

Enhancing Semantic Search with Retrieval-Augmented Generation and Agentic AI

Francisco Azeredo

Instituto Superior Técnico, Universidade de Lisboa

MSc Thesis Defense



Presentation Outline

- ① Problem & Motivation
- ② Objectives
- ③ System Architecture
- ④ Evaluation & Results
- ⑤ Conclusions & Future Work

Research Problem

Core Challenge

How to handle **interdependent information** in enterprise settings where information is **dynamic** and later reports can nullify earlier ones?

Key gaps in existing solutions:

- Keyword search ignores semantic meaning and document structure
- Naive RAG retrieves isolated information without considering dependencies
- Cannot track how information relates to, updates, or contradicts each other
- No standard tools for agents to traverse information relationships

Thesis Objectives

Main goal: Handle information interdependencies through graph-based agentic reasoning.

Specific objectives:

- ① Design a schema that captures enterprise document structure and relationships
- ② Implement GraphRAG: knowledge graph construction from document corpus
- ③ Develop agentic graph traversal for multi-hop reasoning over dependencies
- ④ Create an MCP server for standardized agent-database interaction
- ⑤ Evaluate against naive RAG on enterprise-like scenarios

Motivation: Why Corporate Search Fails

Traditional keyword search limitations:

- Cannot capture context, semantics, or document structure
- Users must guess exact terms and file locations
- Ignores relationships between information pieces

Naive RAG limitations in enterprise settings:

- Retrieves relevant information but **ignores interdependencies**
- Cannot detect when newer reports nullify or update older information
- Treats information store as static, missing **dynamic information flow**
- High risk: outdated or contradicted information in critical decisions

The Document Interdependency Problem

Example scenario:

- **Report A (Jan 2024):** "Vendor X approved for contracts up to \$100K"
- **Report B (Mar 2024):** "Vendor X approval suspended due to audit"
- **Query (Apr 2024):** "Can we contract with Vendor X?"

Naive RAG problem:

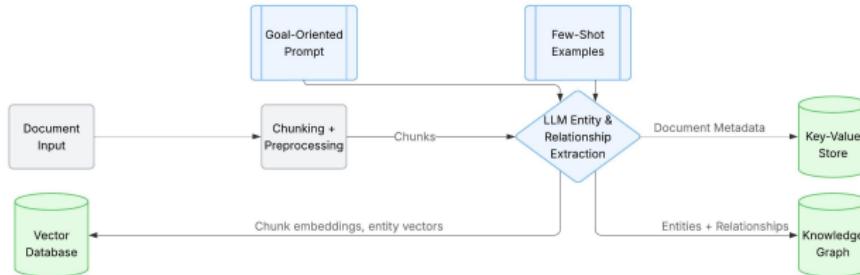
- May retrieve Report A (highly relevant to query)
- Misses that Report B nullifies Report A
- Returns **accurate but outdated** information → incorrect answer

Solution needed: Graph-based reasoning to traverse information relationships

System Architecture Overview

Three-layer architecture:

- ① **Processing:** OCR, chunking, embeddings, metadata extraction
- ② **Storage:** Vector database (Weaviate) + Knowledge graph for information relationships
- ③ **Reasoning:** ReAct agent with schema-aware tools (MCP)

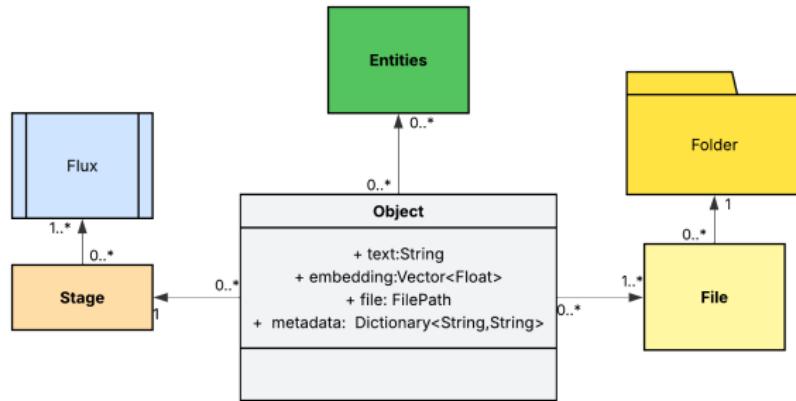


Schema-Aware Data Model

Six core classes model enterprise document structure:

- **Fluxo** (Workflow), **Etapa** (Step): process structure
- **Entidade** (Entity): companies, people, products
- **Pasta** (Folder), **Ficheiro** (File), **Metadados** (Metadata)

Key benefit: Cross-references enable both semantic search *and* deterministic multi-hop traversal.

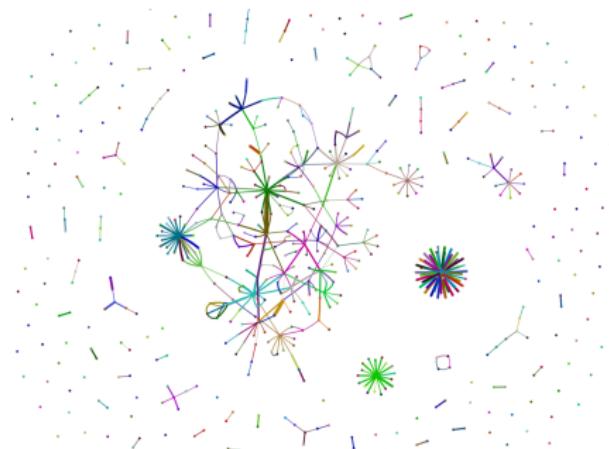


Knowledge Graph Construction (MiniRAG)

Lightweight GraphRAG implementation:

- LLM extracts key entity words from document chunks
- Nodes with matching entity words are merged
- Creates graph structure connecting related information
- Requires supervision but aids in processing large document sets

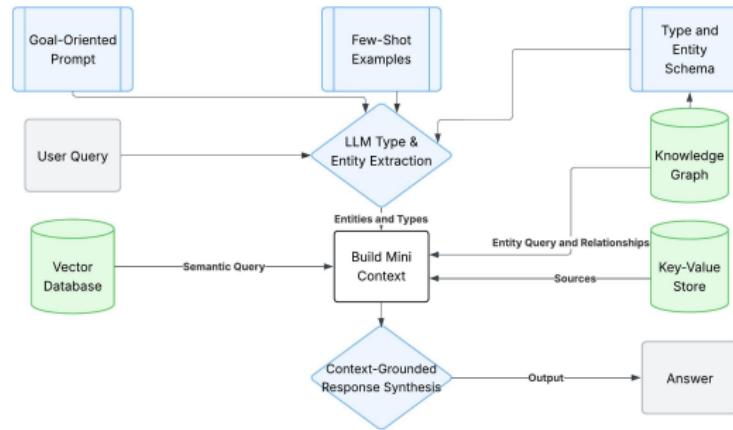
Goal: Facilitate discovery of information relationships across documents through entity-based graph structure.



Agentic Graph Traversal (Mini Query Mode)

Multi-hop reasoning over knowledge graph:

- ① Extract entities and temporal context from query
- ② Retrieve relevant information via vector search
- ③ **Traverse graph** to find related/updating information
- ④ Build context considering dependencies and temporal order
- ⑤ Generate answer from complete, up-to-date evidence

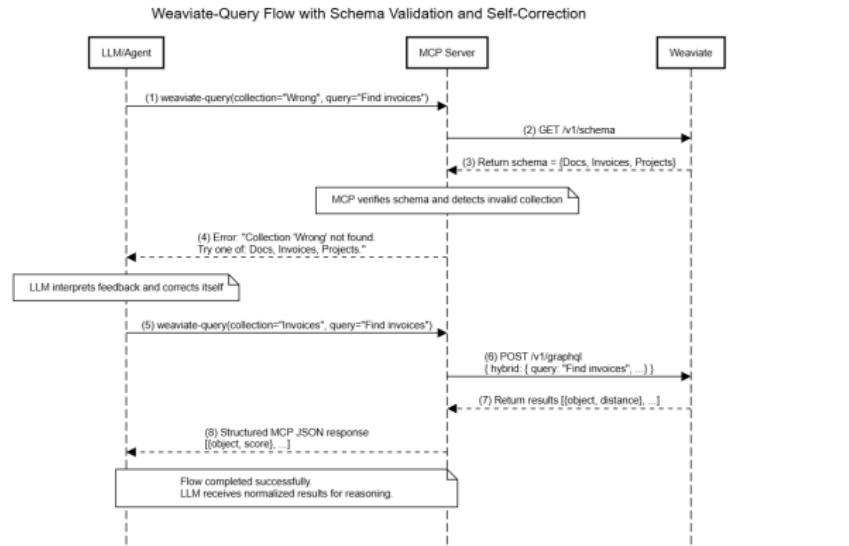


Weaviate MCP Server: Agent-Database Bridge

Model Context Protocol (MCP) standardizes agent tool interfaces.

Our contribution: First MCP server for Weaviate

- Exposes tools: `weaviate-query`, `weaviate-follow-ref`, etc.



Results: Retrieval Performance

Experiment: Single-collection retrieval on enterprise-like dataset

Key findings:

- **Agentic RAG:** 60.4% retrieval rate, 61% correct answers
- Outperforms all baselines by large margin (2-4×)
- Agentic approach combines multi-step reasoning with hybrid search

Table: Document retrieval and answer quality metrics

Method	Avg. Retrieval Rate	Correct Answers
Agentic RAG	60.4%	61%
LexRank	33.6%	29.5%
BART	17.8%	3.0%
Naive RAG	13.8%	19.1%

"Correct Answers" = answers passing Youden-optimized quality threshold

Results: Multi-Collection Retrieval

Experiment: Retrieval across heterogeneous document collections

Key findings:

- Agent successfully selects appropriate collection for each query
- All agentic methods outperform naive baseline (2-3x)
- Different strategies (REST API, LexRank, DOCX) suit different content

Table: Multi-collection retrieval and answer quality metrics

Method	Avg. Retrieval Rate	Correct Answers
Mixed REST	46.5%	15.8%
Mixed LexRank	39.4%	24.6%
Mixed DOCX	33.0%	31.5%
Naive RAG	13.8%	19.1%

Demonstrates generalization to diverse document types and sources

Main Contributions

- ① **Problem identification:** Formalized the information interdependency challenge in enterprise semantic search
- ② **GraphRAG architecture:** Schema-aware system capturing information relationships, updates, and temporal dependencies
- ③ **Agentic graph traversal:** Multi-hop reasoning framework to handle dynamic information and nullification
- ④ **MCP server:** First standardized tool for agents to query Weaviate with schema validation
- ⑤ **Evaluation:** Demonstrated 2-4× improvement over naive RAG on enterprise scenarios

Conclusions

- **GraphRAG with agentic traversal** handles information interdependencies that naive RAG cannot (60% vs 14-34% retrieval accuracy)
- **Multi-hop reasoning** over knowledge graphs enables tracking how information updates, supersedes, or contradicts earlier information
- Successfully addresses **dynamic information storage** problem in enterprise settings
- MCP server provides standardized, validated access to Weaviate for any LLM agent
- Demonstrates importance of **information structure and relationships** beyond semantic similarity

Limitations & Challenges

Experimental limitations:

- Computational and budget constraints limited scale of experiments
- No public benchmark exists for structure-aware enterprise retrieval

Open challenges:

- LLM hallucinations not fully eliminated (though greatly reduced)
- Privacy and security considerations in enterprise deployment
- Scalability to millions of information pieces requires further optimization

Future Work

- Evaluate stronger models and longer context windows.
- Develop public benchmarks for structure-aware retrieval.
- Explore advanced reasoning (multi-agent, graph reasoning, GraphRAG).
- Integrate with enterprise systems and enforce ACLs at scale.
- Scale to millions of information pieces with optimized graph traversal.

Thank You

Thank you!

Francisco Azeredo
Instituto Superior Técnico

Questions?