SCMRAG: Self-Corrective Multihop Retrieval Augmented Generation System for LLM Agents

Rishabh Agrawal
University of Western Ontario
London, Canada
ragrawa9@uwo.ca

Murtaza Asrani
University of Western Ontario
London, Canada
masrani2@uwo.ca

Hadi Youssef
University of Western Ontario
London, Canada
hyousse8@uwo.ca

Apurva Narayan
University of Western Ontario
London, Canada
apurva.narayan@uwo.ca

ABSTRACT

Existing Retrieval-Augmented Generation (RAG) systems primarily depend on static knowledge vectorstores which combine semantic similarity algorithms with reranking. This often leads to outdated information and retrieval errors. In this paper, we propose SCMRAG, a Self-Corrective Multihop Retrieval Augmented Generation system for LLM agents. We introduce an LLM-assisted dynamic knowledge graph creation step to enhance information retrieval and mitigate hallucinations. Unlike traditional RAG systems, SCMRAG includes a self-corrective agent driven mechanism that autonomously identifies and retrieves missing information from external web sources. Furthermore, SCMRAG's internal reasoning agent determines whether the knowledge graph provides sufficient information or if a corrective step is needed. It further improves retrieval accuracy and efficiency. We benchmark the effectiveness of SCMRAG on five datasets - MultiHop-RAG, ARC AI2, PopQA, PubHealth, and WikiBio; showing significant improvements in retrieval precision and hallucination reduction across diverse tasks. Our results highlight SCMRAG's potential to redefine how LLM agents interact with knowledge bases, offering a more adaptable and reliable solution for a wide range of applications.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**; **Information extraction**; **Heuristic function construction**; **Planning with abstraction and generalization**.

KEYWORDS

RAG, LLM, Agents, Knowledge Graph, Vectorstore

ACM Reference Format:
Rishabh Agrawal, Murtaza Asrani, Hadi Youssef, and Apurva Narayan. 2025. SCMRAG: Self-Corrective Multihop Retrieval Augmented Generation System for LLM Agents. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

The large-scale adoption of Large Language Models (LLMs) has revolutionized almost every industry. These can range from healthcare [14] and education [10] to even cyber-security [24] and software development [17]. While LLMs demonstrate impressive capabilities in natural language understanding and generation, their dependability and trustworthiness still face significant challenges [7]. Open weights LLMs constantly struggle in delivering accurate and up-to-date information. This issue is magnified when LLMs rely solely on their internal knowledge, leading to a phenomenon known as 'hallucinations' [22].

'Hallucinations' occur when the LLM generates incorrect or fabricated information with unwarranted confidence. The output is often presented in a way that appears credible, with no indication of uncertainty or error. As a result, there is an urgent need for reliable, contextually aware and dynamically updated retrieval systems to ensure LLMs can perform effectively in real-world applications [13].

Retrieval-Augmented Generation (RAG) systems have emerged as a powerful solution to address the limitations of LLMs [25]. By integrating external knowledge bases, RAG models can retrieve relevant information during inference. This enables LLMs to handle tasks beyond their static training data. RAG systems are especially useful in reducing hallucinations and improving accuracy, particularly in fields like healthcare and education that require accurate and current information [25]. However, traditional RAG systems often rely on ranking algorithms to improve semantic similarity based search results [21]. This involves retrieving a large number of documents from a knowledge base which are then scored on a relevance metric for selecting top-n relevant documents [25].

Though traditional RAG systems reduce 'hallucinations', they have significant limitations. The main disadvantage is that they are static, unable to adapt to evolving knowledge or contexts, and frequently returns outdated or incomplete information [22]. Moreover, such systems cannot autonomously correct errors or fill gaps in the retrieved content, especially when complex multihop queries are involved [19]. The retrieval process itself can introduce errors by fetching irrelevant or tangential information. Furthermore, document semantic similarity based the top-n retrieval algorithms are susceptible to retrieving texts that misrepresent the query context when the knowledge base becomes too large or complex [2].

In this paper, we propose SCMRAG: a Self-Corrective Multihop Retrieval Augmented Generation system for LLM agents. SCMRAG introduces a novel paradigm that moves beyond the static retrieval methods of traditional RAG systems by integrating a dynamic, LLM-assisted knowledge graph for information retrieval. This knowledge graph evolves with the system, updating and refining itself based on the SCMRAG's agent driven interactions and query-answer pair generations. Crucially, SCMRAG also includes a self-corrective mechanism, enabling it to identify when information is missing or inadequate and autonomously retrieves it from external sources (e.g. web, enterprise information sources, or any other available information resources) by generating a new retrieval query without relying on predefined algorithms. This self-corrective step ensures that up-to-date and accurate information is always accessible.

Another key feature of SCMRAG is its LLM agent driven internal reasoning agent. It gives the system the decision-making capability to determine whether the knowledge graph contains sufficient information to answer a query or whether a corrective step is necessary to enhance the retrieval process. It enables SCMRAG to adapt to a wide range of tasks and domains while minimizing hallucinations.

SCMRAG ensures that only the most relevant content is retrieved from available data sources, even when the knowledge base is incomplete or outdated. The key contributions of our proposed method are as follows:

1. We introduce a novel RAG paradigm that employs a dynamic, self-updating knowledge graph to guide multihop retrieval, allowing for more context-aware and accurate information retrieval.
2. We propose a self-corrective, agent-driven mechanism that enables SCMRAG to autonomously update missing or outdated information by fetching data from external sources.
3. We achieve state-of-the-art performance on four datasets, even when using a quantized LLM with significantly fewer parameters. Notably, these results are obtained without any LLM fine-tuning.
4. We demonstrate that SCMRAG's advanced reasoning capabilities significantly reduce hallucinations by ensuring that only the most relevant and accurate information is provided to the LLM for generation.

The paper is organized as follows: **Related Work** reviews the existing literature in this domain. **SCMRAG Framework** outlines the problem and introduces our proposed solution. **Methodology** explains each module of the framework in detail. **Experimental Setup** describes the datasets and evaluation methodology. **Evaluation** presents and analyzes the experimental results, including an ablation study to assess the impact of different components. **Conclusion and Future Work** summarizes our findings and outlines future research directions.

2 RELATED WORK

The concept of augmenting generative models with external retrieval systems gained prominence with early work on open-domain question answering, where LLMs were found to suffer from hallucinations—generating factually incorrect or fabricated information. [13] This was true even for foundational models with large-scale pretraining with vast amounts of knowledge.

2.1 Initial Work

Works such as the RAG model proposed by Lewis et al. [13] were instrumental in showing that augmenting a generation model with a retrieval step could greatly improve the factual correctness of AI-generated text. Lewis et al. introduced the two-stage RAG process, where a retriever is responsible for fetching relevant documents based on a query, and a generator produces text conditioned on these retrieved documents. This approach was proven to outperform purely generative or purely extractive models in tasks such as knowledge-based QA and passage generation. This dual system ensures that the model's output is grounded in real-world data. It highlighted the importance of coupling retrieval systems with LLMs to enhance performance in open-domain tasks.

2.2 Advances in RAG Architecture

Several models have introduced further innovations to improve the efficiency and accuracy of retrieval mechanisms. Karpukhin et al. developed Dense Passage Retrieval (DPR) [9], a technique that leverages dense vector representations for more accurate retrieval of semantically relevant passages. DPR became foundational in improving the retriever's ability to return highly relevant documents from vast corpora.

Later advancements in RAG systems sought to optimize both the retrieval and generation phases. Fusion-in-Decoder [8] integrated multiple retrieved documents simultaneously within the decoder, allowing the model to generate answers that more holistically synthesized information from various sources. This method allowed for more contextual outputs, and was effective in handling multi-hop questions requiring reasoning across multiple documents.

A critical issue with these approaches is the reliance on static retrieval corpora, which limits the system's ability to access up-to-date information, leading to outdated or incomplete responses in rapidly evolving domains. Moreover, the retriever and generator components in transformer based RAG models are generally trained separately. This often leads to mismatches between retrieved documents and generated content.

2.3 Dynamic RAG Paradigms

Recent work improves retrieval and generation through more dynamic systems that adapt their context based on some of a feedback loop [25]. One notable direction is the integration of self-corrective mechanism, wherein models continually adjust and refine their outputs based on externally retrieved information from the web [22].

Other dynamic RAG models that adapt their retrieval strategies mid-generation are part of a newer wave of research aimed at improving both the relevance and accuracy of information retrieval during the text generation process. REALM [5] is one such method which jointly trained retriever and generator allow retrieval updates during generation. RETRO [3] improves efficiency by chunking long documents into segments and retrieving relevant text chunks dynamically. Various recurrent RAG mechanisms explore iteratively refining the retrieval process as the generation progresses [25].

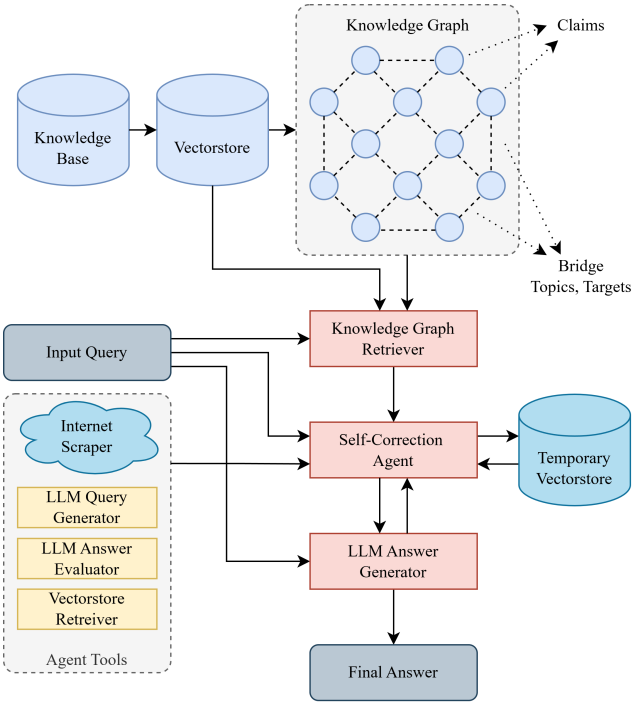

Figure 1: Overview of SCMRAG

A recent method, GRAG [6], includes the use of knowledge graphs to guide dynamic retrieval in real time. While GRAG improves multi-hop reasoning and reduces hallucinations, it is computationally complex as it relies on creating dynamic k-hop ego-graphs from a main static textual graph, which then undergo a soft pruning process to filter out irrelevant nodes and edges. It also cannot adapt to a growing knowledge base or incomplete/missing information.

3 SCMRAG FRAMEWORK

3.1 Task Formulation

The problem of solving the standard Retrieval-Augmented Generation (RAG) task can be described as follows: Given an input query X and a corpus of vectorized documents $C = \{d_1, d_2, d_3, \ldots, d_n\}$, the goal is to generate an answer Y by retrieving a subset of the corpus, denoted as $D = \{d_{r_1}, d_{r_2}, \ldots, d_{r_k}\}$, where $D \subseteq C$ and k is the number of top-ranked relevant documents. The RAG task is divided into two sub-tasks:

1. **Retrieval task**: Performed by the retriever R, where the objective is to retrieve the top k most relevant documents from C for the given query X. Mathematically, this can be represented as:

$$D = R(X, C) \tag{1}$$

where R is the retrieval function and D is the set of retrieved documents.

2. **Generation task**: Performed by the generator G, where the objective is to generate the output Y based on the retrieved documents D and the input query X. This is represented as:

$$Y = G(X, D) \tag{2}$$

where G is the generation function that produces the final answer Y using both the query X and the subset of documents D.

Thus, the standard RAG task involves optimizing both the retriever R and the generator G to ensure that the generated answer Y is both relevant and accurate based on the information present in the corpus C. As previously discussed, the standard RAG approach has limitations in retrieving up-to-date information. It also cannot comprehensively address multihop queries and hallucinations. To overcome these issues, we introduce modifications to the retriever by integrating a graph retriever $\mathcal{G}$ and a self-corrective agent $\mathcal{S}$ without the need to fine-tune the generator G.

Algorithm 1 Multihop Graph Retrieval Process $\mathcal{G}$

Require: Query X, Knowledge Graph K, Threshold τ, Number of Results n, Number of Nodes m
Ensure: Final set of relevant information chunks
Step 1: Query Representation
Prompt LLM to generate query topic T_q and query entities T_e
Step 2: Edge Evaluation
Evaluate similarity between query topic and bridge topics
Step 3: Identify sub-graph G', for top-m results
Step 4: Top-n Retrieval
Recursively identify the top-n claims for context $\hat{K}$
GRAPH RETRIEVER(Graph G', Node v)
Step 5: Completeness Check and Recursive Replacement
$\forall v_{\text{match}} | T_e \in G'$, do GRAPH RETRIEVER($G', v_{\text{match}}$)

3.2 Overview of SCMRAG

We implement a novel graph retriever $\mathcal{G}$ that generates a subgraph from a dynamic knowledge graph K, where $K = \mathcal{F}(C)$ and $\mathcal{F}$ maps the corpus C into a structured knowledge graph K. $\mathcal{G}$ retrieves the top n documents, denoted as $\hat{K}$, using a graph retrieval algorithm (detailed in section 4.2). The self-corrective agent $\mathcal{S}$ further refines this retrieval by taking the input query X and the documents $\hat{K}$, producing a new set of documents $\hat{S}$ (described in section 4.3). This two-stage retrieval ensures more accurate and relevant information. The final generation task is then modified to generate the output Y based on the query X and the set of refined documents, ensuring higher precision in the generated answers. The graph retrieval and generation tasks can be described mathematically as:

$$\hat{K} = \mathcal{G}(X, K), \quad \hat{S} = \mathcal{S}(X, \hat{K}), \quad Y = G(X, \hat{K} \cup \hat{S}) \tag{3}$$

4 METHODOLOGY

4.1 Knowledge Graph Construction

We construct a knowledge graph K from the document corpus C, which can also be dynamically updated with new documents sourced by the self-corrective agent. Each document in C is subdivided into smaller information chunks. A binary classification model, adapted from [26], is employed to evaluate whether a given chunk is factual. This step ensures that only factual information

Algorithm 2 Self-Corrective Mechanism $\mathcal{S}$

Require: Query X, Retrieved Context $\hat{K}$, Maximum Iterations c
Ensure: Final Answer Y, Updated $\hat{K}$

Step 1: Initial Answer Generation
The LLM agent generates an initial answer Y
Step 2: Evaluate Support Level L **of** Y
Step 3: Web Scraping for Additional Evidence
if Answer is not fully supported **then**
    Agent generates a retrieval query $\hat{X}$.
    The agent scrapes the web for query $\hat{X}$ as $\hat{V}$.
    Extract top-n chunks $\hat{S}$ from $\hat{V}$.
end if
Step 4: Update context $\hat{K}$ **and Answer** Y
Step 5: Final Answer and Knowledge Base Update
Once the answer Y is fully supported, append $\hat{S}$ to K

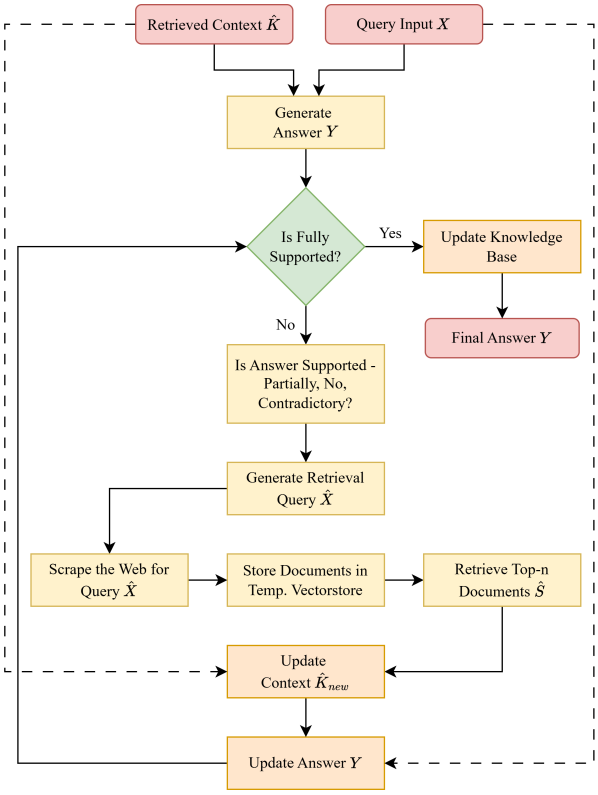

Figure 2: Execution Graph for Self Corrective Mechanism.

is retained, thereby preventing the knowledge graph from being cluttered with irrelevant or uninformative content.

For each factual chunk, we prompt a large language model (LLM) to generate a 'claim'. A 'claim' is defined as a statement or assertion that expresses a belief, opinion, or fact, devoid of ambiguous references to entities such as persons, places, or things. Claims serve as the foundational elements of the knowledge graph.

Additionally, for each claim, we generate a 'bridge target' and a 'bridge topic', which form the nodes and edges of the graph, respectively. The 'bridge target' is the specific individual, group, organization or concept that the statement or assertion within a text is directed towards or about which it is making a case. The 'bridge topic' is a phrase representing the claim's central argument or concept. These bridge topics and targets link similar claims, creating the interconnected structure of the knowledge graph. The knowledge graph construction process can be expressed mathematically as follows:

1. **Chunking the Corpus**: Let $C = \{d_1, d_2, \ldots, d_n\}$ represent the corpus of documents. Each document d_i is subdivided into a set of chunks $\{c_1, c_2, \ldots, c_m\}$.
2. **Fact Classification**: We define a binary classification model f_{class} that determines whether a chunk c_j is factual. The set of factual chunks F can be written as:

$$F = \{c_j \in d_i \mid f_{\text{class}}(c_j) = 1\} \tag{4}$$

3. **Claim Generation**: For each factual chunk $c_j \in F$, we use an LLM to generate a claim given an instruction prompt $claim_p$, resulting in a set of claims $Claims$:

$$Claims = \{\text{LLM}(c_j, claim_p) \mid c_j \in F\} \tag{5}$$

4. **Bridge Topics and Bridge Targets** connect similar claims. They are generated by instructing an LLM with a bridge topic BT_p and a bridge target BTG_p prompt respectively. The nodes and edges of the knowledge graph $K = (V, E)$ are constructed as:

$$V = \{\text{LLM}(c_j, claim_p)\} \tag{6}$$

$$E = \{(\text{LLM}(c_j, \text{BT}_p), \text{LLM}(c_j, \text{BTG}_p))\} \tag{7}$$

Thus, the knowledge graph K is built from the claims as nodes and the relationships between them (bridge topics and targets) as edges, ensuring that the graph is concise, factual, and dynamically expandable as new information becomes available.

4.2 Multihop Graph Retrieval Process

We design a novel knowledge graph search algorithm to retrieve the relevant information chunks. The retrieval process begins with an input query X, for which we generate a 'query topic' and a set of 'query entities' using a large language model (LLM). Here, a 'query entity' refers to the specific individual, group, or organization that the query or statement is directed towards or references.

The retrieval process leverages the structure of the knowledge graph by performing a multihop search. This search involves evaluating edges between nodes (claims) using a vectorization-based semantic similarity score between the 'query topic' and the 'bridge topics'. An edge is deemed relevant only if the 'bridge target' of an edge aligns with an entity from the set of 'query entities'.

Once a subgraph containing m nodes is identified, we retrieve the top n results by calculating the semantic similarity score between the 'query' and the 'claim' vectors associated with each node in the subgraph. To ensure completeness, the set of retrieved documents must cover all 'query entities'. If any query entity is missing from the retrieved set but is present within the subgraph, we recursively replace the weakest retrieved result with the closest matching node that has the missing query entity. top m and top n are empirically set to 32 and 6 respectively.

Finally, the original chunks from which each finalized claim was generated are retrieved, providing the relevant information needed

to answer the query. This multihop graph retrieval algorithm ensures that the most relevant, complete, and contextually accurate information is retrieved from the knowledge graph, improving the overall precision and relevance of the results.

4.3 Self-Corrective Mechanism

We introduce a fully automated, LLM agent-driven algorithm designed to determine whether an answer Y is supported by the information in the retrieved context $\hat{K}$, without the need for model fine-tuning or the use of special tokens for self-correction [2]. This is achieved through a zero-shot ReACT [23] reasoning chain, where the agent is equipped with tools to scrape the web, store retrieved documents in a vectorstore [22], retrieve relevant documents as evidence by generating a new query, generate an answer for the original prompt based on this evidence, and assess whether the generated answer is supported by the evidence.

The agent is initially tasked with answering the query X using the context $\hat{K}$. After generating an answer, it evaluates the support level based on the following criteria:

1. **Fully supported**: The answer is entirely consistent with the evidence, or it mirrors extractions from the evidence. This applies when the generated output closely matches the information in the retrieved context.
2. **Partially supported**: The answer is somewhat supported by the evidence, but significant portions of the output are not addressed in the evidence. For example, if a query involves two concepts and the evidence only discusses one, the answer is classified as "Partially supported."
3. **No support**: The answer either ignores the evidence or is unrelated to it. This also applies when the provided evidence is irrelevant to the query.
4. **Contradictory**: The answer directly contradicts the evidence.

If the agent determines that the answer is not fully supported, it can autonomously make use of the web scraping tool to search for additional information. New documents are then stored in a temporary knowledge base and indexed using a vectorstore. The agent can retrieve top n new documents based on a query to better support the existing answer. The original answer is updated using the new context and this process is repeated iteratively for a maximum of $c \leq 5$ number of steps or until the answer to the original query is fully supported by the updated evidence. The new information chunks in the final evidence set are then appended to the knowledge base and the knowledge graph is updated with the new information (as described in section 4.1).

5 EXPERIMENTAL SETUP

We conducted extensive experiments to demonstrate SCMRAG's superior performance and generalizability across a variety of Retrieval-Augmented Generation (RAG) tasks. Our evaluation spanned five datasets, encompassing both short-form and long-form generation tasks. In this section, we briefly describe the datasets, tasks, and baseline methods used to benchmark SCMRAG.

[1] $*$ values are taken from [22]; † values are taken from [19]
[2] + custom implementation of FactScore using LLaMA 3.1

5.1 RAG Datasets, Tasks and Metrics

We evaluate the performance of SCMRAG on the following datasets:

MultiHop-RAG (MRAG) [19]: A dataset designed to test multihop reasoning, where models are required to retrieve and combine information from multiple sources to answer complex multihop queries that cannot be answered from a single document.

ARC AI2 [4]: A challenging question-answering dataset that contains science-based multiple-choice questions, designed to test the reasoning and knowledge retrieval capabilities of AI models.

PopQA [15]: A dataset of factoid questions about pop culture, covering a wide range of topics, including movies, music, and celebrities. It is used to evaluate the model's ability to retrieve and generate factual knowledge in a trivia-style format.

PubHealth [11]: A dataset focused on fact-checking health-related claims using true-or-false questions. It is used to test the model's ability to retrieve accurate and trustworthy medical information and generate reliable responses based on scientific literature.

WikiBio [12]: A dataset for long-form generation tasks, consisting of Wikipedia biography entries. The model is expected to generate coherent, factual summaries of people based on structured biographical data.

Following previous work, FactScore [16] was adopted as the evaluation metric for WikiBio. We reimplemeted FactScore using LLaMA 3.1 (8b) [20] as the original python implementation has deprecated modules. Accuracy was adopted as the evaluation metric for MultiHop-RAG, PopQA, PubHealth, and Arc AI2. We use LLaMA 3.1 (8B) 8-bit Instruct [1] model for evaluating SCMRAG generated answers.

5.2 Language Models and Baseline Methods

To assess the effectiveness of SCMRAG, we utilized the open-weight LLaMA 3.1 (8B) 8-bit model as the LLM generator. The use of 8-bit quantization was employed to improve generation efficiency and evaluate SCMRAG's generalizability across various RAG datasets. By using an 8-bit quantized model with 8 billion parameters, we focused on the actual enhancements in retrieval performance, rather than depending on the inherent capabilities of an LLM with a large number of parameters. This is further supported by comparing the performance of LLaMA 3.1 (8B) 8-bit with GPT-3.5 Turbo [18] in Table 2, where we evaluate both models on the selected datasets without incorporating RAG.

We tested SCMRAG in two configurations: with web-search based self-correction (SCMRAG) and without web-search based self-correction (SMRAG). This allowed us to isolate the impact of SCMRAG's knowledge graph-based retriever without additional web documents. Furthermore, SCMRAG was benchmarked against state-of-the-art RAG methods in Table 1, including CRAG [22], Self-RAG [2], and Self-CRAG [22], which are known for their advancements in retrieval-augmented generation. These comparisons highlight SCMRAG's competitiveness and generalisability in both retrieval performance across diverse datasets and the quality of generated responses.

Our focus on LLaMA 3.1 (8B) 8 bit model was intentional, as we wanted to ensure that any improvements observed were attributed to our RAG approach, rather than the underlying LLM. GPT-3.5 Turbo, though a common benchmark in prior works, is deprecated

Table 1: Comparison of SCMRAG with baseline methods across five datasets. [1] [2]

Model	PopQA	WikiBio	PubHealth	ARC AI2	MRAG
RAG (GPT 3.5 Turbo)	50.8*	78.0*	69.2*	**75.3***	0.56†
CRAG [22]	59.3	74.1	75.6	54.8	-
Self-RAG [2]	54.9	81.2	72.4	67.3	-
Self-CRAG [22]	61.8	86.2	74.8	67.2	-
SMRAG **(Ours)**	60.2	83.9+	75.3	74.5	67.1
SCMRAG (Ours)	**76.6**	**87.4+**	**87.7**	73.2	**67.6**

Table 2: LLM performance on evaluated datasets.

Dataset	Number of Queries	LLaMA 3.1 (8B) 8-bit	GPT-3.5 Turbo
PopQA	14267	23.7	29.3
WikiBio	1000	63.6	71.8
PubHealth	985	65.0	70.1
ARC AI2	1172	77.2	75.3
MRAG	2556	30.2	44.0

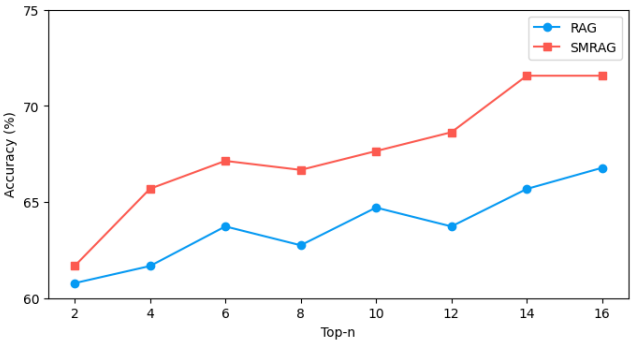

Figure 3: Comparison of Retrieval Performance and Generation Accuracy between RAG and SMRAG.

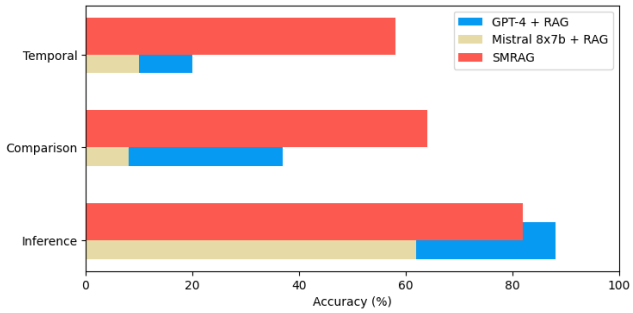

Figure 4: RAG vs SMRAG Accuracy on different query types.

and not freely accessible. Therefore, it was essential to first establish that LLaMA 3.1 is either on par with or weaker than GPT-3.5 Turbo, as described in Section 5.2, ensuring that our method's improvements were not reliant on a more powerful LLM. This approach mirrors the experiment design used for Corrective RAG evaluation [22], where GPT-3.5 Turbo was used for comparison but not integrated into their own RAG approach.

6 EVALUATION

6.1 Performance Comparison with Baseline Methods

Table 1 demonstrates that SCMRAG consistently outperforms baseline RAG methods, especially on datasets that require complex retrieval and reasoning. For PopQA, SCMRAG achieved 76.6%, surpassing the previous best method, Self-CRAG (61.8%) by a large margin of almost 15%. This result especially highlights the effectiveness of our agent based corrective step for scraping factual information about various topics from the internet. On the long-form WikiBio dataset, SCMRAG's dynamic knowledge graph integration led to a FactScore of 87.4, exceeding Self-CRAG's 86.2. SCMRAG also excelled in PubHealth, reaching 87.7%, significantly outperforming other models, thus highlighting its ability to retrieve accurate and trustworthy medical information.

For ARC AI2, while SCMRAG performed competitively with a score of 73.2%, it was slightly lower than RAG's 75.3%. However, both SMRAG and No RAG outperformed SCMRAG with 74.5% and 77.2% accuracy respectively, suggesting that scientific reasoning tasks may depend more on the LLM's internal knowledge and are prone to hallucinations when additional context is introduced. To understand this behaviour, we performed a qualitative analysis of these results in Section 6.3. Nonetheless, for MultiHop-RAG (MRAG), SCMRAG led with 67.6%, showcasing its strength in multihop reasoning. This significant improvement over RAG (56%) is a testament to SCMRAG's graph-based retrieval approach, which excels in synthesizing information from multiple sources.

Table 2 provides a baseline comparison between LLaMA 3.1 (8B) 8-bit and GPT-3.5 Turbo across the same datasets without retrieval augmentation. While GPT-3.5 Turbo generally outperformed LLaMA 3.1 on most datasets, LLaMA 3.1 (8B) 8-bit showed comparable performance on ARC AI2, suggesting it is on par with GPT-3.5 Turbo for scientific reasoning tasks. Although the underlying LLaMA 3.1 (8B) 8-bit performs significantly worse on some datasets, SCMRAG substantially enhances performance in more complex, retrieval-intensive scenarios, particularly in datasets like PubHealth and MRAG. In these cases, SCMRAG's ability to dynamically retrieve, synthesize, and correct information offers substantial improvements over baseline RAG methods.

Table 3: Qualitative evaluation of incorrect SCMRAG responses for ARC AI2.

Question	Answer	Options	GT
The morning temperature in a city is 41°F. If a sunny, mild day is forecast, which temperature is most likely for 2:00 p.m.?	Based on the given knowledge base, the most likely temperature for 2:00 p.m. on a sunny, mild day in a city with a morning temperature of 41°F is around 55-65°F (13-18°C).	A: 32°F B: 41°F C: 78°F D: 98°F	C
Which natural event occurs with the most frequency?	Based on the provided knowledge base, the natural event that occurs with the most frequency is floods	A: sunrise B: equinox C: full moon D: solar eclipse	A
Which statement is true about the solar system?	Based on the provided knowledge base, the Solar System has evolved considerably since its initial formation.	A: The gas planets are closer to the Sun. B: The solid planets are closer to the Sun. C: The planets are arranged in order of increasing size. D: The solid planets are much larger than the gas planets.	B

Table 4: Ablation study for effect of each proposed module on SCMRAG's generation performance.

Configuration	MRAG	PopQA
SCMRAG	67.6	76.6
w/o Self Correction	67.1	60.2
w/o Knowledge Graph	45.4	59.7
w/o RAG	30.2	23.7

SCMRAG's integration of a dynamic knowledge graph and self-correction mechanisms allows it to outperform not only the base LLM models but also other state-of-the-art retrieval-augmented generation systems. This demonstrates its strong potential for real-world applications that require precise and comprehensive information retrieval. SCMRAG excels in scenarios where advanced reasoning and multihop retrieval of weakly connected information are critical, highlighting its ability to effectively manage complex queries and deliver accurate, context-aware results.

6.2 Retrieval Analysis

We evaluated SCMRAG's ability to retrieve accurate documents and generate high-quality answers using only its dynamic knowledge graph and reasoning capabilities. Specifically, we compared the performance of SMRAG (SCMRAG with LLaMA 3.1 (8B) 8-bit without web-based self-correction) against a traditional RAG system across multiple top-n retrieval settings on MultiHop-RAG dataset, which demonstrated the highest improvement in generation accuracy compared to using no RAG (67.6% vs. 30.2%). We also analyzed SCMRAG's performance across different query types described in the Multihop-RAG dataset.

In Figure 3, we present the retrieval accuracy of SMRAG and standard RAG across varying numbers of top-n retrievals. SMRAG consistently outperformed RAG in document retrieval accuracy, with its performance improving more significantly over time. While the initial gains were modest at top-2 retrievals (60.7% for RAG vs. 61.6% for SMRAG), SMRAG's accuracy steadily increased over further top-n, reaching 71.5% at top-14, compared to RAG's 65.6%, 66.7%, at top-14 and top-16 respectively. This pattern supports our hypothesis that semantic similarity-based RAG scales worse with fewer top-n documents, while SMRAG's graph-based retrieval scales more effectively with increasing knowledge base size and complexity. This highlights SMRAG's strength in multihop retrieval, where it excels at retrieving information distributed across multiple documents.

We also evaluated SMRAG's top-6 retrieval performance on three different MRAG query types: Inference, Comparison, and Temporal queries. Figure 4 displays the accuracy results for each query type and compares it with the GPT-4 and Mistral 8x7b based RAG systems proposed in [19]. For **Inference Queries**, which constituted 36.20% of the dataset and requires complex reasoning over multiple retrieved documents, GPT-4 RAG [19] outperformed SMRAG, achieving an accuracy of 88% compared to SMRAG's 82%. This is likely because of the lower reasoning capability of LLaMA 3.1 (8B) 8bit compared to GPT-4 due to its smaller parameter size and quantisation.

In contrast, **Comparison Queries** and **Temporal Queries**, which made up 37.98% and 25.82% of the dataset respectively, revealed SCMRAG's significant advantages. For comparison queries, SMRAG achieved 64% accuracy, far surpassing GPT-4 RAG's 35%. Similarly, for temporal queries, SMRAG reached 58% accuracy, compared to GPT-4 RAG's 20%. These types of queries require more advanced retrieval and benefit from SCMRAG's knowledge graph, which can analyze document relationships and connect temporal sequences more effectively, enabling it to reason across multiple documents and pinpoint the most relevant information.

These results demonstrate that SCMRAG can independently retrieve and generate high-quality answers, making it highly effective for tasks that require complex reasoning. The dynamic knowledge

graph used as the core retrieval mechanism allows SCMRAG to reduce reliance on external resources for self-correction while maintaining high retrieval accuracy. Without relying on external web-based corrective searches, SCMRAG was able to synthesize complex information from the knowledge graph and generate accurate answers, particularly for intricate queries that involve evaluating relationships over multiple documents.

6.3 Qualitative Analysis

We conducted an analysis of the performance drop on the ARC AI2 dataset when using SCMRAG by validating the results for 158 queries where SCMRAG provided incorrect answers, but the base LLM produced correct answers without retrieval. This evaluation helps us determine whether the additional context from the knowledge base and web-scraped documents negatively influenced the LLM's answer generation. The LLM was specifically instructed to generate answers for the multiple-choice options based on the retrieved context, which could have introduced conflicting or misleading information. A few notable examples of such queries are presented in Table 3.

Our analysis revealed that the main issue was a combination of incorrect retrieval and the design of the question prompt itself. The questions were highly contextual to the multiple-choice options, but since the retrieval process was based solely on the question and not the options, the system often retrieved irrelevant or overly generic documents that matched the question semantically but failed to account for the specificity of the options. A potential solution would be to modify the retrieval process by including the options in the query and instructing the LLM to first generate a refined question that incorporates the context of the options before building the knowledge graph based on the revised question.

We also conducted an LLM-assisted qualitative evaluation of SCMRAG's long-form generation for WikiBio prompts, focusing on two key metrics: specificity (whether the biography was generated for the correct person) and completeness (whether all key facts from the ground truth were included in the generated text). If either of these conditions was unmet, the answer was marked as incorrect. This was performed in parallel to FactScore evaluations, as the FactScore process took a considerable amount of time to complete and showed high variance across multiple runs with our specific configuration. SMRAG had a 92% accuracy, while SCMRAG achieved 96% accuracy for both specificity and completeness. These findings further validate SCMRAG's potential for high-quality long-form generation tasks where integrate relevant information from both the knowledge graph and web documents is crucial.

6.4 Ablation Study

The ablation study in Table 4 highlights the contribution of each key module in SCMRAG by evaluating its performance on the MRAG and PopQA datasets with certain components removed. When the self-correction mechanism was disabled, there was a small but noticeable decline in performance, particularly on PopQA, where the accuracy dropped from 76.6% to 60.2%. This indicates that self-correction based web document retrieval plays a significant role in improving factoid-based retrieval tasks by enhancing the system's ability to gather and refine information through our dynamic agent. On MRAG, the drop was less severe, from 67.6% to 67.1%, suggesting that the multihop reasoning tasks in MRAG are less dependent on self-correction.

The removal of the dynamic knowledge graph had a much greater impact on MRAG, with score plummeting from 67.6% to 45.4%, showcasing the knowledge graph's critical role in enabling multihop reasoning across documents. On PopQA, performance also fell to 59.7% without the knowledge graph. This indicates though the answers for PopQA queries benefit from the knowlege graph based retrieval, they are simpler to tackle when compared to multihop queries . Finally, the removal of the entire retrieval-augmented generation (RAG) framework caused the most substantial drop in accuracy, with MRAG falling to 30.2% and PopQA to 23.7%, clearly illustrating the necessity of RAG for improving LLM generation performance. These results demonstrate that the dynamic knowledge graph and self-corrective agent framework are particularly vital in handling complex queries and ensuring accurate retrieval.

7 CONCLUSION AND FUTURE WORK

In this paper, we introduced SCMRAG, a Self-Corrective Multihop Retrieval-Augmented Generation system for LLM agents. SCMRAG presents a novel paradigm in the retrieval-augmented generation landscape by leveraging a dynamic, LLM-assisted knowledge graph combined with a self-corrective mechanism. This approach allows the system to autonomously decide when additional information is required, retrieve missing or incomplete data from external sources, and iteratively refine its responses.

Through extensive experimentation across multiple datasets and RAG tasks, SCMRAG demonstrated superior performance over traditional RAG systems, as well as state-of-the-art models like CRAG, Self-RAG, and Self-CRAG. Our comparison with both open-weight models like LLaMA 3.1 (8B) and closed-source foundational models like GPT-3.5 Turbo and GPT-4 further showcased SCMRAG's capabilities, especially in maintaining factual consistency and achieving retrieval accuracy in both short-form and long-form generation tasks. SCMRAG significantly reduces hallucinations and generates answers that are more accurate and contextually relevant.

Looking forward, SCMRAG opens up new possibilities for deploying more robust and adaptable LLM agents in real-world applications, such as customer service, healthcare, and knowledge-intensive domains. Future work could explore scaling SCMRAG to handle larger knowledge graphs, further optimizing the self-corrective mechanism, and extending its use across additional domains to enhance the generalizability and efficiency of LLM-driven systems. Ultimately, SCMRAG represents a critical step towards more reliable, adaptable, and precise LLM agents, capable of dynamically updating their knowledge and continuously improving their performance.

ACKNOWLEDGMENTS

We acknowledge NSERC Discovery Grant RGPIN 05163-2019 for financial support. This work is done in collaboration with Daniel Thanos from Sygen.AI.

REFERENCES

[1] Meta AI. 2024. Llama 3.1 - 8B. https://huggingface.co/meta-llama/Llama-3.1-8B. Accessed: 2024-10-13.

[2] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511* (2023).

[3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, 2206–2240.

[4] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457* (2018).

[5] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, 3929–3938.

[6] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. GRAG: Graph Retrieval-Augmented Generation. *arXiv preprint arXiv:2405.16506* (2024).

[7] Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric P. Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Yang Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. 2024. TrustLLM: Trustworthiness in Large Language Models. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=bWUU0LwwMp

[8] Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282* (2020).

[9] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[10] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences* 103 (2023), 102274.

[11] Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. *arXiv preprint arXiv:2010.09926* (2020).

[12] Rémi Lebret, David Grangier, and Michael Auli. 2016. Generating Text from Structured Data with Application to the Biography Domain. *CoRR* abs/1603.07771 (2016). arXiv:1603.07771 http://arxiv.org/abs/1603.07771

[13] 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.

[14] Ali Maatouk, Nicola Piovesan, Fadhel Ayed, Antonio De Domenico, and Merouane Debbah. 2024. Large language models for telecom: Forthcoming impact on the industry. *IEEE Communications Magazine* (2024).

[15] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511* 7 (2022).

[16] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251* (2023).

[17] Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an LLM to Help With Code Understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 97, 13 pages. https://doi.org/10.1145/3597503.3639187

[18] OpenAI. 2023. GPT-3.5 Turbo. https://openai.com. Accessed: 2024-10-13.

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

[20] Raja Vavekanand and Kira Sam. 2024. Llama 3.1: An In-Depth Analysis of the Next-Generation Large Language Model.

[21] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, et al. 2024. Retrieval-Augmented Generation for Natural Language Processing: A Survey. *arXiv preprint arXiv:2407.13193* (2024).

[22] Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884* (2024).

[23] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* (2022).

[24] Jie Zhang, Haoyu Bu, Hui Wen, Yongji Liu, Haiqiang Fei, Rongrong Xi, Lun Li, Yun Yang, Hongsong Zhu, and Dan Meng. 2024. When LLMs Meet Cybersecurity: A Systematic Literature Review. arXiv:2405.03644 [cs.CR] https://arxiv.org/abs/2405.03644

[25] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473* (2024).

[26] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197* (2022).