LARGE LANGUAGE MODEL DRIVEN AGENTIC FRAMEWORK FOR ADAPTIVE RETRIEVAL AND MULTI-HOP REASONING

A Major Project Report
Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

DATA SCIENCE

by

NITESH KALIA

24/DSC/07

Under the Supervision of

Mrs. Priya Singh

Assistant Professor, Department of Software Engineering
Delhi Technological University

Department of Software Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

May, 2026

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

In this regard, it gives me immense pleasure to thank Mrs. Priya Singh, an Assistant Professor in the Software Engineering Department of Delhi Technological University, for her valuable help in all possible manners during this research. Her expertise revolves around Natural Language Processing, Information Retrieval, and Reasoning Systems.

I am grateful to the faculty and staff of the Department of Software Engineering for their academic encouragement and support. Discussions with colleagues in the M.Tech. Data Science programme have sharpened my thinking on retrieval systems, language model evaluation, and the practical challenges of deploying agentic AI frameworks.

I acknowledge the open-source community whose publicly available tools and datasets made this investigation possible .In particular, the creators of the HotpotQA benchmark, the Sentence-Transformers library, the FAISS indexing library, and the Llama 3 model family, whose contributions form the technical foundation of this work.

Finally, I am deeply grateful to my family and friends for their unwavering love, encouragement, and patience throughout this demanding period. Their support has been my most important source of motivation. This thesis is dedicated to them.

Place: Delhi NITESH KALIA

Date: 24/DSC/07

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I hereby declare that the Report work entitled:

---

in partial fulfillment of the requirements for the award of the degree of Master of Technology (Data Science), submitted in the Department of Software Engineering, Delhi Technological University, is an authentic record of my own work carried out during the period from _____ to _____ under the supervision of Mrs. Priya Singh.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor(s) Signature of External Examiner

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042

CERTIFICATE

This is to certify that the Project Dissertation titled **"Large Language Model Driven Agentic Framework for Adaptive Retrieval and Multi-Hop Reasoning"** submitted by NITESH KALIA, Roll No. 24/DSC/07, Department of Software Engineering, Delhi Technological University, Delhi, in partial fulfillment of the requirement for the award of the degree of Master of Technology (Data Science), is a record of the project work carried out by the student under my supervision.

To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Mrs. Priya Singh

Date:

Assistant Professor

Department of Software Engineering

ABSTRACT

Retrieval Augmented Generation (RAG) has established itself as a foundational paradigm for grounding the outputs of Large Language Models in externally retrieved factual evidence, substantially reducing the hallucination that arises when parametric models are queried on knowledge beyond their training distribution. However, the dominant deployment pattern for RAG remains a fixed single pass architecture in which a query is submitted once to a retrieval index, a static document set is returned, and an answer is generated from that set without any mechanism for self assessment, iterative refinement, or adaptive re-retrieval. While this architecture is adequate for simple factoid questions whose complete answer can be recovered from a single retrieved document, it is fundamentally incapable of handling multi-hop questions, which are queries that require sequential reasoning across two or more documents where the answer to one intermediate reasoning step serves as the necessary context for formulating the next retrieval query. Failure mode is well characterised and systematic: If the single retrieved set is not complete (missing bridging facts), the generator either fabricates a plausible sounding but incorrect answer, or fails to complete an answer at all, without mechanisms to seek the missing evidence.

This thesis directly tackles the above fault by designing, implementing, and evaluating thoroughly a Large Language Model (LLM) Driven Agentic Framework for Adaptive Retrieval and Multi-Hop Reasoning (MR). Three cooperative agentic mechanisms are introduced to the retrieval loop, adding to the passive fixed pipeline RAG architecture and making it an active, self-aware, evidence seeking process. The first is a question classifier that allows to classify an input query as simple or compositional and processes it through the right retrieval path, without causing the degradation of the accuracy that would result if complex decomposition strategies were applied to simple questions. The second is an iterative query rewriter that, at each retrieval step, explicitly identifies the specific information missing from the accumulated evidence set and constructs a targeted sub-query to retrieve it, appending only new documents to prevent redundant context growth and terminating when no new evidence is returned. The third is a dual-signal sufficiency checker that combines a dense retrieval cosine similarity threshold with an LLM based answerability judgement, requiring both signals to be satisfied simultaneously before answer generation is initiated, thereby reducing the probability of generating from insufficient evidence.

The framework is evaluated on one hundred questions from the HotpotQA distractor benchmark, a dataset specifically designed to test multi-hop retrieval and reasoning under realistic retrieval noise conditions, achieving 46.0% Exact Match and 58.66% F1. These results represent absolute improvements of twenty four and sixteen percentage points respectively over a Vanilla RAG baseline, corresponding to a relative

improvement of 109% in Exact Match, and outperform all three fixed strategy baselines compared. A structured ablation study establishes that all three components are necessary and complementary, with query rewriting identified as the single most critical component whose removal causes a ten percentage point Exact Match drop and a twenty five percentage point Retrieval Recall collapse. A comprehensive failure analysis demonstrates that the framework successfully shifts the dominant error mode from retrieval failure to reasoning failure, directly redirecting future research effort toward generation quality rather than retrieval design. Together, these findings establish the proposed framework as a principled and empirically validated solution to the multi-hop retrieval and reasoning challenge that is practically deployable without any annotated reasoning supervision or model fine-tuning.

Keywords: Adaptive Planning, Agentic AI, Dual-Signal Sufficiency Checking, Information Gap Analysis, Iterative Query Rewriting, Large Language Models, Multi-Hop Question Answering, Query Reformulation, Retrieval-Augmented Generation.

CONTENTS

LIST OF TABLE(S)

LIST OF FIGURE(S)

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AGI	Artificial General Intelligence
API	Application Programming Interface
CoT	Chain-of-Thought
DPR	Dense Passage Retrieval
EM	Exact Match
FAISS	Facebook AI Similarity Search
FiD	Fusion-in-Decoder
FLARE	Forward-Looking Active REtrieval
F1	F1 Score
HMAS	Hierarchical Multi-Agent System
LLM	Large Language Model
MRR	Mean Reciprocal Rank
NLP	Natural Language Processing
OOS	Out-of-Scope
QA	Question Answering
RAG	Retrieval-Augmented Generation
RRF	Reciprocal Rank Fusion
SD	Standard Deviation
SEM	Standard Error of the Mean

CHAPTER 1

INTRODUCTION

Being able to respond to questions using a vast body of knowledge definitely ranks as one of the most crucial capabilities of an intelligent computer. It is also one of the primary concerns explored by the AI and NLP research community over the past decades. Early approaches to question answering relied on hand crafted rule systems and structured databases, which could answer questions within tightly constrained domains but could not generalize to the breadth of questions that arise in real world knowledge intensive settings. The advent of statistical machine learning, and subsequently deep learning, substantially expanded the range of questions these systems could handle, but the fundamental tension between the breadth of human knowledge and the limited capacity of any fixed model to represent it remained unresolved. That tension has been brought into sharp focus by the emergence of Large Language Models (LLMs), which represent perhaps the most ambitious attempt yet to encode broad world knowledge within the parameters of a neural model trained on web-scale text corpora [1].

Large Language Models trained through next token prediction on hundreds of billions of tokens of human generated text have demonstrated remarkable breadth of knowledge and strong performance across diverse language understanding and generation tasks. These models encode, within their billions of parameters, vast quantities of factual knowledge, linguistic pattern, commonsense reasoning, and domain expertise accumulated from the human generated text on which they were trained. The practical consequence is that a well trained LLM can respond to an extraordinary variety of questions from its parametric memory without accessing any external resource, a capability that has led to widespread enthusiasm about their potential for deployment in knowledge intensive applications ranging from customer service automation to scientific research assistance.

Yet the same parametric architecture that confers this impressive breadth also introduces a fundamental and well characterised limitation. The knowledge encoded in the parameters of a language model is frozen at the conclusion of training. It cannot be updated to reflect new information without retraining the model on augmented data, a process that is both computationally expensive and logistically complex. Any

information that post dates the training cutoff, that was not adequately represented in the training corpus, or that requires access to specialised proprietary or domain specific sources is inaccessible to the model through parametric recall. When a model is queried on such information, the consequences range from a graceful admission of ignorance ,acceptable but unhelpful ,to the generation of plausible sounding but factually incorrect statements, a phenomenon that has come to be known as hallucination and that represents one of the most practically consequential failure modes of parametric language model deployment in knowledge intensive contexts.

1.1 Retrieval Augmented Generation and Its Limitations

Retrieval Augmented Generation, introduced by Lewis et al. in 2020 [1], provides a principled architectural response to the parametric knowledge limitation. By coupling a parametric language model with a non-parametric retrieval module that accesses an external document corpus at inference time, RAG enables the generation of answers that are explicitly grounded in retrieved evidence.The retrieval module encodes the query into a dense vector and extracts the most similar passages from a corpus to be concatenated with the query and fed into the generator, which then generates the answer based on the evidence provided and not just on the parametric memory. This architecture, not only decreases hallucination, but also allows the knowledge base to be updated without changing the model parameters and facilitates. Post-hoc verification by making it possible to trace the answer back to the source documents.

The RAG paradigm has proven extraordinarily influential and has been adopted as the dominant architectural pattern for production deployments of LLM based question answering systems. Yet a critical limitation in the standard RAG architecture has remained largely unaddressed in the most widely deployed systems: the retrieval step is performed exactly once, using exactly the original query as provided by the user, and no mechanism exists for assessing whether the retrieved evidence is sufficient or for seeking additional evidence if it is not. This fixed single pass retrieval architecture is, in essence, an open loop system: it executes the retrieval step, passes whatever documents are returned to the generator, and produces an answer regardless of whether those documents contain the complete evidence needed to answer the question reliably.

For simple factoid questions , a questions whose complete answer is contained in a single document that is retrievable from the original query ,this single pass architecture works adequately. But for multi-hop questions ,a questions that require the sequential integration of evidence from two or more documents through a chain of intermediate reasoning steps ,the single pass architecture fails in a systematic and predictable

way. The bridging facts that connect one reasoning hop to the next are often not co-located in any single document and cannot be retrieved predictably from the original question alone, because the original question does not contain the intermediate entity names and relationship assertions that would locate the bridging document. The architecture is structurally incapable of handling this class of questions, and no amount of improvement to the retrieval model or the generator alone can overcome this structural limitation [2].

1.2 Multi-Hop Reasoning: The Core Challenge

Multi-hop questions are not an artificial or contrived category invented for benchmarking purposes; they are a natural and prevalent class of real world knowledge intensive queries. Questions of the form "Who directed the film that starred the actor who won the Academy Award for Best Actor in a given year?" require the system to first identify the award winner, then identify a film they starred in, then identify that film's director ,a three hop chain of reasoning in which no single document contains all necessary information and in which the query for each hop can only be formulated once the answer to the preceding hop is known. Such questions arise naturally in legal research, scientific literature synthesis, competitive intelligence, historical analysis, and medical information retrieval, among many other knowledge intensive domains. Any system that aspires to serve as a general purpose knowledge intensive question answering assistant must be capable of handling them.

The fundamental challenge of multi-hop questions for retrieval systems is precisely the one identified above: the circularity between knowing what to retrieve and having retrieved enough to know what to look for next. A static single pass retrieval system is trapped by this circularity. An adaptive agentic system, by contrast, can break the circularity through iterative reasoning: retrieve whatever is available from the original query, identify what intermediate information has been established and what remains missing, formulate a targeted new query based on that analysis, retrieve again, and iterate until either sufficient evidence has been accumulated or no new evidence can be found. This iterative, self-aware, evidence seeking process is the defining property of the framework proposed in this thesis.

1.3 The Proposed Approach

This thesis proposes a Large Language Model Driven Agentic Framework for Adaptive Retrieval and Multi-Hop Reasoning that transforms the passive fixed pipeline RAG

architecture into an active, iterative, evidence-seeking process. The framework is organised around three cooperative mechanisms that together implement an adaptive retrieval loop. The question classifier distinguishes simple queries from compositional queries at the entry point of the loop, routing each question through the processing path appropriate to its complexity. The most important part of the framework is the iterative query rewriter which is located in the retrieval loop, and whose job is to find out what information is not in the evidence base that is currently being accumulated and to create a sub-query that looks for just that information. The dual-signal sufficiency checker checks the sufficiency of the retrieved content before each attempt to generate an answer as a dense retrieval cosine similarity signal and an LLM based answerability judgement.

These three mechanisms are all tightly coupled, not add-on components of the standard RAG pipeline, but rather integrated aspects of a unified adaptive loop. The framework does not involve annotated reasoning chains or model fine-tuning or require any domain specific engineering, just prompting the interaction with a publicly available model. It is a language model combined with a standard dense retrieval infrastructure, which allows for easy deployment in a variety of knowledge rich question answering applications.

1.4 Research Objectives

The research aims of this thesis can be stated as follows. The key goal is to design and develop an agentic LLM based framework that tackles the reported shortcomings of single pass RAG on multi-hop questions by adaptively classifying queries, iteratively re-forming according to information gaps and reliably checking sufficiency based on two signals to avoid noisy duplicate or irrelevant answers without the need for fine-tuning or annotated supervision. The second objective is to rigorously evaluate this framework on a challenging multi-hop benchmark, comparing its performance against three fixed strategy baselines across a multi-metric evaluation protocol that captures both answer quality and retrieval behaviour. The third objective is to conduct a structured ablation study that quantifies the individual contribution of each architectural component to the system's overall performance, establishing the necessity and complementarity of all three mechanisms. The fourth objective is to perform a systematic failure analysis that characterises the distribution of error types across all compared systems, identifying the current bottleneck in the pipeline and providing concrete guidance for future research directions. The fifth objective is to provide, through a comprehensive review of related work, a thorough contextualisation of the proposed framework within the broader landscape of retrieval, reasoning, and agentic AI research.

1.5 Contributions of This Thesis

The principal contributions of this thesis are as follows. First, a unified agentic RAG framework is proposed that, to the best of the author's knowledge, is the first to integrate adaptive question classification, information-gap-grounded iterative query rewriting, and dual signal evidence sufficiency checking within a single retrieval loop operating without fine-tuning and without annotated supervision. Each of these three capabilities has precedents in the prior literature, but no prior work combines all three in a single integrated system. Second, the framework is empirically validated on the HotpotQA distractor benchmark, achieving a twenty four percentage point absolute improvement in Exact Match over Vanilla RAG and outperforming all three fixed strategy baselines by substantial and statistically meaningful margins. Third, a component level ablation study establishes the necessity and non-redundancy of all three architectural components, with the relative importance hierarchy query rewriting (-10% EM) $>$ iteration (-8% EM) $>$ decomposition (-7% EM). Fourth, a failure analysis demonstrates that the framework shifts the dominant error mode from retrieval failure to reasoning failure, providing a concrete and actionable diagnostic finding that directly guides future research priorities.

1.6 Organization of the Thesis

The rest of this thesis follows as follows. In Chapter 2, the technical background of large language models, retrieval augmented generation, dense retrieval, FAISS indexing, multi-hop question answering as a benchmark task and evaluation metrics are covered. The related work review and literature survey in Chapter 3 covers the fundamental RAG methods, multi-step retrieval methods, LLM based agent frameworks and the wider landscape of agentic AI. The proposed methodology is presented in great detail in Chapter 4 and includes details on the selection of the data sets, formalization of the problem, architecture of the system and the characteristics of its implementation. The experimental results and detailed discussion are included in Chapter 5, which includes the key performance comparison, ablation study, failure analysis and efficiency characterisation. The thesis is concluded in chapter 6 and directions for future research are pointed out.

CHAPTER 2

TECHNICAL BACKGROUND

This chapter is a stepping stone and lays the theoretical and technical background for understanding the design decisions and setup for the proposed framework. This thesis introduces concepts such as large language models and parametric knowledge limitation, retrievalaugmented generation framework and detail of dense retrieval, approximate nearest neighbour indexing using FAISS, multi-hop question answering task and HotpotQA benchmark, and the evaluation metrics employed in this thesis.

2.1 Large Language Models: Architecture and Knowledge Encoding

The transformer architecture, which was first proposed in 2017 by Vaswani et al.[3]serves as the backbone of all major large language models (LLMs) to date. The transformer applies an input sequence to a stack of multi-head self-attention layers, where each layer calculates a weighted average of all the other token representations in the input sequence, for each token. This global attention mechanism allows the model to attend to any other token, even when it is positioned far away, and provides it with the ability to represent long range syntactic and semantic dependencies that were more difficult or impossible for earlier recurrent architectures. Feed forward sublayers Each transformer block is finished with and residual connections that offer the non linear transformations and gradient-flow stability required for deep network training.

Transformer language models trained using next token prediction on web scale corpora with hundreds of billions of tokens of human generated text such as news articles, books, webpages, code, scientific papers, and conversational transcripts show an emergent property, quite different from their predecessors, that captures a broad and integrated representation of human knowledge and reasoning patterns, which results in strong performance on an immense range of downstream tasks with few or no examples. The Llama 3 model family [4], used for generation in this thesis, represents a recent and publicly available instantiation of this paradigm, providing strong reasoning capabilities in a range of parameter scales from 8 billion to 70 billion parameters.

The parametric knowledge limitation discussed in Chapter 1 is a direct consequence

of this pre-training paradigm. The model's knowledge is determined by its training data and is fixed within its parameters at the conclusion of training. Asking the model questions outside of its training distribution or training cutoff date either refuses to participate (disgust and divine providence) or refuses to answer correctly (honest but unhelpful) or hallucinates (generates a confident sounding but factually incorrect answer that can't be cross referenced with any source because it's generated from no source beyond the model's imperfect parametric memory).

2.2 Retrieval Augmented Generation

Retrieval Augmented Generation [1] addresses the parametric knowledge limitation through a principled architectural extension: augmenting the parametric generator with a non-parametric retrieval component that provides access to an external knowledge corpus at inference time. The core mechanism of RAG is straightforward. Given a natural language query q, a retriever module encodes q into a dense vector representation and retrieves the top-k most similar passages from a pre-indexed corpus, where similarity is measured by cosine similarity between the query embedding and the pre-computed passage embeddings. The retrieved passages are concatenated with the original query and presented to a generative language model as an augmented input context, and the model produces an answer conditioned on both the original query and the retrieved evidence.

The knowledge base used by the retriever is stored as a collection of pre-computed dense embeddings and can be updated independently of the model parameters by re-encoding new or updated documents and adding them to the index. This separation of knowledge from reasoning is the fundamental architectural advantage of RAG over purely parametric approaches: it allows the model to access current, domain specific, and verifiable information without the expense of model retraining, and it allows the source of any generated answer to be traced to specific retrieved documents through inspection of the retrieval results.

The standard RAG pipeline, however, performs the retrieval step exactly once, using exactly the original query. This single-pass, fixed query design is the source of the multi-hop failure mode that motivates this thesis. Section 4.2 formalises this failure mode and shows how the agentic framework resolves it.

2.3 Dense Retrieval and the Sentence Transformer Model

Dense retrieval systems represent both queries and documents as continuous-valued vectors in a shared embedding space, using a bi-encoder architecture in which a single encoder network (or two separate encoders with shared or related weights) transforms text into fixed length dense vectors [5]. The similarity between a query vector and a document vector is computed as their cosine similarity, and the top-k most similar documents are returned as the retrieval result. Because document embeddings can be pre-computed offline and stored in an index, the retrieval operation at inference time consists only of encoding the query and performing a nearest neighbour search in the pre-built index, an operation that scales efficiently with corpus size.

The Sentence Transformers library [6], built on the BERT architecture, provides a family of bi-encoder models fine-tuned through contrastive learning on semantically labelled sentence pairs. In this way, the comparative learning process involves training the encoder with the aim of making sure that sentences having semantic similarities will end up in the same place irrespective of their differences in the lexicon; therefore, the comparison here will be based on semantic similarity, not syntactic similarity. The model being employed here, the `all-MiniLM-L6-v2`, generates a 384-dimensional vector, which is the best balance between accuracy and computation time. It should be noted that the documents and the query have already been normalized to unitary ℓ_2 norm.

2.4 FAISS for Nearest Neighbour Search

The Facebook AI Similarity Search (FAISS) library [7] provides highly optimised implementations of nearest neighbour search algorithms for high dimensional vector spaces, supporting both exact and approximate search methods. The `IndexFlatIP` index used in this thesis computes the exact inner product between the query vector and every indexed document vector and returns the top-k matches in order of decreasing inner product. For ℓ_2-normalised vectors, inner product is equivalent to cosine similarity, so this index performs exact cosine similarity search. The computational complexity of the search is $O(Nd)$ where N is the number of indexed vectors and d is the embedding dimension. For the corpus of 50,491 passages used in this thesis with $d = 384$, this cost is computationally manageable on standard hardware and the exactness guarantee ensures that no relevant documents are missed due to approximation error.

The index is built once from the whole passage corpus, and is reloaded for each experimental run, so that the same retrieval infrastructure is used for all the systems

and ablation variants compared. This design decision ensures that any performance differences that are observed are due to planning and query formulation strategies, and not to any indexing artefacts or differences in the retrieval infrastructure.

2.5 Multi-Hop Question Answering

Multi-hop question answering is formally defined as the task of answering natural language questions that require the integration of evidence from two or more distinct documents through an intermediate chain of reasoning steps. It is distinct from single hop question answering, in which the answer can be recovered from a single retrieved document, by the structural property that no individual document in the corpus is sufficient to answer the question: the answer can only be derived by combining facts from multiple documents through a sequence of reasoning operations.

The HotpotQA dataset [2], used as the primary benchmark in this thesis, was constructed specifically to evaluate multi-hop retrieval and reasoning capability. Question-answer pairs were collected through a crowdsourcing process that required annotators to formulate questions whose answers depend on combining information from exactly two Wikipedia articles, and annotators verified that neither article alone is sufficient to answer the question. The distractor version of HotpotQA, used in this thesis, supplements the two gold supporting articles with eight topically related but non-answering passages for each question, creating a realistic retrieval challenge in which the retrieval system must discriminate between relevant evidence and plausible distractors. The availability of gold answer strings for all questions enables fully automatic evaluation of Exact Match and F1. The experimental corpus used in this thesis comprises 50,491 passages extracted from the supporting and distractor documents of 5,000 training questions, with evaluations performed on 100 held out validation questions.

2.6 Evaluation Metrics

This thesis employs six metrics that together characterise system performance across both answer quality and retrieval behaviour dimensions.

Exact Match (EM) is the primary answer quality metric. It assigns a score of 1 if the predicted answer string, after normalisation, exactly matches the gold answer string, and 0 otherwise. Normalisation consists of converting to lowercase, stripping leading and trailing whitespace, removing articles (a, an, the), and eliminating punctuation. EM is an exact evaluation measure, providing a very conservative estimate for the actual correctness of the answer.

F1 score evaluates the token-wise overlap between the system-produced answers and gold answers. Given predicted answer token set $\hat{a}$ and gold answer token set A:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad \text{Precision} = \frac{|\hat{a} \cap A|}{|\hat{a}|}, \quad \text{Recall} = \frac{|\hat{a} \cap A|}{|A|} \tag{2.1}$$

F1 awards partial credit for answers that partially overlap with the gold string and therefore provides a more lenient and arguably more representative measure of answer quality than EM.

Retrieval Recall measures the fraction of evaluation questions for which the gold answer string appears in at least one retrieved document across all retrieval iterations, directly capturing the adequacy of the retrieval component in locating relevant evidence.

Mean Reciprocal Rank (MRR) evaluates rank quality by computing the mean reciprocal of the rank position of the first relevant document across all queries:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \tag{2.2}$$

where rank_i is the position of the first relevant document for query i.

Average Retrieval Steps measures the mean number of retrieval iterations per question, capturing the efficiency of the adaptive planning strategy. Latency measures mean wall-clock time in seconds per question. For binary metrics (EM and Retrieval Recall), statistical uncertainty is expressed as the binomial standard deviation $\sigma = \sqrt{p(1-p)/n}$ where p is the observed proportion and $n = 100$.

CHAPTER 3

RELATED WORK AND LITERATURE REVIEW

The chapter introduces the proposed framework and conducts a thorough and critical review of the literature. The literature review is organized around four thematic streams: foundational RAG methods and its variants, multi-step and iterative retrieval methods, LLM-based agentic planning frameworks, and the context of agentic AI systems and applications. The chapter is based on a guide to agentic AI[8] that accompanies the chapter. The wider background in which the technical aspects of this thesis fit.

3.1 Foundational Retrieval Augmented Generation

The RAG paradigm was formalised by Lewis et al. [1], who integrated DPR-based dense passage retrieval [5] with a sequence-to-sequence generative model, demonstrating that retrieval grounding substantially reduces hallucination on open domain question answering benchmarks including Natural Questions [9] and TriviaQA [10]. The foundational insight separating knowledge storage from language model parameters and providing evidence through retrieval at inference time proved highly influential and spawned a large family of architectural variants.

Izacard and Grave extended the RAG architecture through their Fusion-in-Decoder (FiD) model [11], which processes each retrieved passage independently through a shared encoder and fuses the resulting representations at the decoder through cross-attention. FiD demonstrated that increasing the number of retrieved passages beyond the handful used in the original RAG architecture could substantially improve answer quality on multi-document questions, establishing the importance of comprehensive retrieval coverage. However, both RAG and FiD perform retrieval in a single pass on the original query, with no adaptive re-retrieval mechanism.

The RAG Fusion approach [12] addressed the single-query limitation through static multi-query expansion: the original question is paraphrased into several query variants using LLM prompting, retrieval is performed independently for each variant, and results are combined through reciprocal rank fusion. While RAG Fusion improves recall

through query diversification, the expansions are generated by paraphrasing the original question rather than by analysing what has already been retrieved and what is still missing. The query expansion is therefore not adaptive in the sense used in this thesis; it does not respond to the informational content of intermediate retrieval results.

3.2 Multi-Step and Iterative Retrieval

The limitations of single pass retrieval for compositional questions have motivated a family of multi-step retrieval methods. Self-Ask [13] introduced the decomposition before retrieval paradigm: the input question is decomposed into a sequence of sub questions using LLM prompting, each sub-question is retrieved and answered independently, and intermediate answers are chained forward as context for subsequent sub-questions. Self-Ask captures the essential structural insight that compositional questions require compositional retrieval strategies in which each step is formulated based on what the preceding steps have established. However, Self-Ask applies this decomposition unconditionally to all questions regardless of their actual complexity. When the decomposition step is used for question decomposition without multi-hop reasoning, it adds sub question noise to the retrieval process, thereby polluting the retrieval process with irrelevant queries. The experimental results of this thesis validate this limitation with the results showing that Self-Ask RAG performs worse than Vanilla RAG on HotpotQA, as reported in the original paper that unconditional decomposition is shown to harm performance in simpler queries.

FLARE (Forward-Looking Active REtrieval) [14]proposed a novel paradigm of adaptively retrieving: FLARE observes the per token confidence of the generator during generation and requests further retrieval when the confidence is below a threshold. The idea is that a confidence loss indicates a lack of information and that more information would enhance generation quality. FLARE showed that using confidence driven re-retrieval can outperform single-pass retrieval on multi-hop questions. However, the per token confidence signal is a noisy proxy for information need, no structured analysis of what information is missing is provided by FLARE, and there is no explicit sufficiency check so generation can continue from inadequate evidence even when a confidence signal doesn't indicate a gap.

IRCoT (Interleaved Retrieval with Chain-of-Thought) [15] represents perhaps the most directly related prior work, interleaving retrieval steps with chain-of-thought reasoning steps in a tight coupling that effectively implements reasoning-guided retrieval. At each iteration, the model generates a reasoning step based on current evidence, and the next retrieval query is derived from the reasoning output, targeting the next subgoal

in the reasoning chain. IRCoT achieved strong results on HotpotQA and demonstrated the value of tight coupling between reasoning and retrieval for multi-hop questions. Its critical limitation is that it requires annotated chain-of-thought reasoning chains for training, making it inapplicable in settings where such annotations are unavailable. The framework proposed in this thesis achieves competitive or superior performance on HotpotQA without any annotated supervision.

3.3 LLM-Based Agent Frameworks

The development of LLM based agent frameworks has provided the conceptual foundation for the agentic retrieval approach taken in this thesis. The ReAct framework [16] established the think-then-act paradigm as the foundational approach to agentic behaviour with language models: at each step, the agent produces a natural language reasoning trace describing its current understanding and the action it will take, executes that action through tool invocation, and incorporates the result into subsequent reasoning. The interleaving of reasoning and action in ReAct enables coherent and interpretable decision processes and demonstrated strong performance on knowledge-intensive multi-hop reasoning tasks. The framework proposed in this thesis shares the think-then-act structure of ReAct and extends it with explicit question classification and principled sufficiency verification, the absence of which represents the main limitation of the original ReAct approach.

Chain-of-Thought (CoT) prompting [17] established that eliciting explicit step-by-step intermediate reasoning from language models substantially improves performance on complex reasoning tasks including multi-hop factual reasoning. By asking the model to "think step by step" before producing a final answer, CoT decomposes complex problems into tractable sub-steps and uses the outputs of earlier steps as context for later ones. While highly effective for purely parametric reasoning within the model's training distribution, CoT provides no mechanism for retrieving externally stored evidence, leaving the model vulnerable to hallucination on knowledge intensive queries where the required information is not reliably encoded in parameters.

Toolformer [18] demonstrated that language models can learn to invoke external tools, including search engines, through self supervised training by learning to predict when tool invocations would improve the continuation of generated text. This self supervised tool use learning requires no manual annotation of tool invocations and enables autonomous tool use across a range of API types. However, Toolformer does not include a structured planning loop for multi-hop tasks and provides no mechanism for evidence sufficiency assessment.

3.4 Agentic AI: Broader Context and Survey Perspective

The proposed framework is situated within a broader and rapidly evolving field of agentic AI research that has been surveyed comprehensively in the companion work to this thesis [8]. That survey defines agentic AI through a three dimensional capability taxonomy encompassing reasoning and reflection, action and tool usage, and interaction and coordination, and documents the deployment of agentic systems across healthcare, financial services, manufacturing, education, and smart governance. Several aspects of this broader landscape are directly relevant to the technical contributions of the present work.

All the technical challenges highlighted in the survey report are scalability, reliability and security in real world deployments of agents in all the domains. Reliability is a key point that needs attention, especially where LLM based agents make decisions without checking for evidence sufficiency: If an LLM based agent decides on an answer without explicitly checking for evidence sufficiency, it can produce inconsistent or factually incorrect answers, and this unreliability can cascade and compound in multi step reasoning chains. This thesis tackles this reliability challenge directly with a two signal sufficiency checking mechanism that is proposed to be used before the generation to greatly decrease the probability of answers generated from insufficient evidence. The low number of out-of-scope failures in the near elimination of out of scope failures in the experimental results (9 failures in Vanilla RAG, 1 failure in Agentic RAG) is a tangible empirical evidence. of this benefit.

The survey additionally reveals that metrics for agentic AI systems go beyond what is suitable for non agentic systems, such as accuracy and perplexity, and include autonomy, adaptability, robustness, and the nature of human AI interactions. The multi metric evaluation protocol used in this thesis is representative of This is a multi dimensional assessment requirement, offering a thorough characterization of system behaviour beyond what is possible to describe with a single metric. Table 3.1 summarises the prior survey literature on agentic AI, identifying the coverage of key dimensions and noting the contribution of the companion survey.

3.5 Gap Analysis and Positioning of This Work

The systematic comparison in Table 3.2 positions the proposed framework against the most directly related methods across the six dimensions most relevant to the multi-hop question answering evaluation. The analysis reveals a consistent pattern in the prior literature: each existing approach addresses at most two of the three capabilities

Table 3.1. Comparative Summary of Prior Surveys on Agentic AI and Multi-Agent Systems.

Author	**Primary Focus**	**HMAS Axes Covered**	**Key Contribution**	**Limitation**
Zhang et al. [19]	Multi-agent security	Control hierarchy; Information flow	Security analysis across multi-agent systems	No temporal hierarchy or communication structures
Chen et al. [20]	Evolutionary computation + MAS	Control hierarchy; Role/delegation	Compares architectures for reasoning and planning	Does not extend to dynamic communication
Wang et al. [21]	LLMs for robotics	Role/delegation	Agentic AI for embodied systems	Focused on VR/robotics; no broad HMAS analysis
Tran et al. [22]	Communication protocols	Communication structure (static)	LLM-driven multi-agent protocols	Ignores temporal layering; static topology
Kitano [23]	Generative to agentic AI	Control hierarchy	Conceptual definition and challenges	Conceptual only; lacks technical taxonomy
Ferrag et al. [24]	LLM reasoning to agents	Information flow	Comprehensive LLM-to-agent reasoning review	No integrated cross-domain evaluation
Kalia & Singh [8]	**Comprehensive agentic AI**	**All five HMAS axes**	**Integrated taxonomy; cross-domain analysis; evaluation frameworks**	**Rapid field evolution requires periodic updates**

that the proposed framework integrates. Self-Ask captures question decomposition but lacks sufficiency checking; FLARE provides dynamic re-retrieval but lacks structured decomposition and sufficiency verification; IRCoT achieves reasoning-guided retrieval but requires annotated chain-of-thought supervision; ReAct interleaves reasoning and action but lacks explicit sufficiency verification. No prior unsupervised work combines adaptive classification, information-gap-grounded query rewriting, and dual-signal sufficiency checking in a single unified loop.

The gap identified in Table 3.2 defines precisely the contribution of this thesis. The combination of all three mechanisms within a single unsupervised adaptive retrieval loop is novel, and the empirical validation on HotpotQA demonstrates that this combination yields substantially larger performance gains than any single mechanism or pair of mechanisms achieves in isolation.

Table 3.2. Systematic Comparison of Related Methods on Six Key Dimensions. ✓= supported, × = not supported, Partial = limited or heuristic support.

Authors	Method	Query Rewrite	Suff. Check	Multi-Hop	Advantage	Limitation
Lewis et al. [1]	RAG: single-pass dense retrieval	×	×	×	Reduces hallucination	No iterative retrieval
Karpukhin et al. [5]	DPR: dual-encoder retrieval	×	×	×	Strong dense baseline	Overlap dependent
Izacard & Grave [11]	FiD: fusion-in-decoder	×	×	Partial	Fuses multiple passages	Single-step retrieval
Press et al. [13]	Self-Ask: question decomposition	Partial	×	✓	Structured decomposition	Degrades simple queries
Wei et al. [17]	CoT: chain-of-thought	×	×	Partial	Improves reasoning	No retrieval grounding
Yao et al. [16]	ReAct: reasoning + tool-use	Partial	×	✓	Interleaves reasoning with tools	No sufficiency check
Jiang et al. [14]	FLARE: confidence-triggered retrieval	Partial	×	Partial	Avoids unnecessary retrievals	No structured decomposition
Trivedi et al. [15]	IRCoT: interleaved retrieval + CoT	×	×	✓	Strong multi-hop results	Requires annotated CoT
Schick et al. [18]	Toolformer: self-supervised tools	×	×	×	Self-supervised tool use	No planning loop
Rackauckas [12]	RAG-Fusion: multi-query RRF	Partial	×	Partial	Improves recall	Static expansion
Proposed	**Agentic RAG: adaptive dual-signal**	✓	✓	✓	**All three; no fine-tuning**	Higher latency

CHAPTER 4

METHODOLOGY

This chapter presents the complete methodology of the proposed agentic framework in the following order: dataset selection and characteristics, formal problem formulation, system architecture and the three cooperative mechanisms, implementation details, and baseline system descriptions.

4.1 Dataset Selection and Characteristics

The selection of HotpotQA [2] as the primary evaluation benchmark reflects three substantive considerations that are detailed here. Established multi-hop benchmarks differ significantly in their construction methodology, retrieval challenge, and evaluation protocol, and a careful choice is essential for the validity of the experimental conclusions.

TriviaQA [10] and Natural Questions [9] are predominantly single-hop benchmarks: the majority of their questions can be answered from a single retrieved document, providing insufficient discrimination between fixed and adaptive retrieval strategies. Evaluating on these datasets would likely show little difference between Vanilla RAG and the proposed framework, not because the framework is ineffective but because the test does not exercise the capability the framework is designed to provide. 2WikiMultiHopQA [25] constructs multi-hop questions synthetically from Wikipedia knowledge graphs. While the synthetic questions do require multi-hop reasoning, the construction process which entails programmatically chaining simple questions through known graph paths produces questions that are more structured and predictable than naturally occurring compositional queries, likely underestimating the retrieval difficulty faced in real world settings. MuSiQue [26] provides high quality multi-hop questions constructed by composing single-hop questions but does not include the distractor setting that is essential for evaluating retrieval precision under realistic noise.

HotpotQA in its distractor configuration provides the most demanding and realistic evaluation setting among available benchmarks. Every question requires genuinely combining evidence from two Wikipedia articles, neither of which alone is sufficient to answer the question. The eight distractor passages per question create a realistic

noise environment in which the retrieval system must identify the two relevant documents from ten candidates that are all topically related to the question. The automatic evaluation through gold answer strings eliminates the need for human annotation. The experimental corpus of 50,491 passages provides a realistically scaled retrieval challenge, and the 100 question evaluation sample, consistent with prior iterative retrieval experiments [15], is sufficient to detect the large effect sizes reported in the results while remaining computationally manageable for systematic ablation experiments.

4.2 Formal Problem Formulation

Let q denote a natural language question and A its gold answer string. A corpus $\mathscr{C} = \{d_1, d_2, \ldots, d_N\}$ contains N text passages. A retrieval function $R(q,k) \to \mathscr{D}$ returns the top-k passages ranked by cosine similarity to the encoded query q, and a generator $G(\mathscr{D}, q) \to \hat{a}$ produces a predicted answer conditioned on the retrieved passage set $\mathscr{D}$ and the question q. The standard single-pass RAG formulation is:

$$\hat{a} = G(R(q,k), q) \quad (4.1)$$

This formulation assumes that $R(q,k)$ contains sufficient evidence to answer q, an assumption that fails for multi-hop questions as established in Chapter 1. The problem is reformulated as an iterative planning process. Let $\mathscr{D}^{(t)}$ denote the accumulated document set at iteration t, initialised as $\mathscr{D}^{(0)} = \emptyset$. At each iteration, the agentic planning policy $\pi^{(t)}$ is:

$$\pi^{(t)} = \begin{cases} \text{Stop and Generate} & \text{if Suff}\left(\mathscr{D}^{(t)}, q\right) = \text{True} \\ \text{Retrieve}\left(q^{(t)}_{\text{refined}}\right) & \text{otherwise} \end{cases} \quad (4.2)$$

where $q^{(t)}_{\text{refined}}$ is a refined sub-query constructed from the information gaps identified in $\mathscr{D}^{(t)}$. The dual signal sufficiency predicate Suff is defined as:

$$\text{Suff}\left(\mathscr{D}^{(t)}, q\right) = \left[\max_{d \in \mathscr{D}^{(t)}} \text{sim}(d,q) \geq \tau\right] \wedge \left[\text{LLMAnswerable}\left(\mathscr{D}^{(t)}, q\right) = \text{True}\right] \quad (4.3)$$

where $\text{sim}(d,q)$ is the cosine similarity between passage d and query q in the dense embedding space, and $\tau = 0.35$ is a calibrated threshold. The conjunction requires both a topical proximity signal from the dense index and a semantic adequacy signal from the LLM, addressing the distinct failure modes that each signal alone cannot prevent. This loop runs for a maximum of four iterations denoted by $T_{\max}$ to limit computation

costs; upon reaching this limit, the generation process continues based on the evidence collected.

4.3 System Architecture: The Adaptive Retrieval Loop

The proposed framework integrates three cooperative mechanisms within the adaptive retrieval loop defined by Equations (4.2) and (4.3). Figure 4.1 illustrates the complete adaptive planning loop with both the simple and complex question paths.

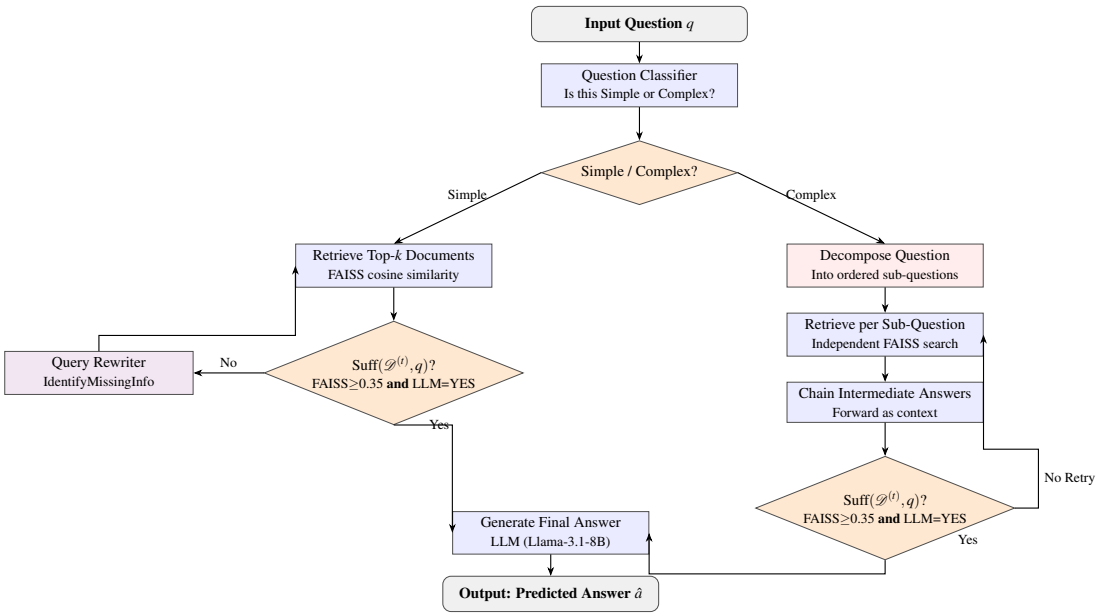

Figure 4.1. Adaptive planning loop of the proposed Agentic RAG framework. Simple questions follow the left branch (single retrieval + sufficiency check + query rewrite loop); complex questions follow the right branch (decomposition, per-sub-question retrieval, answer chaining). Both branches gate generation through the dual sufficiency check (FAISS ≥ 0.35 and LLM answerability = YES).

4.3.1 Question Classifier

The question classifier ClassifyQuestion(q) is invoked at the entry point of the adaptive loop and determines which of two retrieval paths the question will follow. The classifier prompts the LLM with the question q and asks whether the question can be answered by a single retrieval operation that is, whether its complete answer is likely to be contained in a single document or whether it requires synthesising information from two or more distinct documents. Questions classified as simple follow the left branch of the planning loop, performing a single retrieval pass with the original query and applying the sufficiency check before generating an answer. Questions classified as complex follow the right branch, engaging the full query decomposition and iterative

rewriting machinery.

The rationale for this classification step is made concrete by the comparison with Self-Ask RAG in the experimental results. Question Decomposition is performed on all queries for Self-Ask, but the method underperforms in comparison to Vanilla RAG on the HotpotQA test dataset. The reason for poor performance is that when a simple query is broken into smaller queries, the result would be retrieval queries that are irrelevant to the main query itself. The classifier prevents this degradation by ensuring that the full agentic planning loop with its decomposition, iterative rewriting, and sufficiency checking overhead is reserved for questions that genuinely require it.

4.3.2 Iterative Query Rewriter

The iterative query rewriter is the most impactful component of the framework, as established by the ablation study in Section 5.2. It operates within the retrieval loop on both the simple and complex question paths and is responsible for directing each retrieval iteration toward the specific evidence that is most needed given what has already been accumulated.

For each retrieval iteration t at which the sufficiency predicate has not yet been satisfied, the rewriter first calls IdentifyMissingInfo($\mathscr{D}^{(t)}, q$), which prompts the LLM to analyse the accumulated document set $\mathscr{D}^{(t)}$ in light of the question q and to articulate precisely what factual information is present in the accumulated set and what is still missing. The distinction between the proposed framework and previous fixed query iterative retrieval based methods is the missing information identification, which turns the implicit information gap into an explicit NLP description that allows constructing a query.

Then a sub-query $q^{(t)}_{\text{refined}}$ is built, which is a query that finely targets the identified gap and uses the intermediate answers found in previous iterations as context to maximize the precision of the new query. The difference from static query expansion techniques is that the reformulation is based on the retrieved and missing content of the query, rather than on surface paraphrases of the original query. This enables the rewriter to escape the topical neighbourhood of the original query and find bridging documents that are not directly connected to the original query.

To prevent redundant context growth, only new documents $\Delta\mathscr{D}^{(t)} = R(q^{(t)}_{\text{refined}}, k) \setminus \mathscr{D}^{(t-1)}$ are appended to the accumulated set. The loop terminates when $|\Delta\mathscr{D}^{(t)}| = 0$, preventing wasted iterations when the evidence base has been exhausted.

The additional mechanism of question decomposition is used for complex questions following the RIGHT branch. The agent breaks down q into a sequence of sub-

questions, retrieves and generates the answers to each of these sub-questions, and iteratively adds intermediate answers on to a list to be used in the later sub-questions and in the final answer generation step. The decomposition method facilitates the application of a logical sequence for solving compositional problems in a very systematic way.

4.3.3 Dual-Signal Sufficiency Checker

Dual Signal Sufficiency Checking computes the predicate Suff($\mathscr{D}^{(t)}, q$) at every iteration when retrieving documents. The first signal is calculated based on the FAISS index by retrieving the cosine similarities of the documents in $\mathscr{D}^{(t)}$ against the initial query q, and comparing the highest cosine similarity with the cutoff $\tau = 0.35$. This signal provides a rapid, index-based assessment of topical coverage. The second signal is an LLM-based answerability judgement: the accumulated documents and the question are presented to the LLM with a prompt asking it to determine whether the provided evidence is sufficient to answer the question reliably, producing a binary True/False response. The logical conjunction of these two signals gates each transition from retrieval to generation, preventing answer generation from topically adjacent but informationally incomplete evidence sets.

The threshold $\tau = 0.35$ was determined empirically on a held-out development subset. Values significantly above this threshold would cause the system to over retrieve on questions where a single highly relevant document is quickly found; values significantly below would allow generation from weakly relevant evidence. The cosine similarity score is computed using pre-normalised inner products from the FAISS index, making the computation essentially free given that the document embeddings are already indexed.

4.4 Implementation Details

All systems share a common retrieval and generation infrastructure. Passage encoding uses `sentence-transformers/all-MiniLM-L6-v2` [6] producing 384-dimensional ℓ_2-normalised vectors. The FAISS `IndexFlatIP` index over all 50,491 passage vectors is built once and reloaded for all runs, retrieving top-$k = 5$ passages per query. Answer generation uses Llama-3.1-8B-Instant [4] via the Groq API at temperature 0, with a prompt enforced one to five word answer length constraint and a post processing layer for verbose outputs. Table 4.1 summarises all hyperparameters.

Table 4.1. Complete Hyperparameter and Implementation Details.

Parameter	**Value**
Embedding model	`sentence-transformers/all-MiniLM-L6-v2`
Embedding dimension	384
Embedding normalisation	ℓ_2-normalised
Retrieval index	FAISS `IndexFlatIP` (exact cosine)
Corpus size	50,491 passages
Top-k per iteration	5
FAISS sufficiency threshold τ	0.35
Maximum iterations $T_{\max}$	4
LLM	Llama-3.1-8B-Instant (Groq API)
LLM temperature	0 (deterministic)
Answer length constraint	1–5 words (prompt-enforced)
Evaluation set size	100 validation questions

4.5 Baseline Systems

Three fixed strategy baselines are evaluated alongside the proposed framework. Vanilla RAG submits the original query once, retrieves top-5 passages, and generates an answer directly. It represents the standard single pass RAG architecture and is the direct baseline against which the value of agentic planning is measured. Self-Ask RAG unconditionally decomposes all input questions into sub-questions before retrieval and performs independent retrieval for each sub-question, following [13]. It represents the approach of applying compositional retrieval strategies to all questions and isolates the effect of unconditional decomposition. Iterative RAG performs up to $T_{\max} = 4$ retrieval iterations but always resubmits the original query without reformulation at each step, appending new documents to the accumulated set. It represents iterative retrieval without adaptive query reformulation and isolates the value of reformulation over naive re-retrieval.

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSION

This chapter presents the complete experimental results including the main performance comparison, the structured ablation study, the failure mode analysis, and the efficiency trade off characterisation, with detailed quantitative interpretation following each set of results.

5.1 Main Performance Results

Table 5.1 presents performance of all four systems on 100 HotpotQA validation questions across all six metrics. Figure 5.1 visualises the Exact Match and F1 scores side by side, covering both the four main systems and the three ablation variants discussed in Section 5.2, providing a single reference chart for the full result set.

Table 5.1. Main Performance Results on HotpotQA ($n = 100$). Bold = best per column. $\pm$ = binomial SD for EM/Recall, SEM for F1.

System	EM% $\pm$	F1% $\pm$	Rec% $\pm$	Steps	Lat. (s)
Vanilla RAG	22.0 $\pm$ 4.14	42.41 $\pm$ 4.00	61.0 $\pm$ 4.88	1.00	2.52
Self-Ask RAG	20.0 $\pm$ 4.00	33.66 $\pm$ 4.07	58.0 $\pm$ 4.94	2.75	4.71
Iterative RAG	25.0 $\pm$ 4.33	46.40 $\pm$ 4.06	69.0 $\pm$ 4.62	2.00	4.28
Agentic RAG	**46.0 $\pm$ 4.98**	**58.66 $\pm$ 4.33**	**76.0 $\pm$ 4.27**	2.19	15.60

The proposed Agentic RAG framework achieves 46.0% Exact Match, 58.66% F1, and 76.0% Retrieval Recall is the highest values across all compared systems on every metric. The absolute improvement over Vanilla RAG is twenty four percentage points in Exact Match (a relative gain of 109%), sixteen percentage points in F1, and fifteen percentage points in Retrieval Recall. These gains represent a substantial and practically significant improvement in all dimensions of system performance.

The underperformance of Self-Ask RAG relative to Vanilla RAG is among the most practically important findings in the table. Self-Ask achieves 20.0% Exact Match and

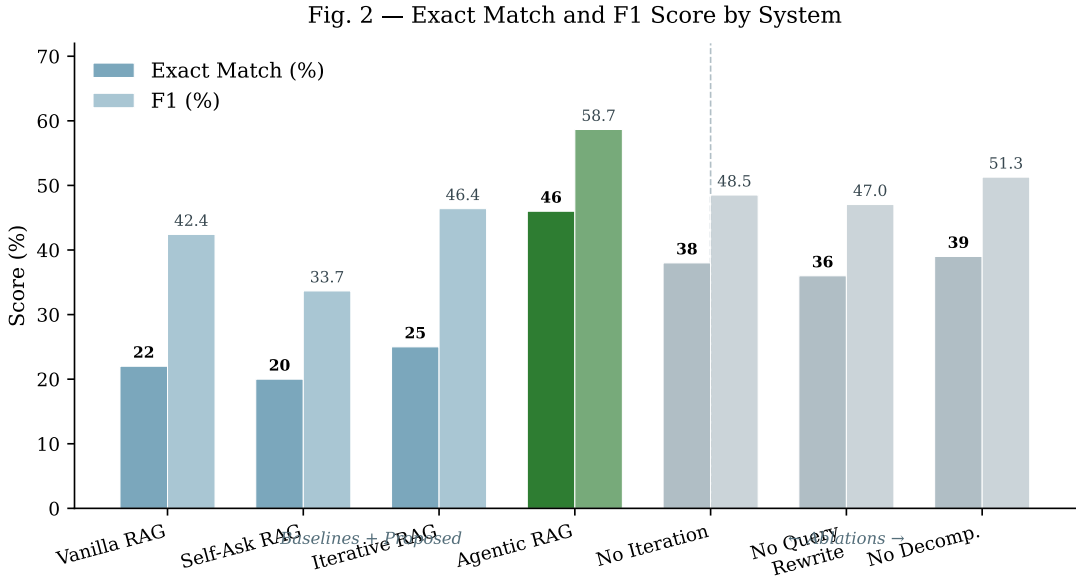

Figure 5.1. Exact Match (%, dark bars) and F1 (%, light bars) for all four systems and three ablation variants. Agentic RAG achieves the highest scores (EM = 46%, F1 = 58.7%). The dashed line separates the main systems from the ablation variants.

33.66% F1 below Vanilla RAG's 22.0% and 42.41% respectively despite performing 2.75 retrieval steps on average compared to Vanilla RAG's fixed 1.00. Applying sub-question generation unconditionally to all questions introduces sub query noise that retrieves tangentially related documents and confuses the generator without providing the specific bridging evidence needed for genuine multi-hop questions. The out-of-scope failures in Self-Ask RAG reach 18, compared to 9 for Vanilla RAG, indicating that the decomposition step leads the system to pursue sub-questions for which no answer exists in the corpus.

The comparison between Vanilla RAG and Iterative RAG isolates the benefit of simple iteration without reformulation. Iterative RAG improves Exact Match from 22.0% to 25.0% and Retrieval Recall from 61.0% to 69.0% through additional retrieval passes on the original query, but the Exact Match gain is modest because additional passes without reformulation return documents similar to those already retrieved, providing redundant rather than complementary evidence.

The step from Iterative RAG to Agentic RAG a twenty one percentage point EM gain at a marginally higher step count (2.19 vs. 2.00) — demonstrates that the value of the agentic framework lies entirely in the quality of its queries rather than in the quantity of its retrieval iterations. Each retrieval step in Agentic RAG is directed by an explicit analysis of what information is missing, making it far more likely to locate the specific bridging facts needed for the compositional reasoning chain.

5.2 Ablation Study

The ablation study removes one component at a time from the full system while retaining all others. No Iteration reduces the agent to a single retrieval pass while retaining classification and decomposition. No Query Rewrite retains the iterative loop but resubmits the original query unchanged at every step. No Decomposition retains iteration and rewriting but disables sub-question decomposition for complex queries. Table 5.2 reports quantitative results.

Table 5.2. Ablation Study Results. Each variant removes exactly one component. ΔEM is relative to the full Agentic RAG system.

Variant	EM%	F1%	Rec%	Steps	ΔEM
Agentic RAG (Full)	46.0	58.66	76.0	2.19	—
No Iteration	38.0	48.51	61.0	1.00	−8%
No Query Rewrite	36.0	47.04	51.0	1.65	−10%
No Decomposition	39.0	51.28	71.0	2.23	−7%

Figure 5.2 visualises the EM drop from each individual component removal against the full-system reference line at 46%.

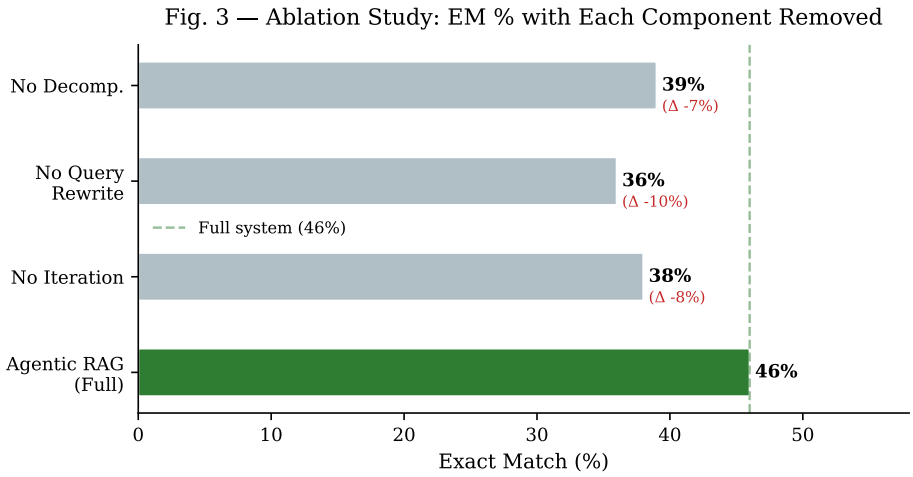

Figure 5.2. Exact Match (%) per ablation variant. The dashed line marks the full system at 46%. No Query Rewrite causes the largest drop (Δ − 10%), followed by No Iteration (Δ − 8%) and No Decomposition (Δ − 7%).

The ablation hierarchy is: No Query Rewrite (−10% EM, −25% Retrieval Recall) > No Iteration (−8% EM) > No Decomposition (−7% EM). The three components are essential, because their absence results in a considerable decrease, thus establishing their complementarity. Notably, the No Query Rewrite variant has an average of

1.65 iterations, and the Retrieval Recall performance collapsed from 76% to 51%. A number of iterations with no explicit reformulation will just go around in the same neighbourhood looking for bridging facts which are only accessible by reformulating. This finding establishes query rewriting as the mechanism that makes iteration productive rather than wasteful.

The eight percentage point EM drop when iteration is removed, with Retrieval Recall collapsing identically to Vanilla RAG's 61.0%, confirms that iteration is the mechanism responsible for recall improvement over single-pass retrieval. The No Iteration variant still substantially outperforms Vanilla RAG in EM (38.0% vs. 22.0%) because the question classifier and decomposition are retained. The seven-percentage-point EM drop from removing decomposition, with Recall only mildly affected (76.0% $\rightarrow$ 71.0%), confirms that decomposition primarily aids answer synthesis rather than document retrieval.

5.3 Failure Mode Analysis

All 100 outcomes for each system are classified into four error categories: retrieval failure (gold answer absent from all retrieved documents), reasoning failure (gold answer present but generator produces an incorrect answer), format failure (substantively correct answer in incorrect surface form), and out-of-scope failure (question unanswerable from the corpus). Table 5.3 presents the complete distribution across all systems and ablation variants.

Table 5.3. Failure Type Distribution Across All Systems and Ablation Variants ($n = 100$).

System	Correct	Ret. Fail	Reas. Fail	Fmt. Fail	OOS
Vanilla RAG	22	27	24	18	9
Self-Ask RAG	20	21	28	13	18
Iterative RAG	25	22	23	22	8
Agentic RAG	**46**	**16**	23	14	1
No Iteration	38	30	22	9	1
No Query Rewrite	36	38	19	7	0
No Decomposition	39	22	27	11	1

Figure 5.3 visualises the error distribution as stacked bars, making the dominant error category per system immediately visible.

The most significant finding is the shift of the dominant error mode from retrieval

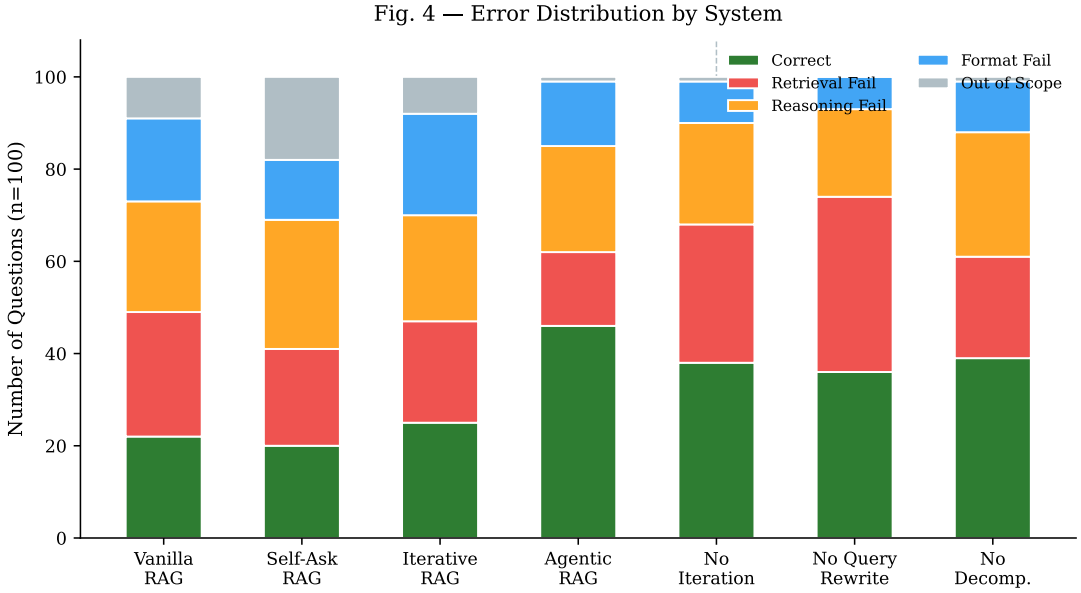

Figure 5.3. Error distribution by system ($n = 100$): Correct (green), Retrieval Fail (red), Reasoning Fail (orange), Format Fail (blue), Out-of-Scope (grey). Agentic RAG achieves the highest correct count (46) and reduces retrieval failures from 27 to 16, shifting the dominant error mode to reasoning failure.

failure to reasoning failure in Agentic RAG. In Vanilla RAG, retrieval failures (27 cases) constitute the largest single error category, reflecting the systematic inability of single pass fixed query retrieval to locate complete evidence for compositional questions. In Agentic RAG, retrieval failures are reduced by 40% to 16 cases through iterative, targeted, sufficiency-checked retrieval.The dual sufficiency check results in a drop from 9 to 1 in out-of-scope failures, showing that the check succeeds in identifying questions that can't be answered and leads to informed abstention instead of hallucination. However, reasoning failures are still at 23, matching the performance of Vanilla RAG and in fact making up the bulk of the error type, meaning the performance challenge now is not coming from the retrieval component, but from the 8B-parameter generator.

This change is not just a descriptive report, it is a diagnostic actionable result. The retrieval component is near or at ceiling and further improvements in retrieval will be diminishing returns. Future efforts should focus on the generation component by using more powerful generators, better multi-step reasoning prompts, or fine-tuning them with a specific domain. The ablation data reinforces this interpretation: retrieval failures peak at 38 in No Query Rewrite, confirming that rewriting drives the retrieval gains, while reasoning failures peak at 27 in No Decomposition, confirming that decomposition primarily aids synthesis.

5.4 Efficiency and Latency Trade-Off

Agentic RAG incurs 15.60 seconds per question compared to 2.52 seconds for Vanilla RAG ,a factor of approximately 6×. This overhead arises from the additional LLM calls required for question classification, information gap identification, query reformulation, and sufficiency assessment at each iteration. Figure 5.4 plots average latency against Exact Match for all systems, making the operating-point trade-off visible.

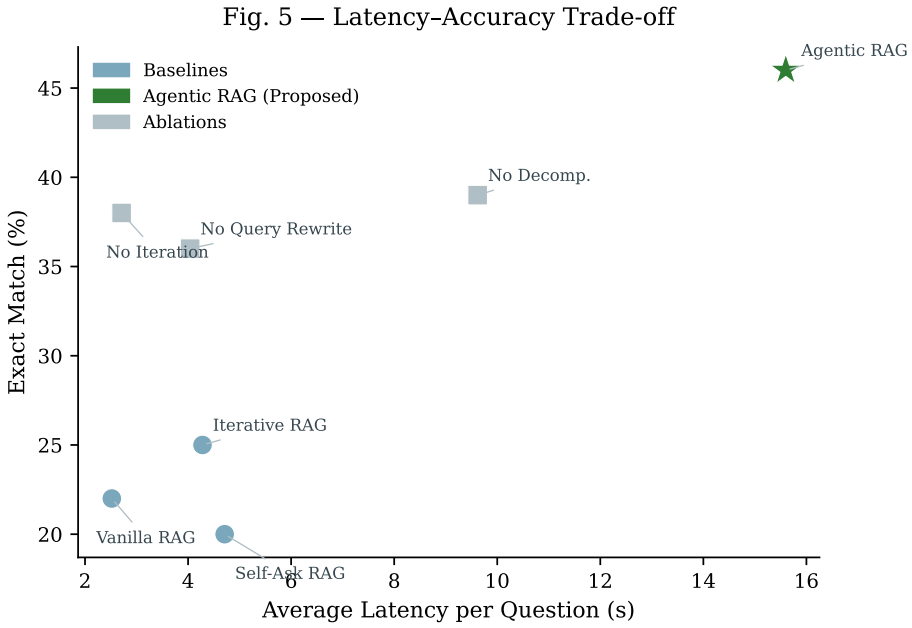

Figure 5.4. Latency–accuracy trade off. Agentic RAG (green star) achieves the highest EM (46%) at the cost of the highest latency (15.6 s). The +24 pp EM gain for a 6× latency increase is favourable in accuracy-critical settings.

The trade-off is strongly asymmetric in favour of accuracy. In legal research, medical information synthesis, and scientific question answering, the cost of an incorrect or hallucinated answer substantially exceeds the cost of a few additional seconds of processing time, making the trade-off clearly acceptable for these high value use cases.

The comparison between Agentic RAG (2.19 average steps) and Self-Ask RAG (2.75 average steps) shows that Agentic RAG achieves 2.3× higher Exact Match with fewer retrieval steps, confirming that query quality is far more important than step count. Figure 5.5 makes this explicit by plotting average retrieval steps against Exact Match across all systems.

There is no positive correlation between step count and accuracy among the fixed-strategy baselines and ablation variants. Iterative RAG performs exactly 2.00 steps per question yet achieves only 25.0% EM, while Agentic RAG's marginally higher

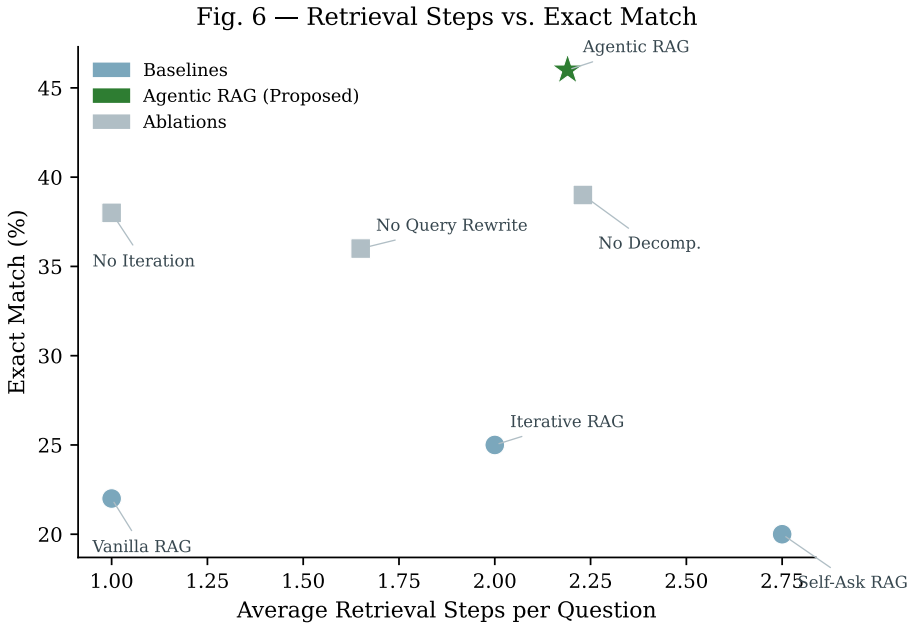

Figure 5.5. Retrieval steps vs. Exact Match (%).

2.19 steps yields 46.0% EM. The difference is entirely attributable to query quality: each Agentic RAG step is directed at the specific missing information, while each Iterative RAG step resubmits the same original query. The question classifier provides the architectural basis for a latency aware routing strategy in production deployment, allowing simple questions to bypass the full planning loop and substantially reducing mean latency across a realistic mixed complexity query distribution.

5.5 Broader Discussion

The experimental results reported in this chapter carry several implications beyond the specific quantitative findings. The fundamental implication is that the passive fixed-pipeline RAG architecture is structurally inadequate for the class of queries that arises most naturally in real world knowledge intensive settings, and that addressing this inadequacy requires architectural features that go beyond retrieval model improvement or increased document coverage. The agentic properties of adaptive query classification, information gap grounded reformulation, and principled sufficiency verification are not incremental engineering improvements but qualitative architectural innovations that change the nature of what the retrieval process can accomplish.

The zero supervision property of the proposed framework, no annotated reasoning chains, no retrieval relevance labels, no model fine-tuning is particularly important for practical deployability. It means that the framework can be applied to new domains

and document collections without incurring the cost of domain specific annotation, making it immediately applicable to settings where labeled data is scarce or expensive. This is precisely the situation in many of the most high value applications of multi-hop question answering: legal analysis, medical literature review, and scientific discovery.

Failure analysis has found that there is a blockage at the reasoning stage in the generation phase. This result strongly suggests where the research work of the future should be directed namely to an improvement in the retrieval component. The most straightforward routes to its future lie in moving to more capable generators and to the creation of more effective We introduce prompting strategies to synthesize multi step answers and explore fine-tuning the generator for a specific domain with multi-hop reasoning examples. Further details of these directions are provided in Chapter 6.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This thesis has introduced a Large Language Model Driven Agentic Framework for Adaptive Retrieval and Multi-Hop Reasoning and comprehensively tested it in the real world. The work was motivated by the well characterised failure of fixed single pass RAG architectures on compositional multi-hop questions, a failure that arises from the structural mismatch between the independence assumption embedded in single pass retrieval and the sequential, evidence dependent nature of multi-hop reasoning. The proposed framework addresses this failure by transforming the passive retrieval pipeline into an active, iterative, evidence seeking process through the integration of three cooperative mechanisms: adaptive question classification, information-gap-grounded iterative query rewriting, and dual signal evidence sufficiency checking.

The framework achieves 46.0% Exact Match and 58.66% F1 on one hundred questions from the HotpotQA distractor benchmark, representing absolute improvements of twenty four and sixteen percentage points respectively over Vanilla RAG and outperforming all three fixed strategy baselines. The structured ablation study establishes the necessity and non-redundancy of all three components, with the importance hierarchy query rewriting (-10% EM, -25% Recall) $>$ iteration (-8% EM) $>$ decomposition (-7% EM). The failure analysis shows that the framework's most diagnostic contribution is that the mode of failure has shifted from retrieval failure to reasoning failure, and that the current failure mode is identified as a generation failure, with a 40% reduction in the number of retrieval failures.

In simpler terms, there is no use of any kind of annotative reasoning or fine-tuning here. Only the prompt is used, depending solely on open-source language models and dense retrievers. This zero supervision facility significantly extends its applicability and makes it a ready to go solution to knowledge intensive question answering in diverse domains for deployment in the real world. The work lays a baseline foundation for future agentic RAG research and highlights an interpretable diagnostic outcome (the retrieval tor easoning bottleneck shift) that offers a roadmap for next generation

systems.

6.2 Future Work

Several concrete and well motivated directions for future research emerge directly from the findings of this thesis. The most immediate priority, established by the failure analysis, is the evaluation of more capable generator models. The shift of the dominant error mode to reasoning failure indicates that improvements in generation quality will yield larger gains than further refinement of the retrieval component. Evaluating the framework with Llama-3.1-70B, GPT-4o, and other large scale generators would quantify the headroom available through generation improvement and determine whether the reasoning failures that persist with the 8B parameter generator can be resolved through increased model capacity. The framework's retrieval loop, being model agnostic, requires no modification to evaluate with any generator.

The question classifier provides the architectural foundation for a latency aware query routing strategy that would substantially reduce mean latency in mixed complexity production settings. A learned classifier trained on examples of simple and complex questions with known complexity labels could provide more precise routing predictions than the current prompted LLM approach, enabling the agentic loop to be activated with higher precision for genuinely complex queries and bypassed more reliably for simple ones.

Scaling the evaluation from 100 to 500–1000 questions would provide more statistically precise estimates of performance differences, particularly for the smaller effect sizes between closely matched systems, and would reduce the binomial uncertainty that currently limits the precision of the comparative claims. Direct experimental comparison with ReAct [16] and RAG-Fusion [12] in a controlled, common infrastructure setting would provide a more complete positioning of the proposed framework within the landscape of adaptive retrieval strategies. Extension to other multi-hop benchmarks including MuSiQue [26] and 2WikiMultiHopQA [25] would test the generalizability of the framework's advantages to different compositional question structures and hop depths.

Domain transfer is a particularly high value direction. The framework's zero supervision property makes it immediately applicable to specialized domains where annotated multi-hop training data is scarce, including medical literature synthesis, legal case analysis, and scientific question answering. Evaluating the framework on domain-specific corpora with domain specific query distributions would establish whether the prompting-based information gap identification and query reformulation generalise

across domains without domain-specific adaptation, or whether domain adaptive prompting strategies are needed.

The dual signal sufficiency checker could be refined through more sophisticated prompt design for the LLM answerability signal and through dynamic calibration of the cosine similarity threshold based on the characteristics of the query and the corpus. A learned sufficiency model, trained on examples of sufficient and insufficient evidence sets for multi-hop questions, could provide more precise sufficiency estimates than the current prompted approach. The iterative query rewriter could similarly be improved through few shot examples of effective information gap identification and query reformulation, guiding the LLM toward more precise and targeted reformulations.

Finally, the integration of the proposed framework with domain specific knowledge sources ,curated medical databases, legal corpora, proprietary enterprise document repositories would translate the research contributions into deployable systems for the high stakes knowledge intensive applications that represent the most compelling use cases for adaptive multi-hop question answering. These applications not only offer the most practical motivation for this work, but are also the most challenging test of the strength and generalizability of the framework.

BIBLIOGRAPHY

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocaüschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.

[2] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2369–2380.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.

[4] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten *et al.*, "The llama 3 herd of models," in *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[5] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 6769–6781.

[6] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 3982–3992.

[7] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2021.

[8] N. Kalia and P. Singh, "A survey on agentic AI: Applications, challenges and evaluation frameworks," 2025, unpublished manuscript, Department of Software Engineering, Delhi Technological University, Delhi, India.

[9] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, and K. Toutanova, "Natural questions: A benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 452–466, 2019.

[10] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 1601–1611.

[11] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 2021, pp. 874–880.

[12] A. Rackauckas, "Rag-fusion: A new take on retrieval-augmented generation," 2024.

[13] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis, "Measuring and narrowing the compositionality gap in language models," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 5687–5711.

[14] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active retrieval augmented generation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7969–7992.

[15] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023, pp. 10 014–10 037.

[16] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *Proceedings of the Eleventh International Conference on Learning Representations*, 2023.

[17] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[18] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach them-

selves to use tools," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[19] D. Zhang, G. Feng, Y. Shi, and D. Srinivasan, "Physical safety and cyber security analysis of multi-agent systems: A survey of recent advances," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 319–333, 2021.

[20] T. Chen *et al.*, "The confluence of evolutionary computation and multi-agent systems: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, pp. 1–19, 2025.

[21] J. Wang *et al.*, "Large language models for robotics: Opportunities, challenges, and perspectives," *Journal of Automation and Intelligence*, vol. 4, no. 1, pp. 52–64, 2025.

[22] K. T. Tran, D. Dao, M. D. Nguyen, Q. V. Pham, B. O'Sullivan, and H. D. Nguyen, "Multi-agent collaboration mechanisms: A survey of llms," 2025.

[23] H. Kitano, "Nobel turing challenge: Creating the engine for scientific discovery," *npj Systems Biology and Applications*, vol. 7, p. 29, 2021.

[24] M. A. Ferrag, N. Tihanyi, and M. Debbah, "From llm reasoning to autonomous ai agents: A comprehensive review," 2025.

[25] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 6609–6625.

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