

warning "Backend update (October 2026)" GitHub Models inference was retired on July 30, 2026. Any references below are retained only for historical context while this lab is migrated. Use Lab 013 ? Free LLM backends to configure Foundry Local or Ollama before continuing.

Lab 026: Agentic RAG Pattern

Level: L200 Path: RAG Time: ~40 min ? Cost: GitHub Free + Docker (local)

What You'll Learn

Why naive RAG fails on complex questions Query rewriting ? improving retrieval before searching Hypothetical Document Embeddings (HyDE) ? generate to retrieve Multi-hop RAG ? iterative retrieval for multi-part questions Self-reflection ? agent evaluates its own answer quality

Introduction

Naive RAG: embed query ? search ? generate. Works fine for simple questions. Fails for:

Vague questions: `"Tell me about it"` (what is "it"?) Multi-hop: `"What's cheaper ? camping or climbing gear? What would I save?"` Knowledge gaps: `"What's the newest product?"` (may need to know current date) Hallucination: model invents facts not in context

Agentic RAG adds reasoning loops around retrieval. The agent *decides* how to search, *evaluates* results, and *retries* if needed.

Prerequisites

Completed Lab 022 (pgvector running + documents ingested) GITHUB_TOKEN set pgvector container running: `docker start pgvector-rag`
tip "Sample data already loaded?" If you ran Lab 022's ingest step with the sample dataset, you already have 42 documents in pgvector ready for this lab. If not, run Lab 022's Step 3 first.

Lab Exercise

Step 1: Query rewriting

Before searching, ask the LLM to rewrite the user's question into better search queries.

[diagram or code sample]

Step 2: Retrieve with multiple queries and deduplicate

[diagram or code sample]

Step 3: HyDE ? Hypothetical Document Embeddings

Instead of embedding the question, generate a *hypothetical answer* and embed that. This often matches the real document better.

[diagram or code sample]

Step 4: Multi-hop RAG

For complex questions, retrieve ? generate partial answer ? retrieve again.

[diagram or code sample]

Step 5: Self-reflection (answer quality check)

[diagram or code sample]

Agentic RAG Architecture Summary

[diagram or code sample]

Next Steps

Agent memory across sessions: ? Lab 027 ? Agent Memory Patterns Evaluate RAG quality at scale: ? Lab 035 ? Agent Evaluation

Supporting kit

The site build generates stable PDF and ZIP downloads. The ZIP contains the lab guide and every supporting file declared for this lab.

Download generated PDF Download generated ZIP

Supporting file: CHEATSHEET.md

Supporting files: Lab 026: Agentic RAG Pattern

Start with the documented local or free path. Record assumptions and validation evidence.

Keep hosted services optional unless the lab marks them required.

Supporting file: README.md

Lab 026: Agentic RAG Pattern

Use these files with the lab guide and generated kit.