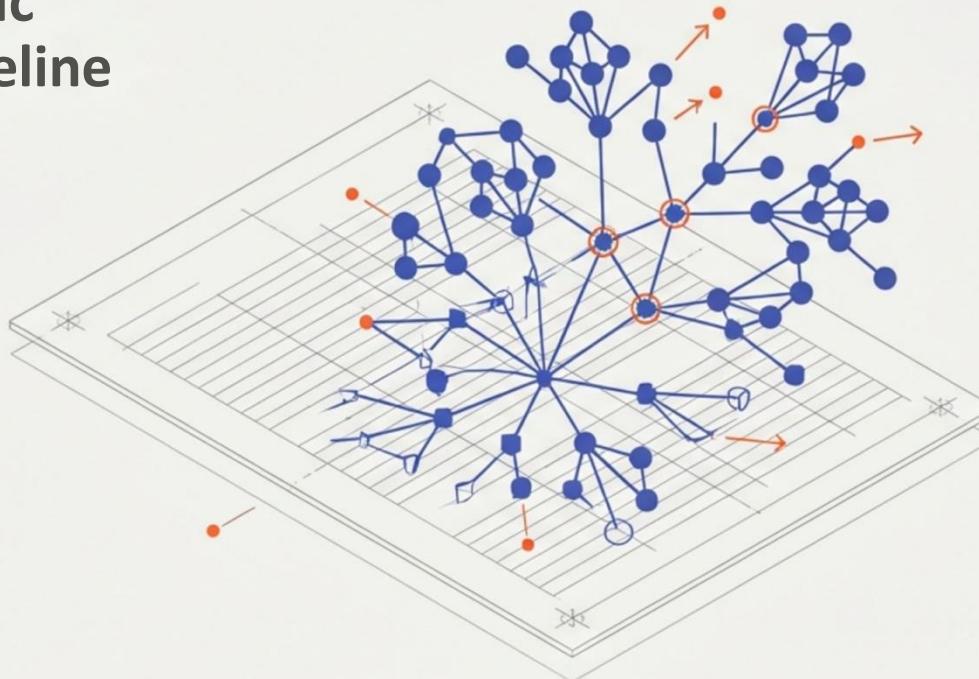


PROJEKT IMARA

Domain-specific GraphRAG Pipeline



MODUL:
AUTOREN:
DATUM:

Abschlussarbeit CAs Machine Learning for Software Engineers (ML4SE)
Marco Allenspach, Lukas Koller, Emanuel Sovrano
17.01.2026

Evaluierung von AI-Native GraphRAG Strategien und
Model Fine-tuning zur Überwindung der Grenzen
klassischer RAG-Systeme.

Abstract: Vektor vs. Graph

Ausgangslage

Naive RAG-Systeme stossen bei komplexen, mehrschrittigen Anfragen (Multi-Hop) an Grenzen. Es fehlt die explizite Modellierung von Beziehungen.

Zielsetzung

Aufbau einer End-to-End-Pipeline (PDF zu Eval) zum Vergleich von reinem Vektor-RAG mit GraphRAG-Varianten (LinearRAG, LeanRAG, GraphMERT).

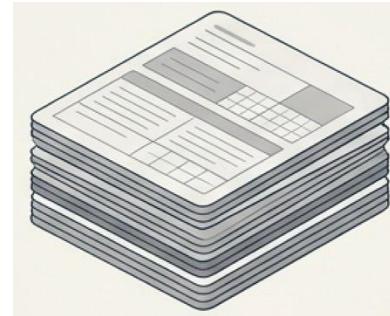
Ergebnis

LinearRAG beweist hohe Skalierbarkeit und Faktentreue ("Groundedness"), erfordert jedoch signifikant höhere Engineering-Investitionen als die Baseline.

Datenbasis

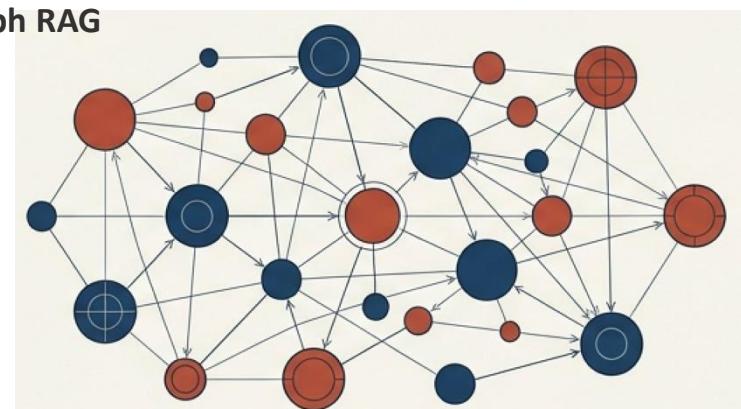
OpenRAGBench (Wissenschaftliche PDFs).

Vector RAG



Vektor (Isoliert)

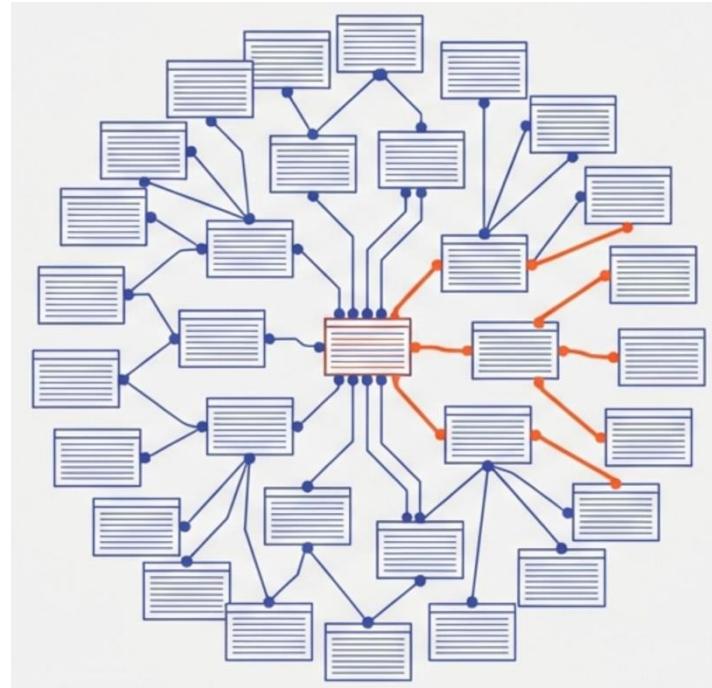
Graph RAG



Graph (Verbunden)

Das Paradigma 'AI-Native GraphRAG'

Struktur als Kontext: Transformation passiver Daten in ein aktives Modell.



Die Grenzen des naiven, vektorbasierten RAG

1. Kontextuelle Fragmentierung:

Chunking zerschneidet den natürlichen Informationsfluss. Zusammenhänge über Abschnittsgrenzen hinweg gehen verloren.

2. Implizite Beziehungen:

Vektorsuche findet Ähnlichkeiten, versteht aber keine Kausalität oder Hierarchien.

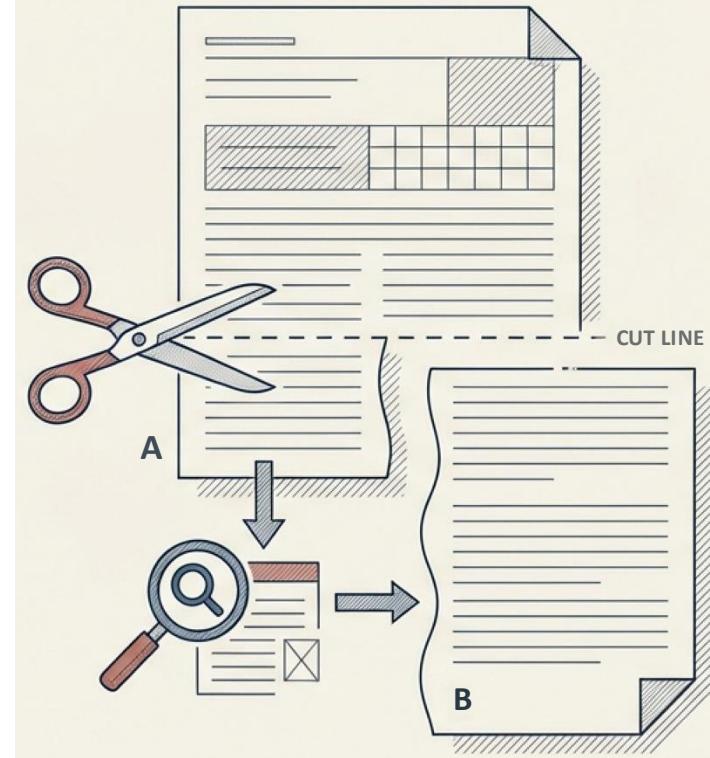
3. Versagen bei Multi-Hop-Reasoning:

Komplexe Fragen, die Logiksprünge über mehrere Dokumente erfordern ($A \rightarrow B \rightarrow C$), können nicht zuverlässig beantwortet werden.

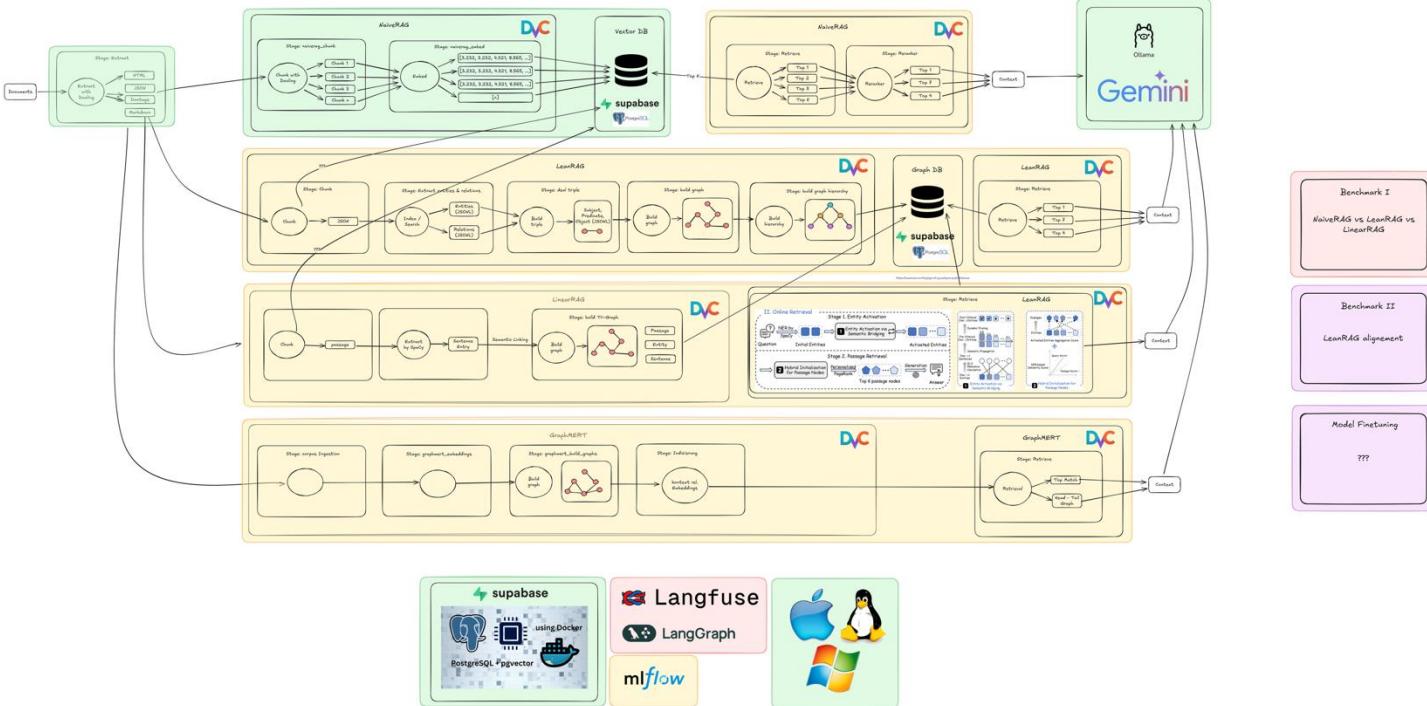
4. Abhängigkeit:

Die Qualität steht und fällt mit der heuristischen Chunking-Strategie.

Fragmentierung



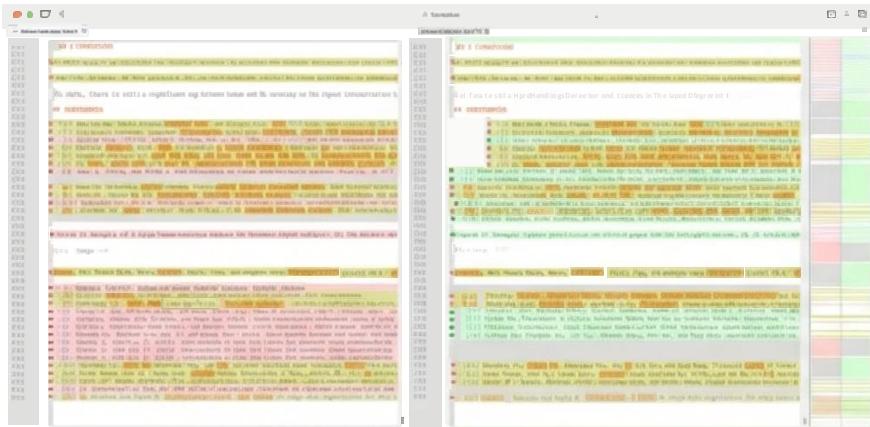
Systemarchitektur & Tech Stack



Stack: DVC, MLflow, Supabase, PostgreSQL (pgvector), Docling

Die Docling-Herausforderung

Standard Parameter (Fehlerhaft)



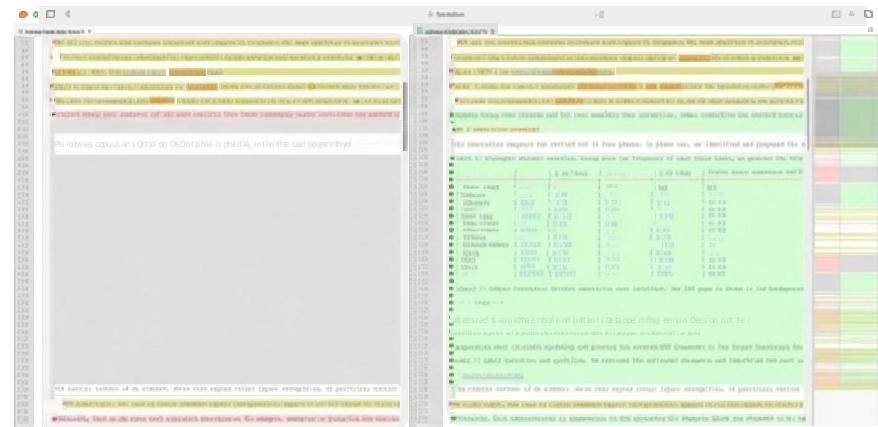
Problem:

Wissenschaftliche PDFs enthalten komplexe Tabellen und Formeln. Standard-Settings führen zu Datenverlust.

Kosten:

Hoher Ressourcenbedarf (GPU/CPU). Das Parsen von Formeln ist der Flaschenhals.

Optimierte Parameter (Strukturiert)



Lösung:

Iterative Optimierung der Parameter
(`do_table_structure=True`
`do_formula_enrichment=True`).

Erkenntnis:

Die Extraktionsqualität definiert die Obergrenze der RAG-Leistung.

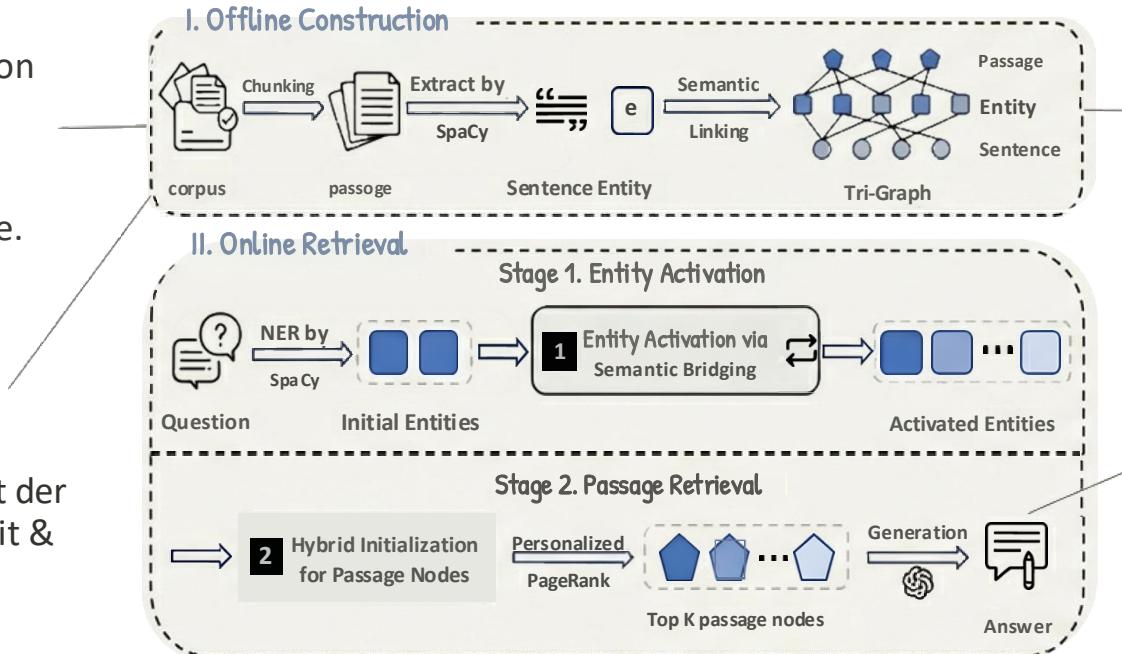
LinearRAG (Konzept)

LLM-frei:

Graph-Konstruktion erfolgt rein algorithmisch (scispacy), ohne teure LLM-Aufrufe.

Lineare Komplexität:

Skaliert linear mit der Korpusgrösse (Zeit & Speicher).



Tri-Graph Struktur:

Knoten-Typen für Passage, Satz (Sentence) und Entität (Entity).

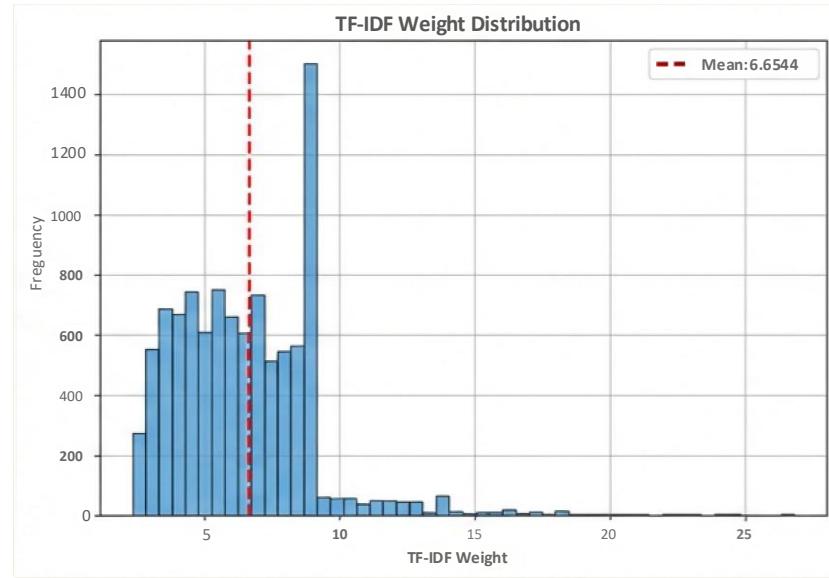
Entität (Entity).

Hybrid Retrieval:
Kombiniert
Vektorsuche mit
PageRank-basierter
Graph-Traversierung.

Stack: DVC, MLflow, Supabase, PostgreSQL (pgvector), Docling.

LinearRAG: Graphstruktur & Metriken

Metrik	Wert
Knoten (Total Nodes)	~1.75 Millionen
Kanten (Total Edges)	~7.0 Millionen
Sparsität (Sparsity)	99.9995%
Passages pro Entity	4.34 (avg)



Symmetrische Verteilung der Kanten-Gewichtung (TF-IDF).

Validierung:

Die extreme Sparsität bestätigt die Effizienz des relation-freien Ansatzes. Die TF-IDF-Verteilung beweist eine ausgeglichene Relevanz-Modellierung ohne 'Super-Nodes', die das Retrieval verzerren würden.

GraphMERT (Neurosymbolisch)

Skalierbare Destillation zuverlassiger Wissensgraphen

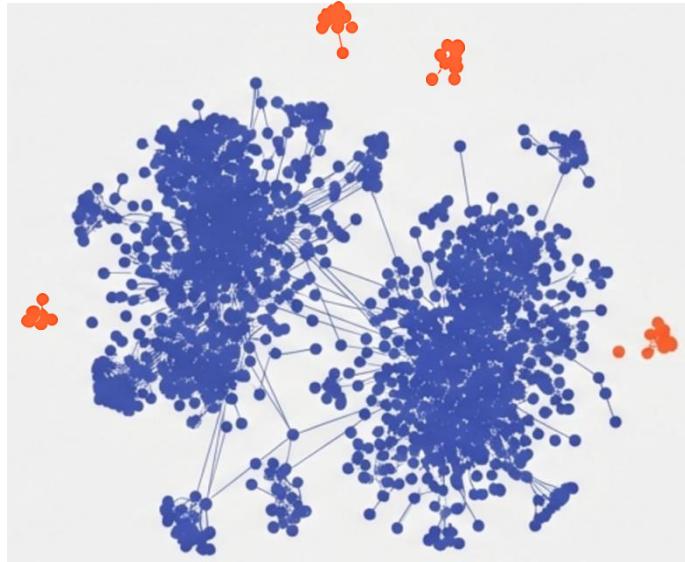
Konzept:

Neurosymbolischer Stack -Neuronales
Lernen von Abstraktionen + Symbolische
Graphen fur verifizierbares Schliessen.

Performance:

Erreichte einen **FActScore** von **69.8%** bei
medizinischen Texten (vs.40.2% bei
Basis-LLMs).

Ziel: Erstellung "faktisch korrekter"
(Provenance) und "valider"
(Ontologie- konsistenter) Graphen.



Node Embeddings

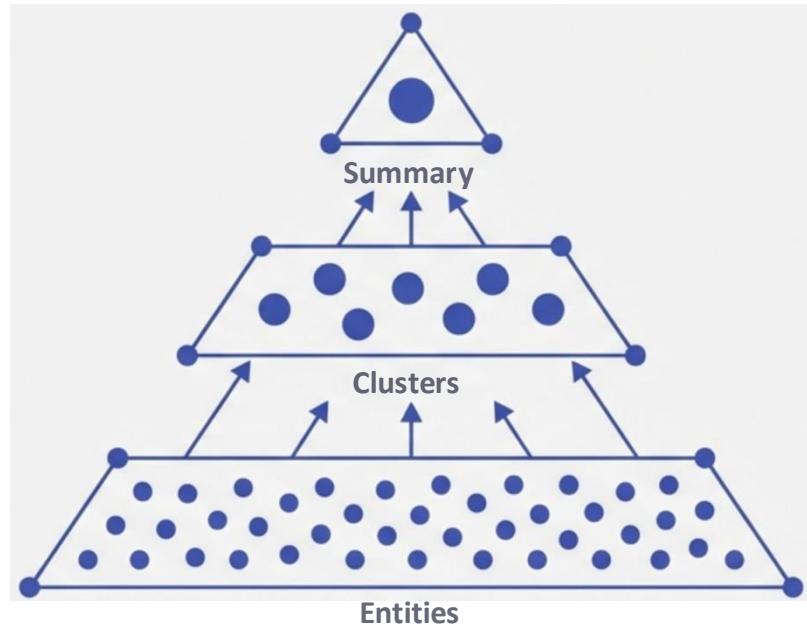
Ansatz 1: LeanRAG (Hierarchisch)

Reduktion von Redundanz durch semantische Aggregation.

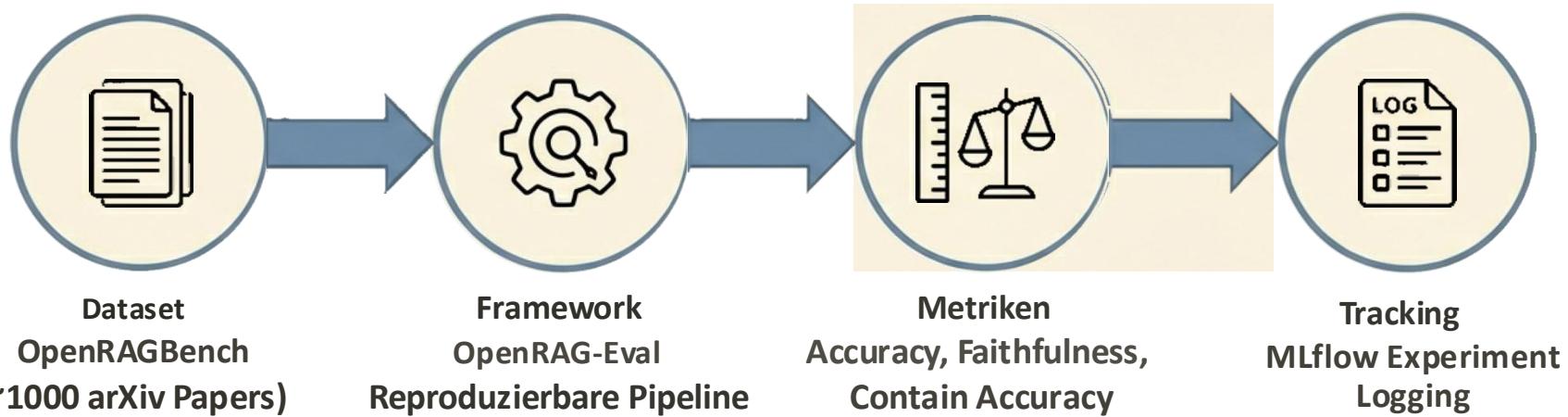
Semantic Aggregation: Gruppierung von Low-Level-Entitäten in semantisch kohärente Cluster/Zusammenfassungen.

Traversal Strategy: Hierarchisches, strukturgeleitetes Retrieval ('Bottom-up' von der Entität zum Cluster).

Impact: Reduziert Retrieval-Redundanz um ca. 46% im Vergleich zu flachen Baselines.



Evaluations-Methodik



Detail Focus

Metrik-Definitionen:

- **Accuracy:** Korrektheit der Antwort (LLM Judge).
- **Faithfulness:** Basiert die Antwort halluzinationsfrei auf den Fakten?
- **Contain Accuracy:** Wurde der relevante Kontext im Retrieval gefunden?

Benchmark Resultate: Linear vs. Naive

results.json ×				results2.json ×				
Retrieval		Generation		Retrieval		Generation		
▶ Expand all	47% Relevance	82% Groundedness	66% Factuality	0% Citations	55% Relevance	100% Groundedness	26% Factuality	0% Citations
▶ How many audio files are in the VoxCeleb 2 development set?	33% Relevance	100% Groundedness	97% Factuality	0% Citations	40% Relevance	100% Groundedness	4% Factuality	0% Citations
▶ What specific design improvements might further reduce formula merging and splitting errors?	20% Relevance	100% Groundedness	63% Factuality	0% Citations	33% Relevance	100% Groundedness	73% Factuality	0% Citations
▶ What kind of utility function represents the broker's preference?	73% Relevance	80% Groundedness	68% Factuality	0% Citations	73% Relevance	100% Groundedness	16% Factuality	0% Citations
▶ How does increased stock price volatility influence the optimal trading strategy?	33% Relevance	100% Groundedness	64% Factuality	0% Citations	53% Relevance	100% Groundedness	21% Factuality	0% Citations
▶ Why was a 62.5 grid size preferred over 31.25 despite lower error metrics?	33% Relevance	83% Groundedness	90% Factuality	0% Citations	27% Relevance	100% Groundedness	20% Factuality	0% Citations
▶ What machine learning algorithm was used in the pipeline?	47% Relevance	75% Groundedness	33% Factuality	0% Citations	67% Relevance	100% Groundedness	11% Factuality	0% Citations
▶ What models are leveraged for optimal order execution?	80% Relevance	31% Groundedness	36% Factuality	0% Citations	80% Relevance	100% Groundedness	7% Factuality	0% Citations

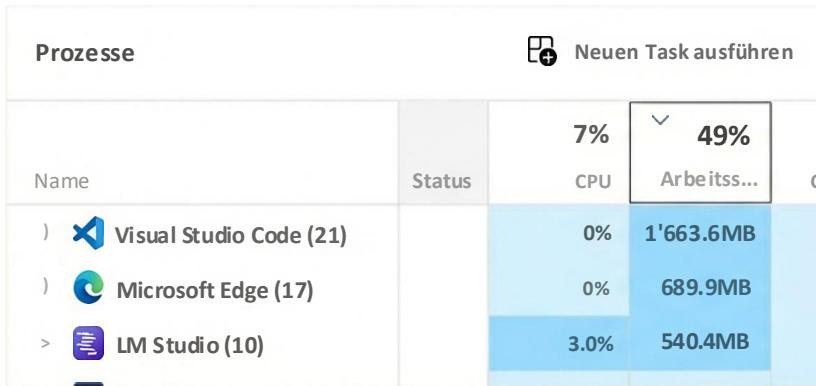
Diskussion: Qualität vs. Aufwand

System	Vorteile (Pro)	Nachteile (Contra)
Naives RAG	Einfaches Setup, geringe Kosten.	Kontext-Fragmentierung, schwach bei Logik-Sprüngen.
LinearRAG	Hohe Faktentreue, skaliert linear, kein LLM-Bottleneck bei Konstruktion.	Hoher Engineering-Aufwand, komplexes Tooling.

Fazit: LinearRAG lohnt sich für grosse, wissensintensive Korpora, wo Halluzinationen inakzeptabel sind.

Lessons Learned: Engineering Challenges & Risiken

Visual Evidence



Hoher Ressourcenbedarf während der Graph-Konstruktion.

- **Ressourcen**

Enormer Strom- und Speicherbedarf (bis zu 3000 kWh, >64GB RAM nötig). 16GB VRAM waren oft unzureichend.

- **Tooling-Komplexität**

Abhangigkeits-Konflikte (Windows vs. Linux, Dvc, Supabase, Docker-Container).

- **Zeitfaktor**

PDF-Extraktion (insb. Formeln) dauert Stunden pro Dokument.

- **Lesson Learned**

'Establish an End-to-End slice early.' - Frühzeitige Integration ist kritisch.

Ausblick & Weiterentwicklung



Datenqualität & Entity Resolution:
Bessere Auflösung von Entitäten



Temporale Graphen
Integration von Zeit und Ort als Dimensionen im Graphen



Hypergraphen
Nutzung von Hyperkanten für komplexes Relation-Clustering



Mehrsprachigkeit
Konzeptgraphen statt reiner Übersetzung



DVC-Integration
Vollständige Orchestrierung von OpenRAG-Eval via DVC-Pipelines

Fazit

- Zielerreichung: Eine reproduzierbare GraphRAG-Pipeline (PDF-to-Eval) wurde erfolgreich aufgebaut.
- Performance: LinearRAG bestätigt theoretische Vorteile (Sparsität, Skalierbarkeit) in der Praxis.
- Trade-off: GraphRAG ist kein "Free Lunch". Der Gewinn an Qualität wird mit hoher technischer Komplexität bezahlt.

Persönliche Erkenntnisse

"Gutes Requirements Engineering ist essenziell. Der Ressourcenhunger war überraschend." - M. Allenspach
"Komplexität explodiert mit LLMs. Iteratives Vorgehen ist Pflicht." - L. Koller
"Infrastruktur limitiert Skalierung.
Benchmarks waren lebenswichtig." - E. So

Projektteam IMARA

Marco Allenspach

Lukas Koller

Emanuel Sovrano

CAS Machine Learning for Software Engineers 2025/2026
Ostschweizer Fachhochschule

