

# Enhancing Semantic Search with Retrieval-Augmented Generation and Agentic AI

Francisco Azeredo

Academic Advisor: Prof. Sérgio Luís Proença Duarte Guerreiro

Industrial Advisor: Eng.º Filipe Mendes Correia (Link Consulting SA)

Instituto Superior Técnico, Universidade de Lisboa

MSc Thesis Defense



# Presentation Structure

- 1 Motivation and Problem
- 2 Objectives
- 3 Architecture and Components
- 4 Related Work (Context)
- 5 Evaluation and Results
- 6 Conclusions and Future Work

## Critical Business Context:

- Enterprises rely on *accurate, current* information for project approvals, compliance decisions, and operational governance
- High-stakes work:** Incorrect or outdated information can lead to regulatory violations, project failures, or financial losses

## Current Problems:

- Professionals spend significant time searching, reading, and analyzing multiple documents
- Manual verification of document dependencies creates **high cognitive load**
- Decisions risk relying on superseded or nullified information
- Even modern semantic search systems fail** in critical situations—they cannot guarantee information currency or detect contradictions

## Document Management Platform:

- Organizes documents with structured workflows and business rules
- Supports evolution from ad-hoc to complex workflow configurations
- Enables end-to-end process automation and digitalization
- Features: document lifecycle management, collaborative work, ERP integration
- Used in public sector and enterprises with rich, complex processes

**Thesis Opportunity:** Leverage this rich structure to enhance semantic search and retrieval accuracy.

# Problem: Example Scenario

## Example Timeline

- Jan: Approve Vendor X (doc A)
- Mar: Suspend Vendor X (doc B)
- Apr Query: “Can we contract Vendor X?”

**Risk:** Retrieves A, ignores B..

**Core Challenge:** Capture and reason over document dependencies.

# Research Gap

- No standard tooling for dependency-aware retrieval
- GraphRAG ideas emerging; enterprise schemas underused
- Agents lack validated schema-constrained traversal tools
- Need: Hybrid semantic + structural + temporal reasoning

# Objectives Overview

- **Represent** enterprise document structure and relations
- **Detect** updates / contradictions across documents
- **Reason** with agents over a knowledge graph
- **Standardize** access (MCP server for Weaviate)
- **Evaluate**: compare naive and baseline RAG

# Thesis Objectives

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

**Secondary goal:** Scalable, optimized storage solution. Efficient document processing (if required). Good explainable query accuracy.

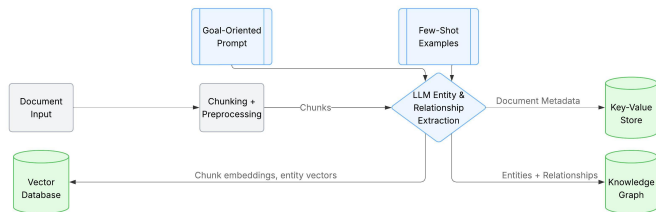


# Objective Details

- 1 Schema design (workflows, entities, metadata)
- 2 Graph construction (entity merging, references)
- 3 Agent traversal (multi-hop, reasoning)
- 4 MCP tools (validated queries, follow references)
- 5 Evaluation (retrieval accuracy, answer quality)

# Architecture Overview

- 1 Ingestion (OCR, chunking, embeddings)
- 2 Storage (Vector DB + Graph)
- 3 Reasoning (Agent + MCP tools)

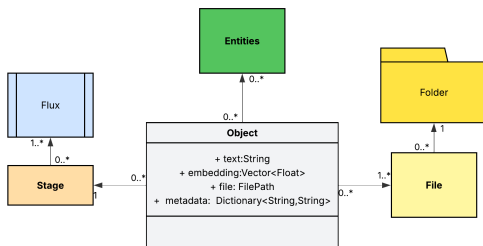


# Schema Data Model

Six core classes:

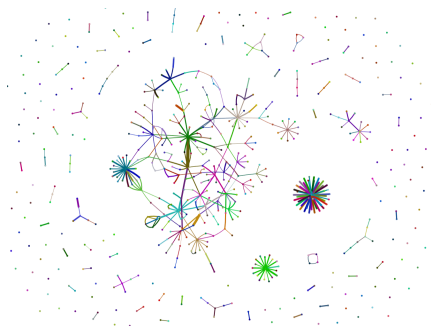
- Workflow (Fluxo), Step (Etapa)
- Entity (Entidade)
- Folder (Pasta), File (Ficheiro), Metadata (Metadados)

Cross-references enable semantic + deterministic traversal.



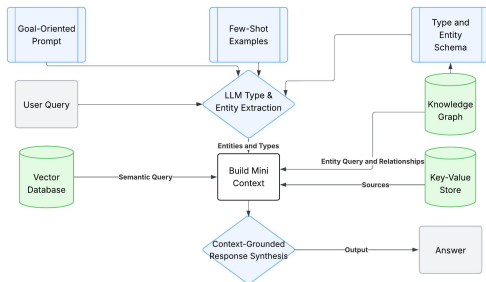
# Graph Construction

- Extract entities from chunks (LLM assisted)
- Merge nodes with matching entities
- Maintain references (updates, supersedes)
- Output: lightweight knowledge graph



# Agent Traversal Flow

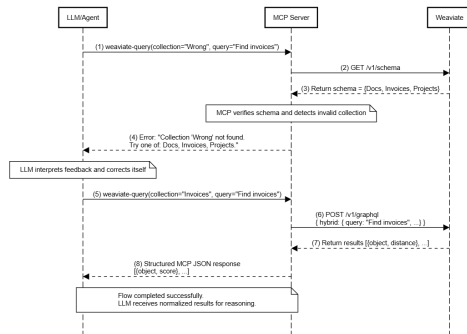
- 1 Parse query (entities + time)
- 2 Hybrid retrieval (vector + filters)
- 3 Follow graph refs (updates/nullifications)
- 4 Assemble consistent evidence set
- 5 Generate answer (current state)



# MCP Server (Weaviate Bridge)

- Standard tool interface for agents
- Validated schema-aware queries
- Tools: weaviate-query, weaviate-follow-ref
- Enables safe multi-hop reasoning

Weaviate-Query Flow with Schema Validation and Self-Correction



# Control Perspective (Robotics/AI)

- Retrieval = perception layer
- Graph = system state representation
- Agent traversal = planning refinement loop
- Updates = state correction (feedback)
- Ensures consistent decision inputs

# Related Work: Semantic and Graph RAG

- Keyword enterprise search: lacks semantics/structure
- Classic RAG: isolated chunk retrieval
- Emerging GraphRAG: entity/relationship surfacing
- Gap: temporal nullification + standardized tooling



# Comparison to Prior Approaches

- Adds temporal precedence handling
- Integrates schema-driven traversal
- Standardizes graph access (MCP server)
- Focus: enterprise interdependency reliability

# Evaluation Questions

- Does graph traversal improve retrieval accuracy?
- Does dependency handling improve answer correctness?
- Does hybrid search generalize across collections?

# Single-Collection Performance

**Agentic RAG:** 60.4% retrieval, 61% correct answers ( $\sim 2-4\times$  over baselines)

**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%

Threshold via Youden optimization.

## Generalization across heterogeneous sources

- Correct collection selection
- 2–3× improvement across strategies
- Robust to format variation

Approach	Avg. Retrieval Rate
Agent_OpenAI_MixedLiHua	93.0%
Agent_OpenAI_Mixed	86.0%
Agent_OpenAI_MixedSynthetic	80.5%
Agent_OpenAI_300	60.4%

**Table:** Share of questions with the correct document retrieved, by result file

# Key Result Insights

- Multi-hop reasoning reduces outdated answers
- Temporal ordering prevents contradiction leakage
- Graph adds structured context beyond embeddings

# Main Contributions

- 1 Formalized enterprise interdependency retrieval problem
- 2 Schema + graph architecture for dynamic updates
- 3 Agentic multi-hop traversal (temporal aware)
- 4 MCP Weaviate server (standardized tools)
- 5 Empirical gains (2–4× over baselines)

- **Graph-aware reasoning:** prevents outdated answers
- Temporal ordering: resolves contradictions
- Schema constraints: reduce hallucination surface
- Tool standardization enables reproducible agent pipelines

- **Graph-aware reasoning:** prevents outdated answers
- **Temporal ordering:** resolves contradictions
- **Schema constraints:** reduce hallucination surface
- **Tool standardization** enables reproducible agent pipelines



# Conclusions



- **Graph-aware reasoning:** prevents outdated answers
- **Temporal ordering:** resolves contradictions
- **Schema constraints:** reduce hallucination surface
- **Tool standardization** enables reproducible agent pipelines

# Conclusions

- **Graph-aware reasoning:** prevents outdated answers
- **Temporal ordering:** resolves contradictions
- **Schema constraints:** reduce hallucination surface
- **Tool standardization** enables reproducible agent pipelines

# Limitations

- Scale constrained (compute / budget)
- No public benchmark (construction overhead)
- Residual hallucinations possible
- Need deeper privacy/ACL integration

# Future Work

- Larger-scale graph + incremental updates
- Benchmark release (structure-aware retrieval)
- Multi-agent verification and consistency checks
- Integration with access control layers
- Publication submission (CONF NAME / JOURNAL TBD)

# Thank You

## Thank you!

Francisco Azeredo  
Instituto Superior Técnico  
Questions?