chee Wei Tan, Md Rizwan Parvez, et al. 2024. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. *arXiv preprint arXiv:2407.04069* (2024).

[21] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

[22] Jiatao Li, Xinyu Hu, and Xiaojun Wan. 2024. SMART-RAG: Selection using Determinantal Matrices for Augmented Retrieval. *arXiv preprint arXiv:2409.13992* (2024).

[23] Yanhong Li, Chenghao Yang, and Allyson Ettinger. 2024. When hindsight is not 20/20: Testing limits on reflective thinking in large language models. *arXiv preprint arXiv:2404.09129* (2024).

[24] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281* (2023).

[25] Vaibhav Mavi, Anubhav Jangra, Adam Jatowt, et al. 2024. Multi-hop question answering. *Foundations and Trends® in Information Retrieval* 17, 5 (2024), 457–586.

[26] Gabriel de Souza P Moreira, Ronay Ak, Benedikt Schifferer, Mengyao Xu, Radek Osmulski, and Even Oldridge. 2024. Enhancing Q&A Text Retrieval with Ranking Models: Benchmarking, fine-tuning and deploying Rerankers for RAG. *arXiv preprint arXiv:2409.07691* (2024).

[27] TO Nelson and L Narens. 1990. Metamemory: A theoretical framework and some new findings. The Psychology of Learning and Motivation. Vol. 26.

[28] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset. (2016).

[29] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! *arXiv preprint arXiv:2312.02724* (2023).

[30] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350* (2022).

[31] Ella Rabinovich, Samuel Ackerman, Orna Raz, Eitan Farchi, and Ateret Anaby Tavor. 2023. Predicting question-answering performance of large language models through semantic consistency. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*. 138–154.

[32] Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. In *European Conference on Information Retrieval*. Springer, 303–313.

[33] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[34] Gregory Schraw and David Moshman. 1995. Metacognitive theories. *Educational psychology review* 7, 4 (1995), 351–371.

[35] Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. 2024. The prompt report: a systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608* (2024).

[36] Sahel Sharifymoghaddam, Ronak Pradeep, Andre Slavescu, Ryan Nguyen, Andrew Xu, Zijian Chen, Yilin Zhang, Yidi Chen, Jasper Xian, and Jimmy Lin. 2025. Rankllm: A python package for reranking with llms. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3681–3690.

[37] Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning, 2023. *URL https://arxiv. org/abs/2303.11366* 1 (2023).

[38] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542* (2023).

[39] Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391* (2024).

[40] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509* (2022).

[41] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. 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)*. 10014–10037.

[42] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[43] Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. 2022. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.

[44] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[45] Diji Yang, Linda Zeng, Jinmeng Rao, and Yi Zhang. 2025. Knowing You Don't Know: Learning When to Continue Search in Multi-round RAG through Self-Practicing. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1305–1315.

[46] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* (2018).

[47] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

[48] Tianyu Yu et al. 2025. Unleashing the Power of Context Repetition for Robust Reasoning in LLMs. *arXiv preprint arXiv:2503.06789* (2025).

[49] Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems* 37 (2024), 121156–121184.

[50] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. 2024. mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 1393–1412.

[51] Zexuan Zhong, Tao Lei, and Danqi Chen. 2022. Training Language Models with Memory Augmentation. In *2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*. Association for Computational Linguistics (ACL), 5657–5673.

[52] Yujia Zhou, Zheng Liu, Jiajie Jin, Jian-Yun Nie, and Zhicheng Dou. 2024. Metacognitive retrieval-augmented large language models. In *Proceedings of the ACM Web Conference 2024*. 1453–1463.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009