yAI SYSTEMS NOTE 005

The Chain Problem: Why Multi-Hop Queries Break RAG Systems

Version 1.0 April 13, 2026

Yassin Hafid
Founder & CEO
LinkedIn

ABSTRACT

Standard retrieve-then-generate RAG pipelines are effectively single-step: retrieve once, generate once. Most queries in knowledge-intensive domains are not. They require chaining evidence across multiple documents; this exposes a failure mode that retrieval quality alone cannot fix. The core finding is counterintuitive: in 73% to 84% of multi-hop errors in a Graph-RAG setting, the correct evidence was already in the retrieved context. The model failed to reason over it, not to retrieve it. Two structural mechanisms drive this: (1) decomposition instability: the quality of intermediate sub-questions determines whether the right evidence is ever found; and (2) error propagation: a wrong intermediate step silently corrupts all downstream reasoning. Both compound with hop depth. Success requires shifting from flat retrieval to reasoning-aware architectures built for multi-step evidence integration.

1. The Question

Standard RAG is designed for a single retrieval step. When a query requires connecting evidence across three or four documents in sequence (each document's relevance conditional on the previous step) what breaks first: retrieval or reasoning? And if the correct evidence is already in the retrieved context, why does the model still fail?

2. Scope and Definitions

> **Multi-Hop Query**
>
> A query whose correct answer requires integrating evidence from two or more distinct documents through sequential reasoning steps.
>
> **Hop Count**
>
> The number of sequential reasoning steps required to answer a query.
>
> **Semantic Distance**
>
> The degree of semantic divergence between a query and its evidence.
>
> **Lost-in-Retrieval**
>
> A multi-hop retrieval failure where a decomposed sub-query fails to retrieve the required intermediate evidence, propagating an evidence gap into all subsequent steps.[2]
>
> **Error Propagation**
>
> A failure mode in sequential RAG where an early incorrect step leads to flawed synthesis and invalidates the final answer, causing errors to cascade through the reasoning trace.[3]
>
> **Reasoning Bottleneck**
>
> The failure mode where the correct evidence is present in the retrieved context but the model fails to reason over it correctly.[5]
>
> **Decomposition Drift**
>
> The divergence of LLM-generated sub-questions from the logical structure required to retrieve the necessary intermediate evidence.[2][3]

This note focuses on RAG systems processing queries that require sequential evidence integration; this structure is particularly relevant in domains such as finance, regulation, and scientific research.

---

3. Key Findings

- **The Reasoning Bottleneck:** In 77% to 91% of multi-hop queries (HotpotQA, MuSiQue, 2WikiMultiHopQA), the "gold" evidence is present in the retrieved context, yet final accuracy for high-parameter models like Llama-3.3-70B remains between 35% and 78%.[5] This indicates that 73% to 84% of failures are internal reasoning errors; the model possesses the evidence but lacks the logical framework to synthesize it. Zarinkia et al. describe SPARQL-based structured prompting (SPARQL-CoT) and graph-walk context compression as inference-time mitigations that reduce context noise; in the reported setup, the fully augmented 8B model achieves performance levels comparable to the unaugmented 70B baseline.[5]
- **The Retrieval Ceiling:** Performance on benchmarks like MuSiQue and HotpotQA saturates at sub-0.50 recall (e.g., 0.46 on HotpotQA and 0.24 on MuSiQue) for dense retrievers.[1] This suggests a structural limitation of similarity-based retrieval for multi-hop queries; retrievers rank passages by semantic similarity rather than the conditional, logical relevance required to connect disparate documents. Liu et al. describe a retrieve-reason-prune cycle as an architectural mechanism for navigating these logical paths.[1]
- **Linear Chains vs. Reasoning Trees:** Sequential decomposition via linear chains is characterized by the lost-in-retrieval phenomenon; missing key entities in a sub-question disrupt the reasoning chain and propagate errors to all subsequent retrieval steps.[2] To mitigate this, Zhu et al. propose ChainRAG that utilizes a retrieve-and-rewrite strategy to recover missing entities within the linear sequence.[2] Alternatively, Reasoning Tree architectures (RT-RAG) utilize a hierarchical structure to explore multiple candidate paths simultaneously; it employs a consensus-based mechanism to minimize the error cascade.[3]
- **Sequential Error Propagation:** Multi-hop failures propagate through the reasoning trace via two structural drivers: (1) inaccurate decomposition, where the omission of key entities disrupts the reasoning chain;[2] and (2) sequential dependency,

where an early incorrect result cascades through the trace and invalidates the final synthesis.[3]

- **The 2D Difficulty Matrix:** Task difficulty is modeled along two orthogonal dimensions: Reasoning Depth (inference steps) and Semantic Distance (the proximity between query and evidence).[4] Error rates increase steadily from the upper-left cell (shallow hops, low semantic distance) to the bottom-right cell (deep hops, high semantic distance) of the matrix, with the steepest rise along the diagonal.[4]

---

4. Technical Deep Dive: Three Compounding Failure Mechanisms

A. The Retrieval Ceiling

Dense retrieval ranks passages by semantic or lexical similarity to the query. For multi-hop queries, intermediate evidence is conditionally relevant; its importance only becomes apparent once a prior step has been resolved. This conditional structure is invisible to similarity-based rankers. The result is a practical recall ceiling for similarity-based retrieval in multi-hop settings: the system struggles to surface evidence whose relevance only becomes visible after an earlier reasoning step has been resolved.[1]

B. Decomposition Instability and Lost-in-Retrieval

Iterative RAG handles multi-hop queries by decomposing them into sub-questions and retrieving evidence for each step sequentially. This decomposition is fragile. LLMs generating sub-questions often omit the key entities required to retrieve the required passage; this is more pronounced when intermediate evidence is semantically distant from the original question. A single lost-in-retrieval failure early in the chain disrupts the reasoning chain and propagates errors through every subsequent step. The original query is nominally answered; the actual answer is built on a missing foundation.[2]

C. The Reasoning Bottleneck

The most analytically significant finding is that retrieval success does not imply reasoning success. When the gold evidence is present in context, models still fail in the majority of error cases (e.g., failing to bridge the logical gap between facts).[5] The chain structure explains why: each reasoning step is a local, step-wise decision susceptible to error propagation;[3] errors do not self-correct across steps, rather they propagate and cascade, often invalidating the final synthesis.[3] Answer-level evaluation cannot locate where in the chain the failure occurred, masking whether the root cause is decomposition, retrieval, or reasoning.[3][5]

---

5. Practical Taxonomy of Multi-Hop Failure Modes

- **Retrieval Ceiling Failure:** Semantic similarity retrieval fails to capture the logical relevance of passages whose importance is conditional on an unresolved prior step. Primarily structural rather than easily fixed by ordinary retriever tuning alone. [1]
- **Lost-in-Retrieval:** The omission of key entities in a decomposed sub-question fails to retrieve required intermediate evidence, disrupting the reasoning chain.[2]
- **Decomposition Drift:** Generated sub-questions deviate from the reasoning coherence of the original query, producing an inaccurate decomposition chain.[2][3]
- **Error Propagation:** An incorrect intermediate result becomes a premise for the next step. The chain cascades early mistakes, invalidating the final synthesis rather than absorbing them.[3]
- **Reasoning Bottleneck:** The correct evidence is retrieved but the model fails to reason over it correctly. The majority of multi-hop errors fall here.[5]
- **Hop-Depth Degradation:** Error rates rise with Reasoning Depth (hop count), independent of semantic similarity difficulty.[4]

---

6. Implications for AI System Design

- **Do not diagnose multi-hop failure as a retrieval problem:** The majority of errors occur when the correct evidence is already retrieved. Improving retrieval alone will not resolve multi-hop failure.[5]
- **Treat query decomposition as a first-class failure point:** Decomposition quality determines every downstream retrieval step. It must be evaluated independently of final answer accuracy.[2][3]
- **Require step-level evaluation, not answer-level accuracy:** Answer-level metrics mask where in the chain failures occur. Distinguishing decomposition, retrieval, and reasoning failures requires step-level attribution.[3][4]
- **Calibrate accuracy expectations to hop depth:** Error rates grow consistently with hop count. Accuracy thresholds designed for single-hop queries do not transfer to deep reasoning chains.[4]
- **Treat semantic distance as a distinct risk axis:** Failure increases with both hop count and semantic distance. Both must be characterized for a given query distribution.[4]

---

7. Open Questions

- **Decomposition evaluation:** How can the quality of query decomposition be measured independently of final answer accuracy to enable early detection of decomposition drift before it propagates through the chain?
- **Logical relevance retrieval:** What retrieval representations would capture conditional relevance rather than semantic similarity to the original query?
- **Reasoning vs retrieval diagnosis:** Given that most multi-hop errors are reasoning failures, what inference-time signals distinguish a reasoning failure from a retrieval failure to enable targeted intervention at the correct layer?
- **Hop-depth threshold:** Is there an empirical hop-depth beyond which current RAG architectures fail systematically, regardless of retrieval quality or model capability?
- **Chain-aware retrieval:** Can retrieval be conditioned on the full chain state (not just the current sub-question) so that each step accounts for what has already been established in prior steps?
- **Intermediate answer verification:** Can intermediate conclusions be verified against retrieved evidence before becoming premises for the next step, preventing error propagation at its origin rather than detecting it at the output?
- **Decomposition-retrieval co-optimization:** Can sub-question generation be made jointly aware of retrieval system constraints to produce sub-questions that are both logically correct and retrievable?

---

8. References

[1] Liu et al., HopRAG: Multi-Hop Reasoning for Logic-Aware Retrieval-Augmented Generation, ACL 2025 (arXiv:2502.12442, 2025).

[2] Zhu et al., Mitigating Lost-in-Retrieval Problems in Retrieval Augmented Multi-Hop Question Answering, ACL 2025 (arXiv:2502.14245, 2025).

[3] Shi et al., Reasoning in Trees: Improving Retrieval-Augmented Generation for Multi-Hop Question Answering, arXiv:2601.11255, 2026.

[4] Lee et al., GRADE: Generating multi-hop QA and fine-gRAined Difficulty matrix for RAG Evaluation, EMNLP 2025 (arXiv:2508.16994, 2025).

[5] Zarinkia et al., The Reasoning Bottleneck in Graph-RAG: Structured Prompting and Context Compression for Multi-Hop QA, arXiv:2603.14045, 2026.

This note synthesizes findings from recent research on multi-hop reasoning failure in retrieval-augmented generation systems. The interpretations presented reflect the author's reading of the current literature.