



Singapore, May 21

The Power of GraphRAG - Architectures and Patterns

Combine Knowledge Graphs with LLMs for Accurate,
Complete, and Explainable AI

Michael Hunger

VP Product Innovation @ Neo4j
GenAI/Cloud/DX



15+ years of graph experience

Main Takeaway - graphrag.com

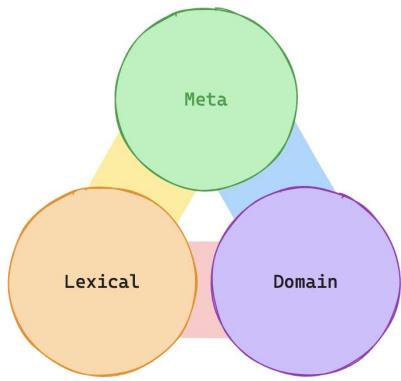
- Graph Models
- GraphRAG Retrieval Patterns
- Papers
- Open for Contribution

GraphRAG with a Knowledge Graph

Connect the dots for better answers

[What is GraphRAG? →](#)

[What is a Knowledge Graph? →](#)



Challenges with GenAI

| | PARROT | CHATGPT |
|--|--------|---------|
| Learns random sentences from random people | ✓ | ✓ |
| Talks like a person but doesn't really understand what it's saying | ✓ | ✓ |
| Occasionally speaks absolute non sense | ✓ | ✓ |
| Is a cute little bird | ✓ | ✗ |

- Lack of enterprise domain knowledge
- Inability to verify or explain answers
- Hallucination
- Ethical and data bias concerns

Challenge

How can you use
domain-specific knowledge
to rapidly build **accurate,**
contextual, and **explainable**
GenAI applications?

How to Help LLMs Do Better?

All of these are useful, but grounding is where **data** adds value

Fine-Tuning

Provide additional training data to better tune GenAI to your use case

Few-Shot Learning

Provide completed examples “shots” to the AI as context in prompts. *a.k.a In-Context Learning*

Grounding

Provide AI with the information to use for generating responses e.g. by using RAG

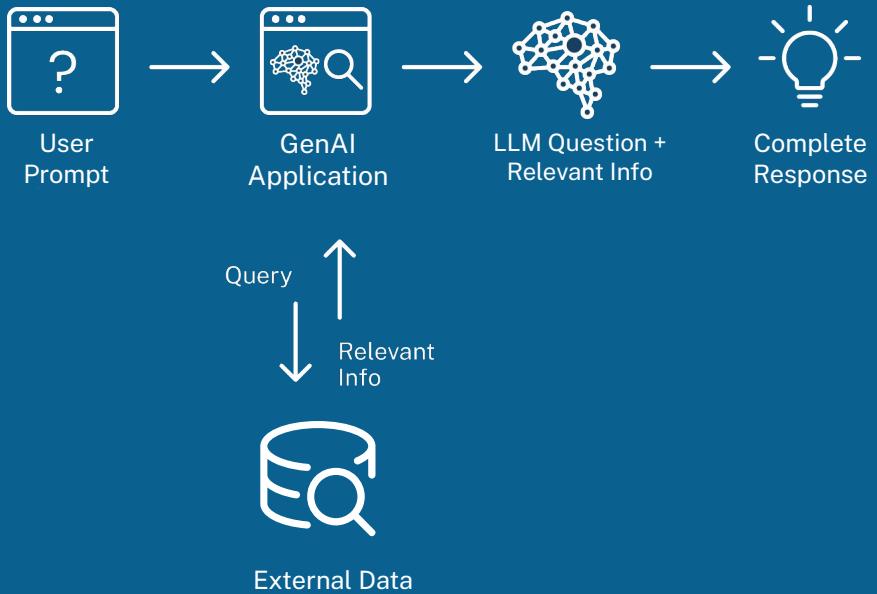


RAG - Retrieval Augmented Generation

Enhance & focus your LLM prompt with data
from trusted sources

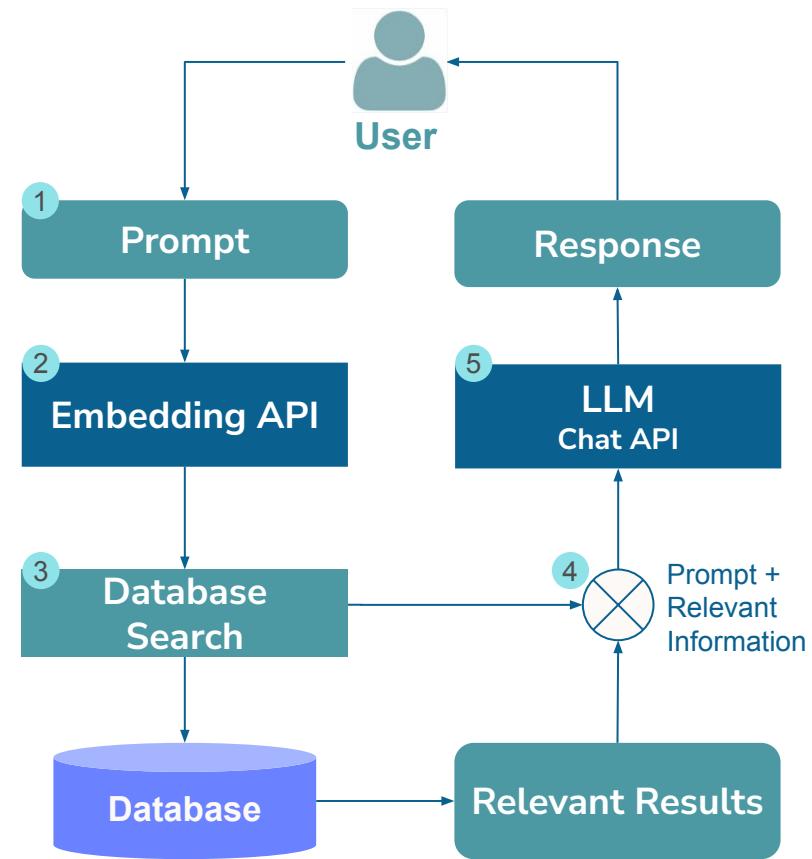
RAG augments the LLM flow by

1. intercepting a **user's prompt**,
2. making a **query to external data**,
3. passing relevant results from the query back to the LLM
4. for a **complete, curated response**.



Vector Retrieval

1. accept prompt from user
2. generate an embedding for the prompt
3. perform a vector/similarity search on the embedding
4. combine user prompt with search results
5. generate natural language results with LLM



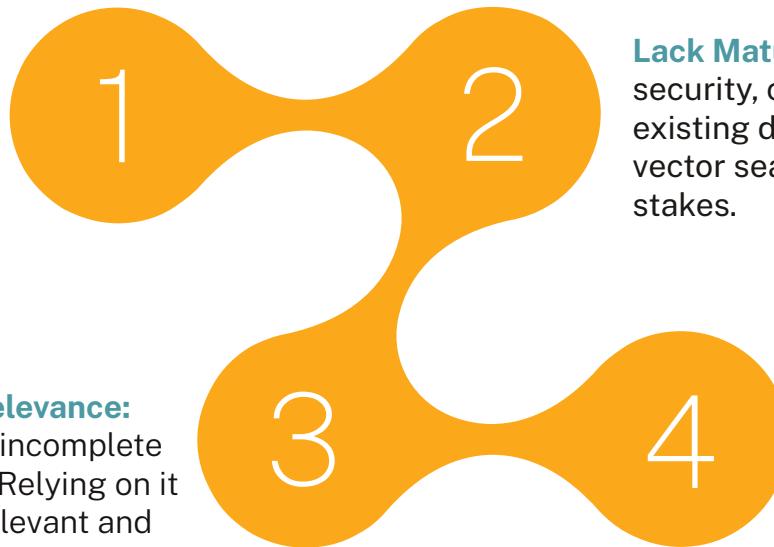


Why RAG With Vector Databases Fall Short

Similarity is insufficient for rich reasoning & explainability

Only leverage a fraction of your data: Beyond simple “metadata”, vector databases alone fail to capture relationships from structured data. Lack of structured operations - filter/aggregations/sorting.

Vector Similarity ≠ Relevance: Vector search uses an incomplete measure of similarity. Relying on it solely can result in irrelevant and duplicative results. Need to take **context** into account.



Lack Maturity: Don't apply the security, compliance, robustness of existing database systems. Also vector search became DBMS table stakes.

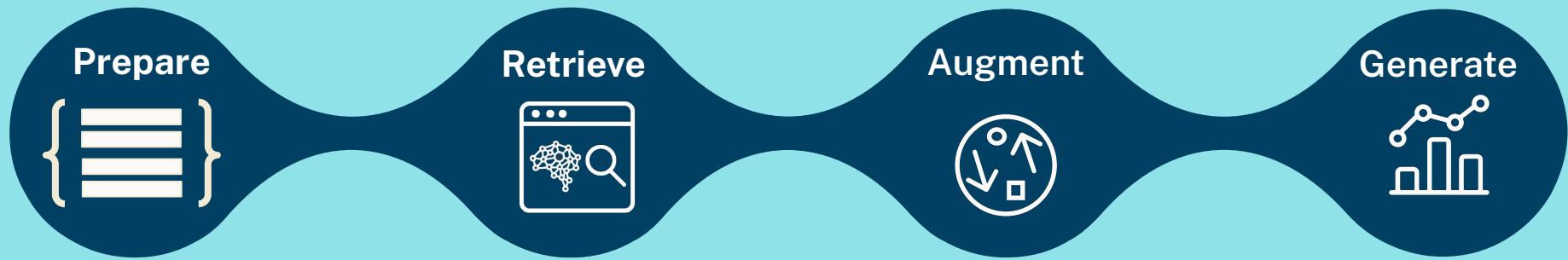
Lack Explainability: The black-box nature of vectors search lacks transparency and explainability.

The Limitations of Text Embeddings in RAG Applications

Advanced RAG Patterns & Shortcomings

RAG is a software design pattern for integrating GenAI Apps with custom data sources, like a database.

RAG Phases - Activities



Data Preprocessing
as an upfront step,
effort & cost for
better retrieval and
generation

Retrieval of **relevant**
information from
data sources via
query, fulltext, vector
and other search
paradigms

Augment



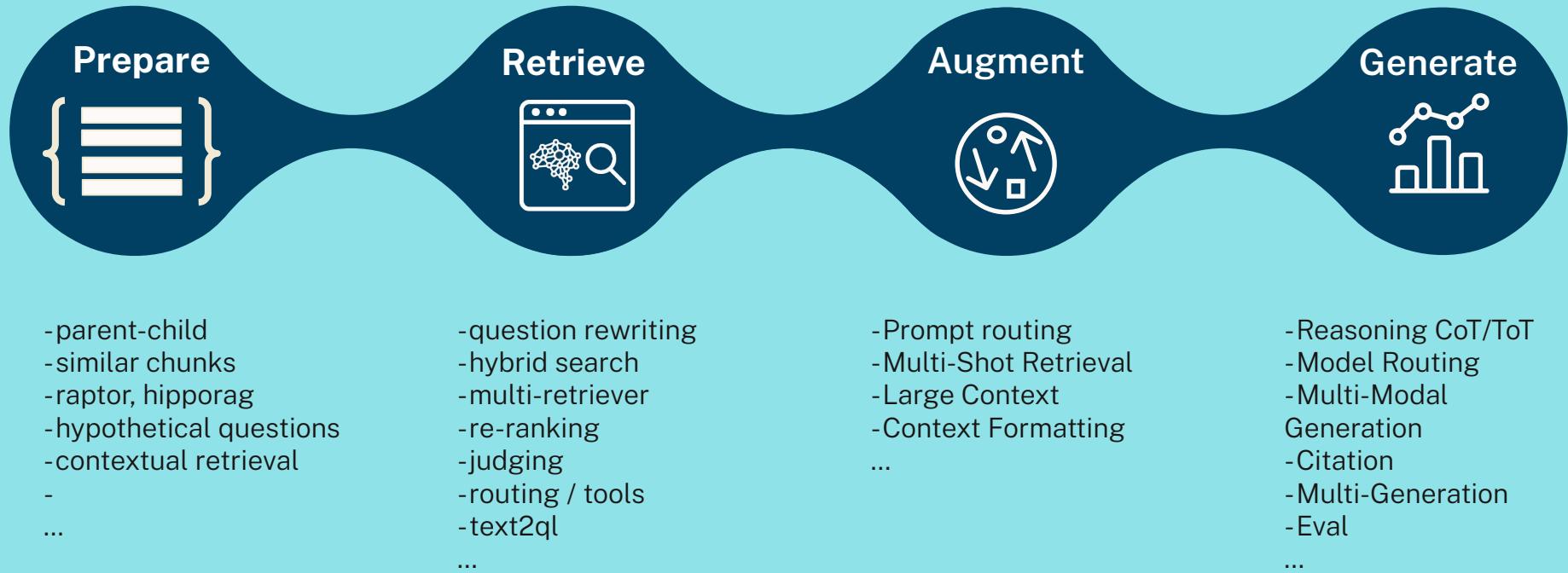
Augment Prompt
answering user question
with **retrieved contextual**
data, optionally custom
prompt, few shot examples,
strong prompt instructions
for context usage

Generate



Use (set of,
fine-tuned) LLMs to
generate end user
answer while
adhering to limits
(tokens, latency)
Here Evaluation of
RAG metrics is also
crucial and LLMOps

RAG Phases - Advanced Patterns





**Most advanced RAG patterns use connections
between pieces of information,
but recreate graph systems with id-lists and
lots of python code.**

Michael Hunger



RAG Shortcomings

- Focus on text fragments
 - no structured data / operations
- Black box operations (vector search, LLM generation)
- Pre-processing effort
- Error propagation
- Lack of Explainability
- Not making enough use of existing, compliant data-sources
- Token count / latency



A dramatic scene of drowning in the ocean. A man is partially submerged, his head above water with a look of distress. To his left, a bright orange lifebuoy floats in the dark green water. The word "GRAPH" is printed in large, bold, black capital letters on the top half of the ring, and "RAG" is printed in large, bold, black capital letters on the bottom half. The water is choppy, with white foam around the lifebuoy and the swimmer.

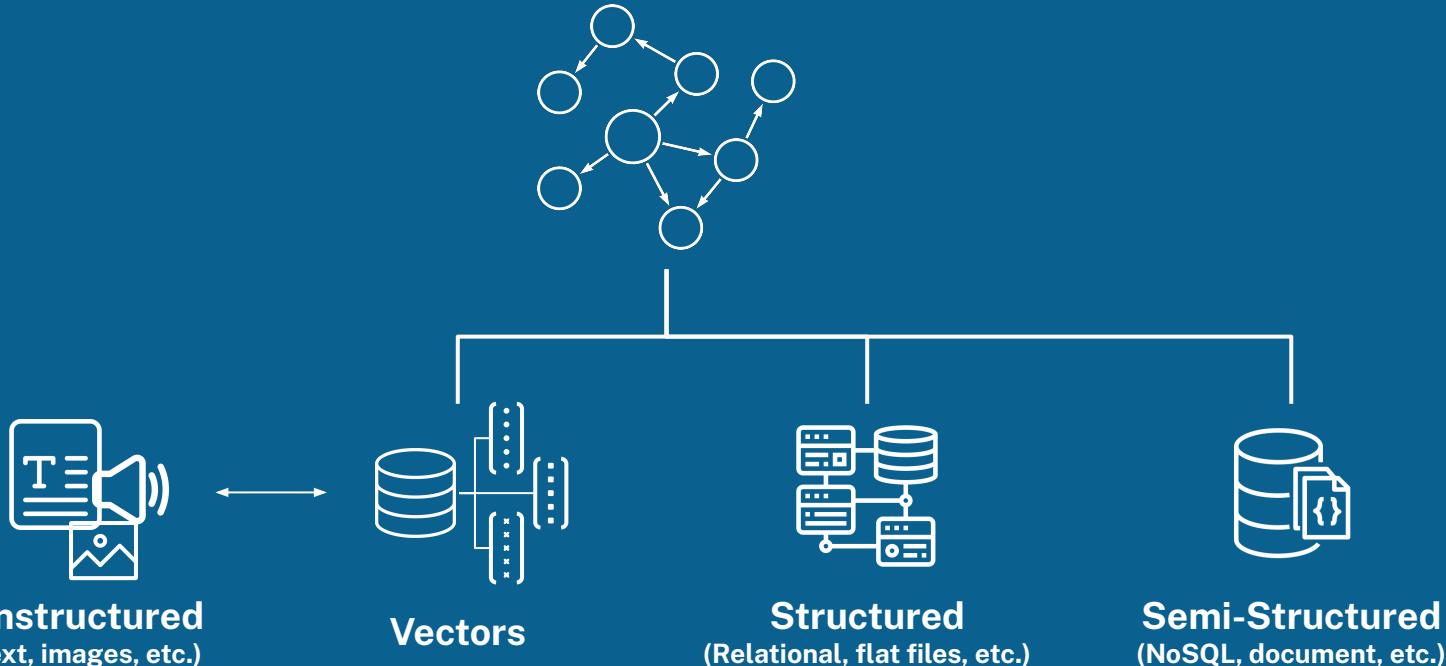
GRAPH
RAG

Structure of Data

Structured, Semi-Structured, Unstructured,
Multi-Modal

What Else to Use for External Data?

Knowledge Graph
Superset data structure for RAG





Unstructured Data

Unstructured data from
PDFs, or images



Mixed Text + Data

Structured data together
with long-form text



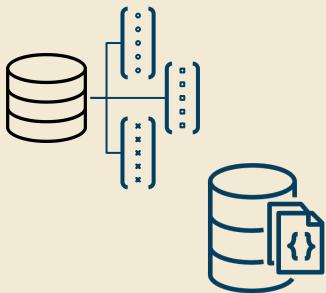
Structured Data

Data records in a
database

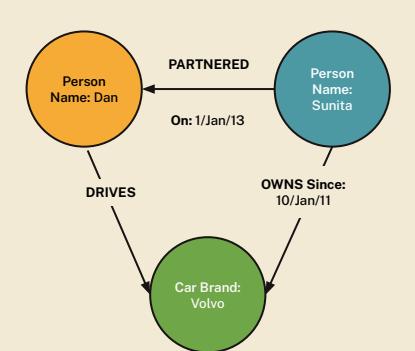
What Data Stores to use for RAG?

Best to have search indexes (i.e vector), entity modeling & structured relationships

Vector / Document DB



Graph Database



RDBMS

| PERSON_ID | NAME |
|-----------|------|
| 1 | Ann |
| 2 | Dan |

| CAR_ID | Model |
|--------|-------|
| 1 | Volvo |

| P1_ID | P2_ID | ON |
|-------|-------|----------|
| 1 | 2 | 1/Jan/13 |

| CAR_ID | DRIVER_ID |
|--------|-----------|
| 1 | 2 |

| CAR_ID | OWNER_ID | SINCE |
|--------|----------|-----------|
| 1 | 1 | 10/Jan/11 |

+7 foreign keys

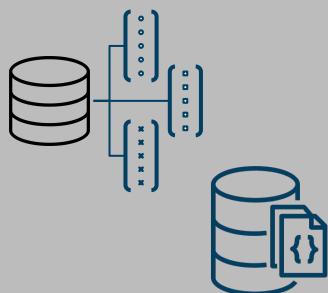
Incomplete data model - maybe good If paired with another DB

Complete data model but Inflexible, verbose, and complex

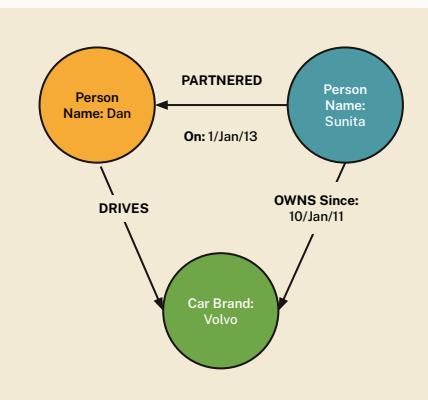
What Data Stores to use for RAG?

Best to have search indexes (i.e vector), entity modeling & structured relationships

Vector / Document DB



Graph Database



RDBMS

| PERSON_ID | NAME |
|-----------|------|
| 1 | Ann |
| 2 | Dan |

| CAR_ID | Model |
|--------|-------|
| 1 | Volvo |

| P1_ID | P2_ID | ON |
|-------|-------|---------|
| 1 | 2 | 1/Jan13 |

| CAR_ID | DRIVER_ID |
|--------|-----------|
| 1 | 2 |

| CAR_ID | OWNER_ID | SINCE |
|--------|----------|-----------|
| 1 | 1 | 10/Jan/11 |

+7 foreign keys

Simple & Compact

Fast & Agile Development

High Performance Queries

Knowledge Graphs

Knowledge Graph

**Facts about people, places, events, or things interlinked
by their relationships**

Human and LLM-friendly readable format

Organizing principle provides context for reasoning
about the data

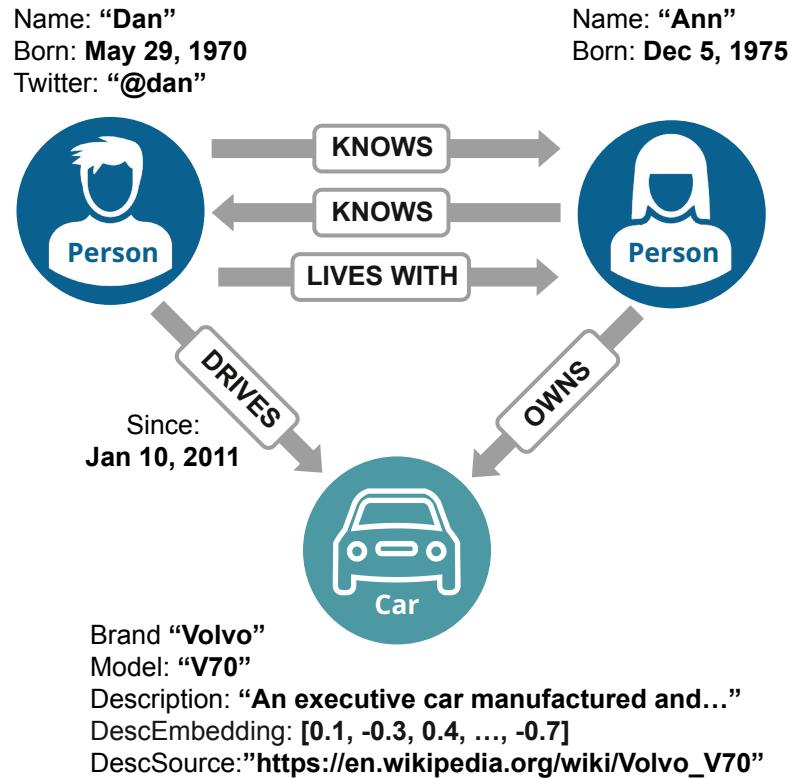
Digital Twin of your organization

Neo4j Graph Components

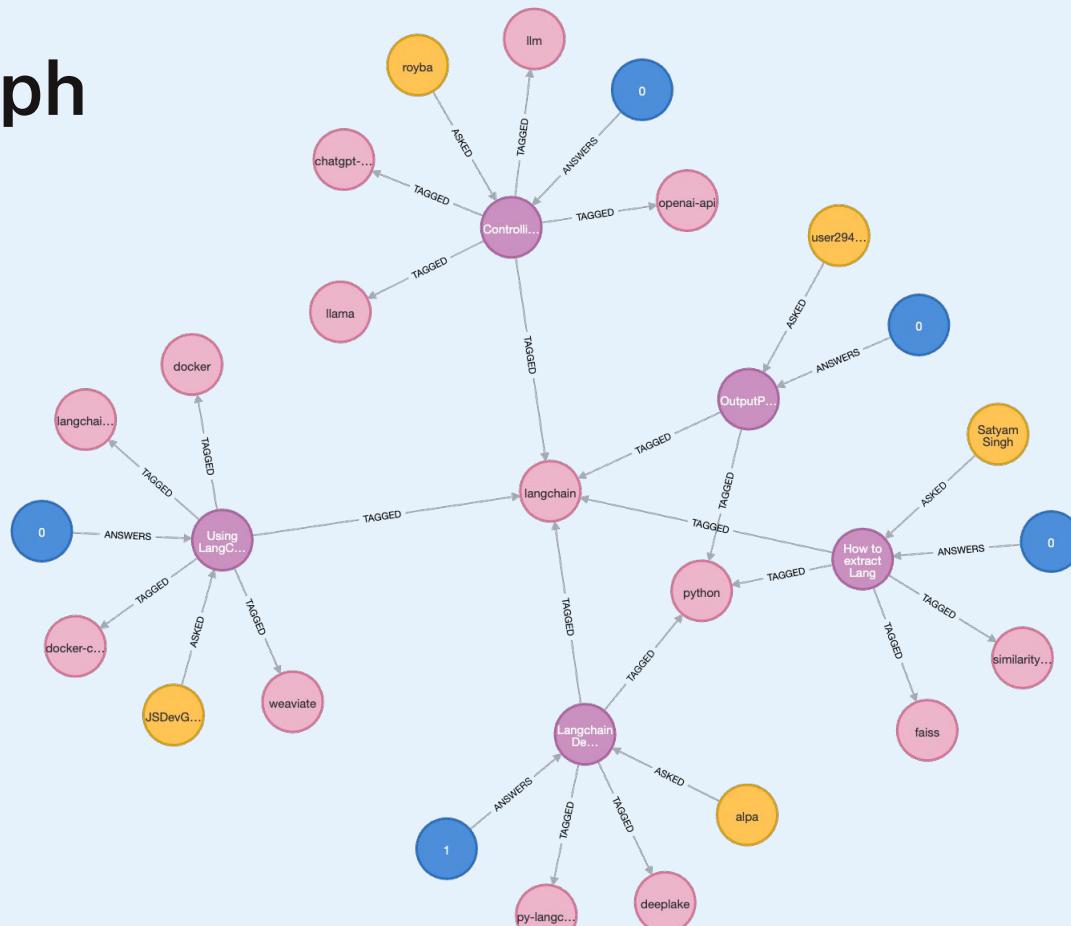
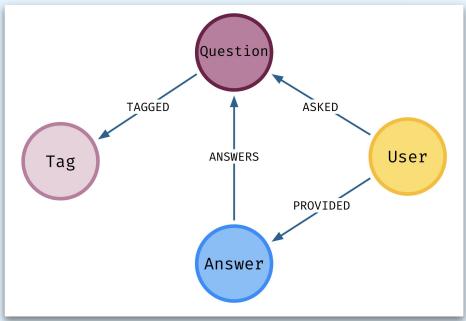
Nodes represent entities in the graph

Relationships represent associations or interactions between nodes

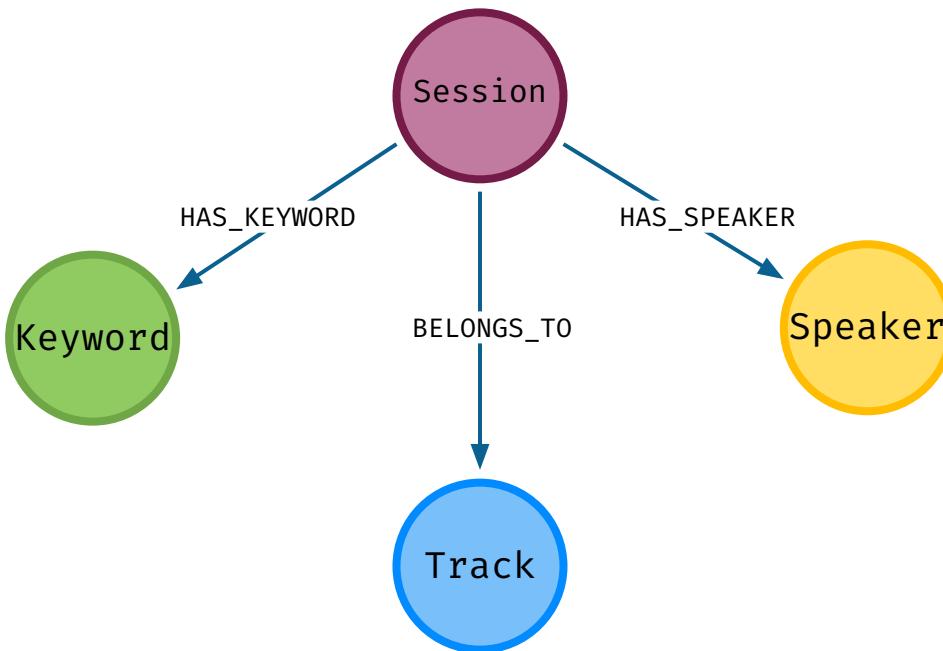
Properties represent attributes of nodes or relationships including vectors, can be *indexed*



StackOverflow Knowledge Graph



Data pattern: from tracks to tagged keywords



(Track) \leftarrow [BELONGS_TO] - (Session)
- [HAS_KEYWORD] \rightarrow (Keyword)

GraphRAG

Better Retrieval using Graph Data Structures

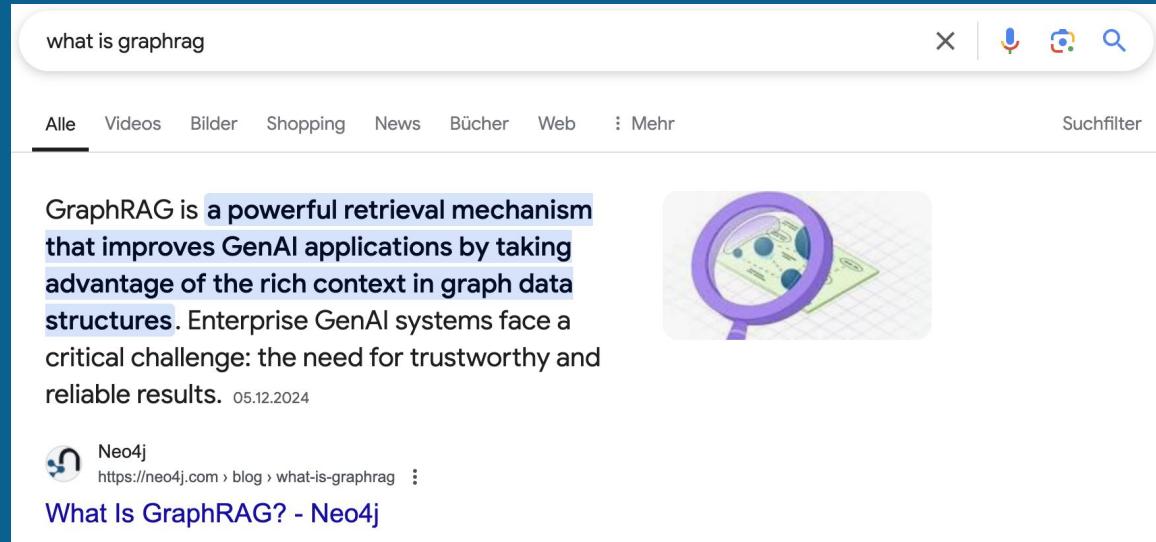
LLMs and Knowledge Graphs



GraphRAG

Advanced RAG Patterns
that benefit from using
Graph Data Structures
for Retrieval for relevant
context and higher
explainability.

Patterns - graphrag.com



what is graphrag

Alle Videos Bilder Shopping News Bücher Web : Mehr Suchfilter

GraphRAG is a powerful retrieval mechanism that improves GenAI applications by taking advantage of the rich context in graph data structures. Enterprise GenAI systems face a critical challenge: the need for trustworthy and reliable results. 05.12.2024

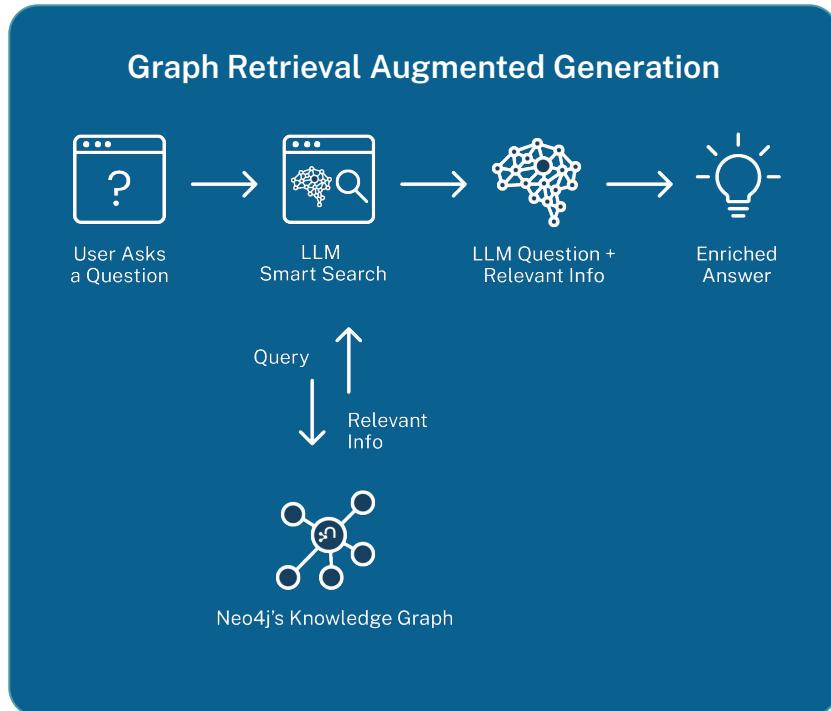
 Neo4j
<https://neo4j.com/blog/what-is-graphrag> ::

[What Is GraphRAG? - Neo4j](#)

Evolving From RAG to GraphRAG

A Neo4j Knowledge Graph combined with LLM's obtains some unique improvements:

- **Relevancy** - Obtain more relevant answers compared to just vector searches
- **Context**: includes domain specific, factual, structured knowledge on your subject
- **Explainability**: Provide the user with more reasoning on how the results were obtained.
- **Security**: Role Based Access Control



GraphRAG (QFS)

From Local to Global: A Graph RAG Approach to Query-Focused Summarization

Darren Edge^{1†} Ha Trinh^{1†} Newman Cheng² Joshua Bradley² Alex Chao³

Apurva Mody³ Steven Truitt²

Jonathan Larson¹

¹Microsoft Research

²Microsoft Strategic Missions and Technologies

³Microsoft Office of the CTO

{daedge, trinhha, newmarcheng, joshbradley, achao, moapurva, steventruitt, jolarso}
@microsoft.com

[†]These authors contributed equally to this work

Abstract

The use of retrieval-augmented generation (RAG) to retrieve relevant information from an external knowledge source enables large language models (LLMs) to answer questions over private and/or previously unseen document collections. However, RAG fails on global questions directed at an entire text corpus, such



GraphRAG is a technique for richly understanding text datasets by combining text extraction, network analysis, and LLM prompting and summarization into a single end-to-end system.

MSFT Research



Graph Retrieval-Augmented Generation: A Survey

Prior works

Derivative works

 List view

 Filters

... More

Origin paper

Graph Retrieval-Augmented Generation: A Survey

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo,... 2024

GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning

Costas Mavromatis, George Karypis 2024

Explore then Determine: A GNN-LLM Synergy Framework for Reasoning over Knowledge...

Guangyi Liu, Yongqi Zhang, Yong Li, Quanming... 2024

Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on...

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Sa... 2023

ChatKBQA: A Generate-then-Retrieve Framework for Knowledge Base Question...

Haoran Luo, E. Haihong, Zichen Tang, Shiya... 2023

Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari,... 2023

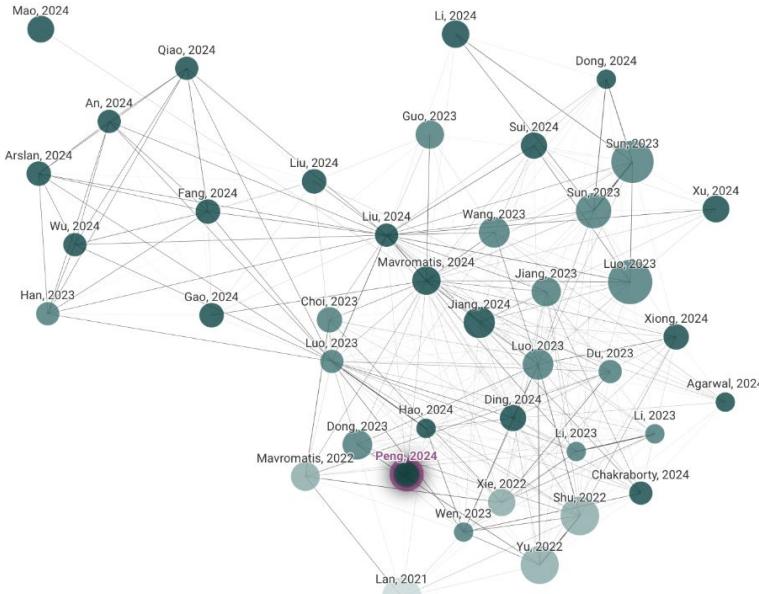
KG-Agent: An Efficient Autonomous Agent Framework for Complex Reasoning over...

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang... 2024

Think-on-Graph: Deep and Responsible Reasoning of Large Language Model with...

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Sa... 2023

Multi-hop Question Answering over Knowledge



Graph Retrieval-Augmented Generation: A Survey

Boci Peng + 6 authors Siliang Tang

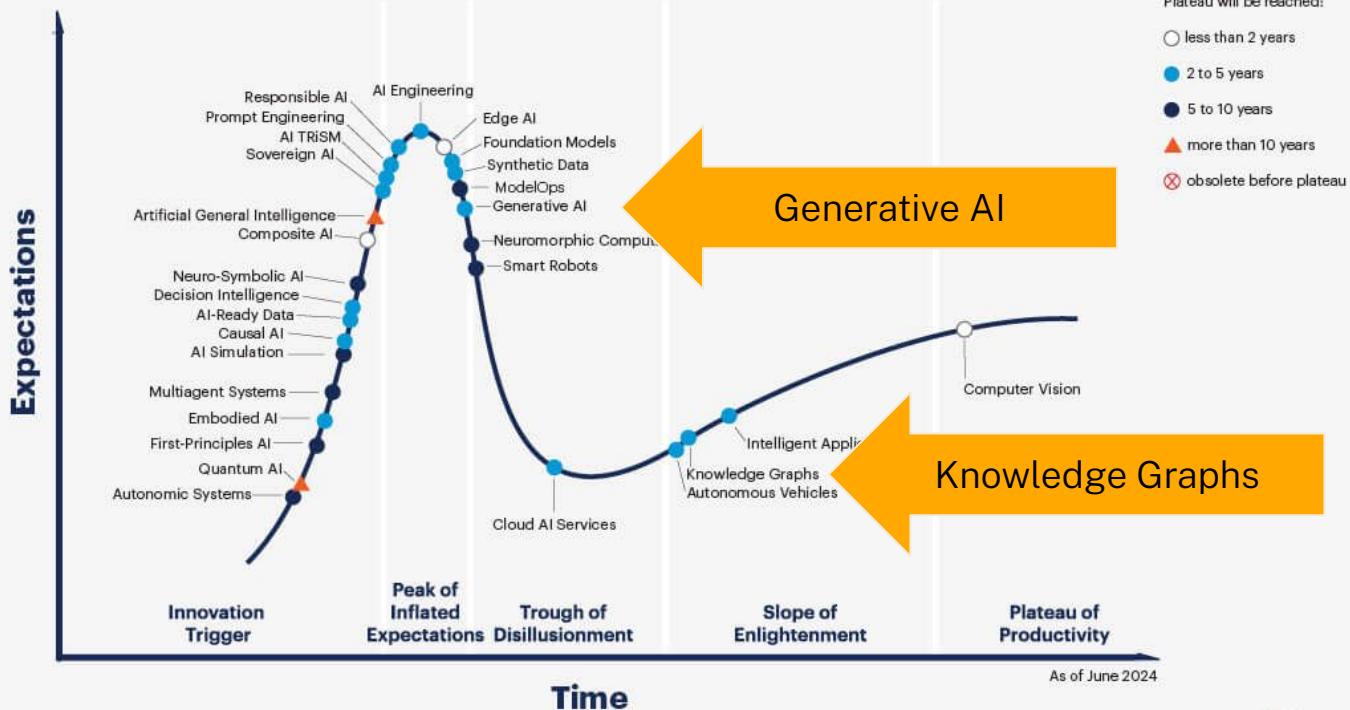
2024, arXiv.org

• • •

Open in: [PDF](#) X

Recently, Retrieval-Augmented Generation (RAG) has achieved remarkable success in addressing the challenges of Large Language Models (LLMs) without necessitating retraining. By referencing an external knowledge base, RAG refines LLM outputs, effectively mitigating issues such as "hallucination", lack of domain-specific knowledge, and outdated information. However, the complex structure of relationships among different entities in databases presents challenges for RAG systems. In response, GraphRAG leverages structural information across entities to enable more precise and comprehensive retrieval, capturing relational knowledge and facilitating more accurate, context-aware responses. Given the novelty and potential of GraphRAG, a systematic review of current technologies is imperative. This paper provides the first comprehensive overview of GraphRAG methodologies. We formalize the GraphRAG workflow, encompassing Graph-Based Indexing, Graph-Guided Retrieval, and Graph-Enhanced

Hype Cycle for Artificial Intelligence, 2024

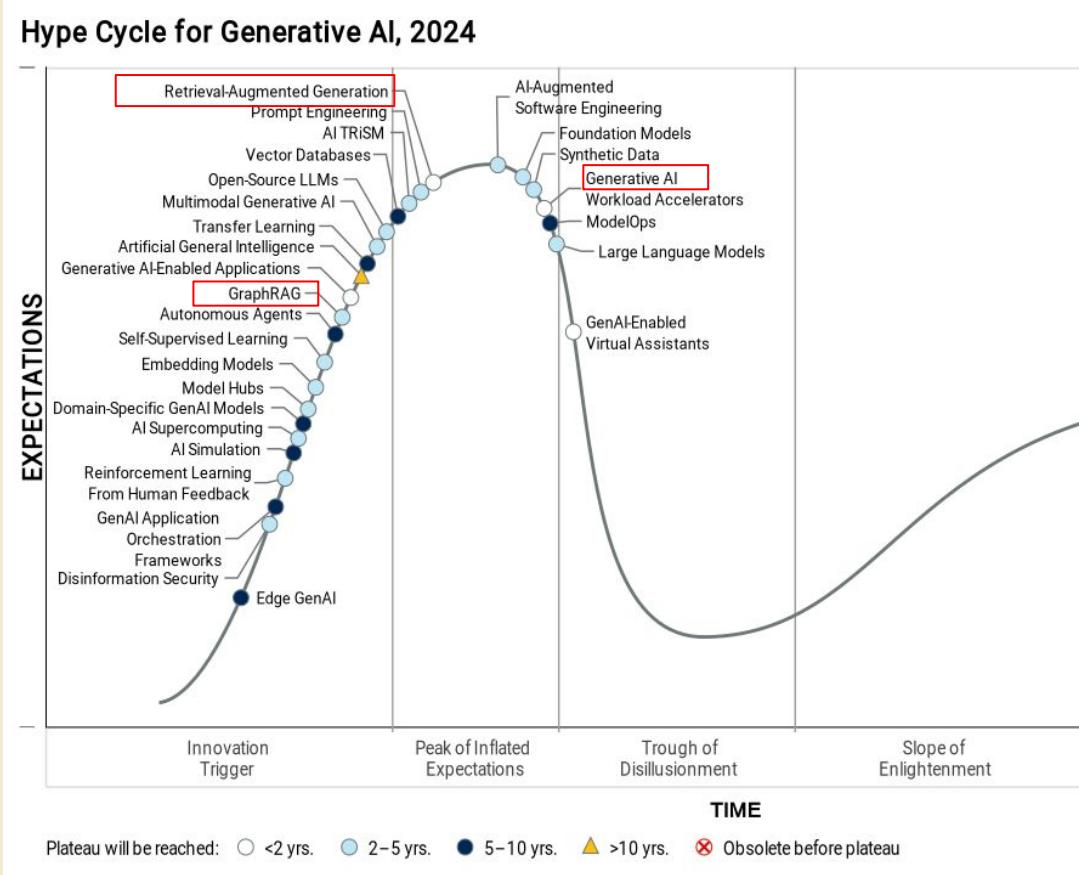


Source: Gartner
Commercial reuse requires approval from Gartner and must comply with the Gartner Content Compliance Policy on gartner.com.
© 2024 Gartner, Inc. and/or its affiliates. All rights reserved. GTS_3282450

Gartner[®]

All the Hype - GenAI -> RAG -> GraphRAG

- Fast Moving Space
- Lots of Excitement
- Inflated Expectations
- Mostly experimentation
- Lack of Standards
- Lack of Experience
- More Risk & Uncertainty



GraphRAG Phases

Chunk



Substructure, **chunk** and **embed** documents into a **lexical graph**

Extract



Extract **entities** and **relationships** with descriptions, facts and claims.

Enrich



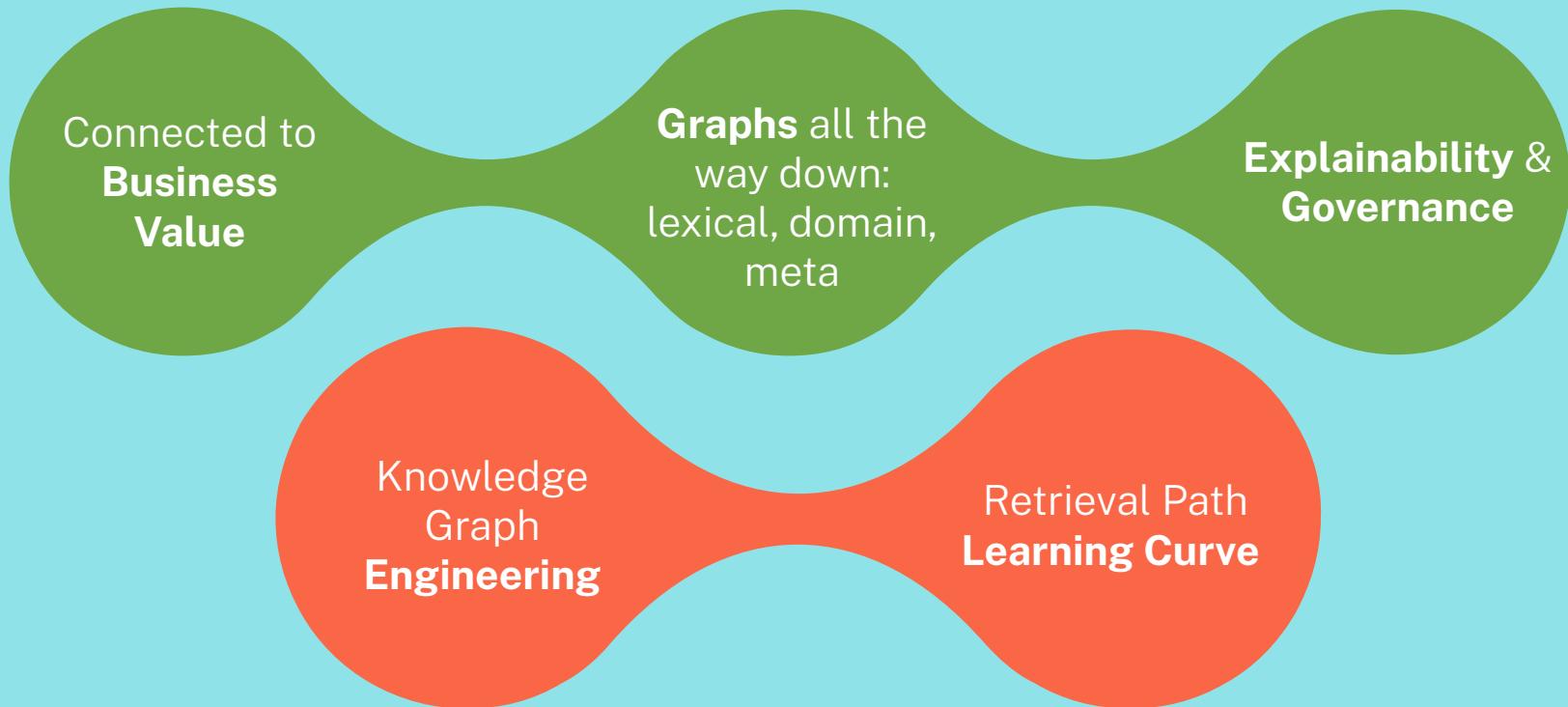
Compute clusters of topical structures, generate **community summaries**
Enrich with other graph features from algorithms.

Search



Enhanced vector search with graph context
Local search over entities and their patterns
Global search over community summaries

The Benefits & Stumbling Blocks of GraphRAG



neo4j

GraphRAG Patterns

RAG - Retrieval using Graph Structures

GraphRAG Pattern Catalog - graphrag.com

1. Survey of GraphRAG research papers
2. Identify and describe the patterns
3. Show the knowledge graph
4. Explain the retrieval method

Goal: easily understood reference guide for developers

Stretch: distill common patterns as they appear

Contributions welcome!



**Patterns are named, proven solutions for
problems in a specific context with forces,
pro's and con's and examples for when and
when not to use them.**

Christopher Alexander, Architect, Creator of A Pattern Language



GraphRAG Pattern Catalog - graphrag.com

1. Pattern name, description, aliases
2. Context, when to apply
3. Pro's and Con's
4. Required/recommended Graph Model and Retrieval Query
5. References to Papers, Examples

Example Pattern

Name: Graph Enhanced Vector Search

Description: The user question is embedded using the same embedder used to create chunk embeddings. A vector similarity search is executed on the chunk embeddings to find k (number previously configured by developer/user) most similar chunks. A traversal of the Domain Graph starting at the found chunks is executed to retrieve more context.

Context: The biggest problem with basic GraphRAG patterns is finding all relevant context necessary to answer a question. The context can be spread across many chunks not being found by the search. Relating the real-world entities from the chunks to each other and retrieving these relationships together with a vector search provides additional context about these entities that the chunks refer to. They can also be used to relate chunks to each other through the entity network.

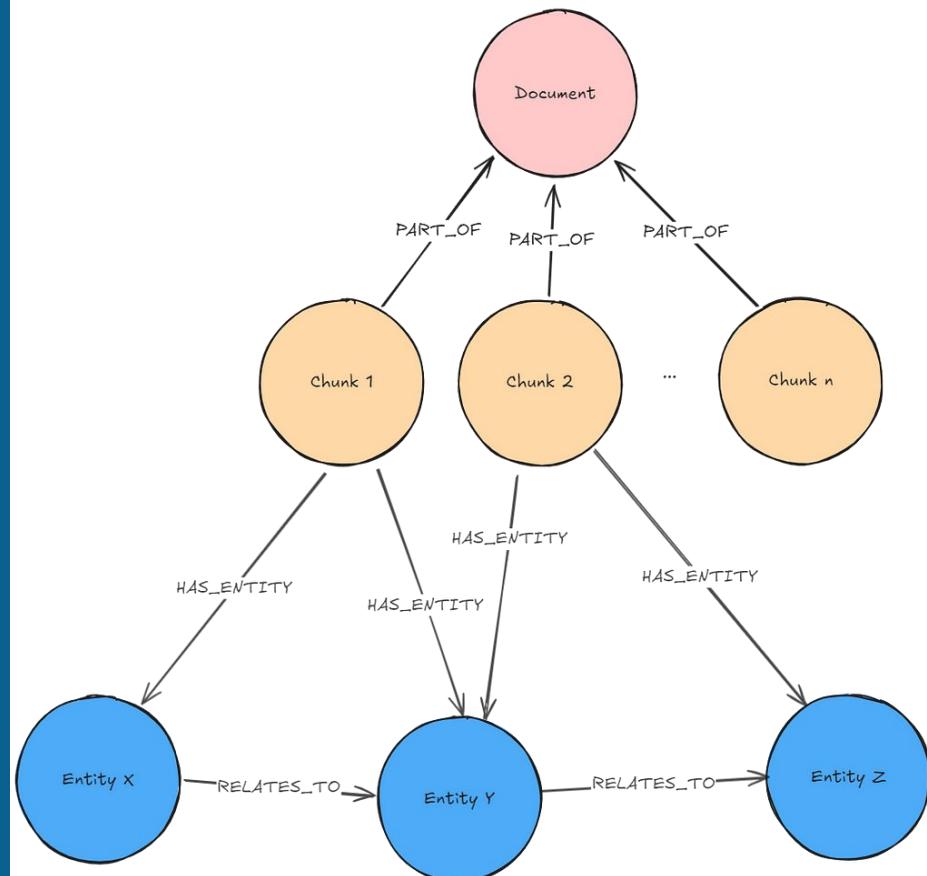
Required pre-processing: Use an LLM to execute entity and relationship extraction on the chunks. Import the retrieved triples into the graph.

Variations: Entity disambiguation, Question-guided/Schema-defined extraction, Entity embeddings, Ontology-driven traversal

```
MATCH (node)-[:PART_OF]->(d:Document)
MATCH (node)-[:HAS_ENTITY]->(e)
MATCH path=(e)((()-[rels:!HAS_ENTITY&!PART_OF]-()){0,2}(:!Chunk&!Document)
...
RETURN ...
```

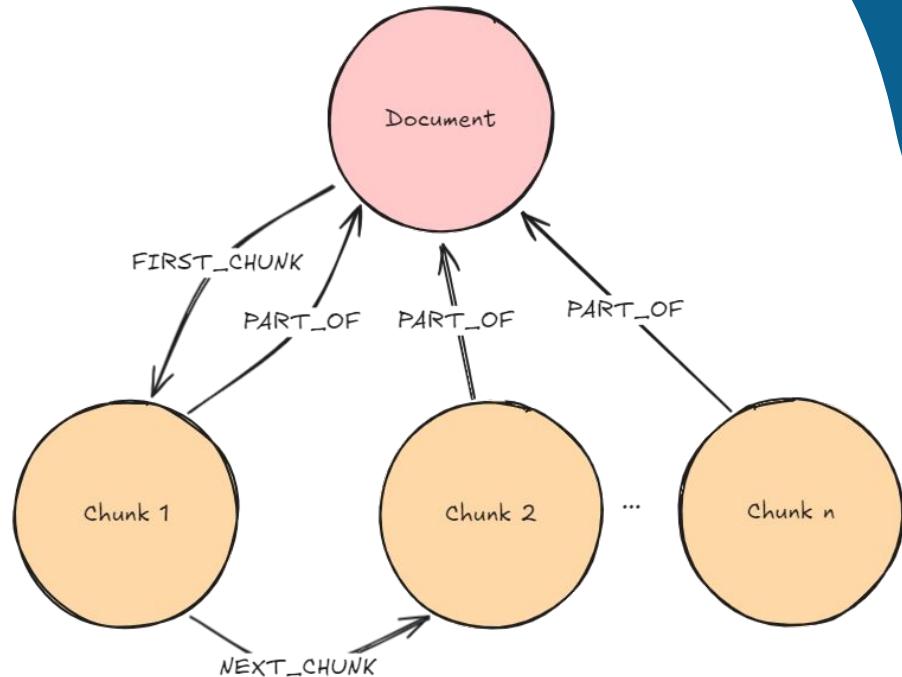
AKA: Graph + Vector, Augmented Vector Search

Knowledge graph shape: Lexical Graph with Extracted Entities

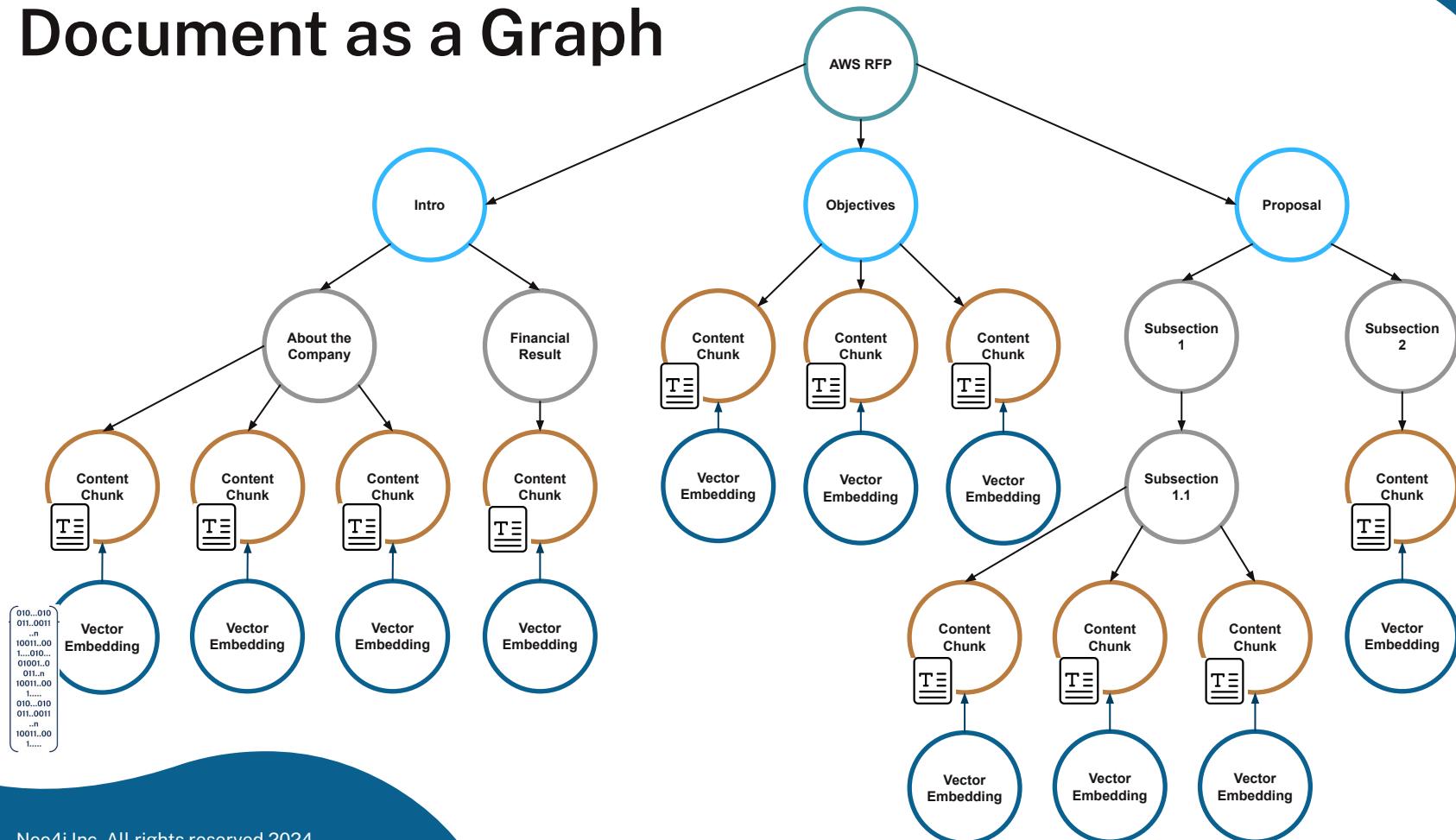


Document Preprocessing - Lexical Graph

1. Chunk Document
 - a. fixed (w/ overlap)
 - b. semantic
 - c. hierarchical
2. Link Chunks to
 - a. Documents,
 - b. Siblings,
 - c. Similar
3. Optional Rewrite Chunks
4. Generate Chunk embeddings



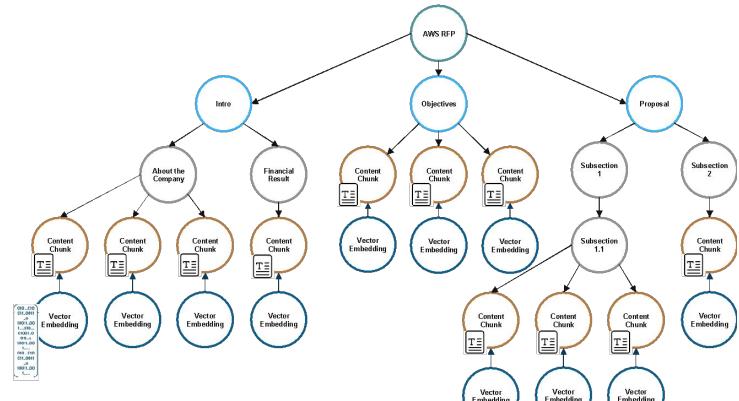
Document as a Graph



Knowledge Graph as the Knowledge Base

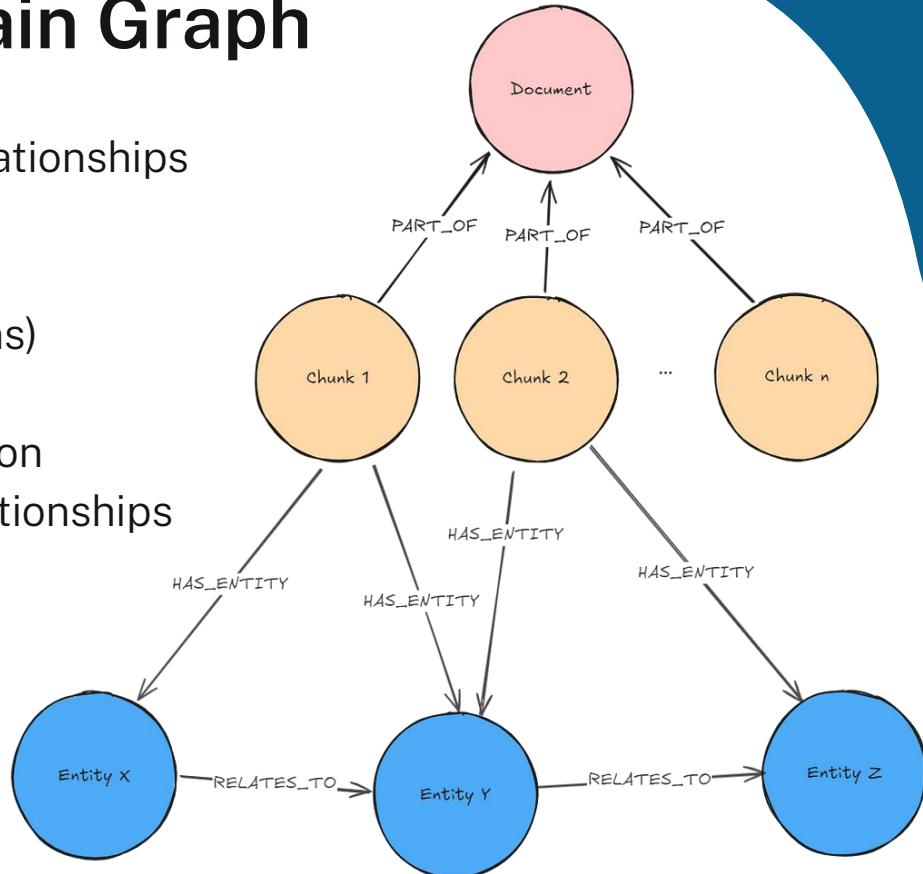
Document as a Graph ➤

Knowledge Graph



Entity Extraction -> Domain Graph

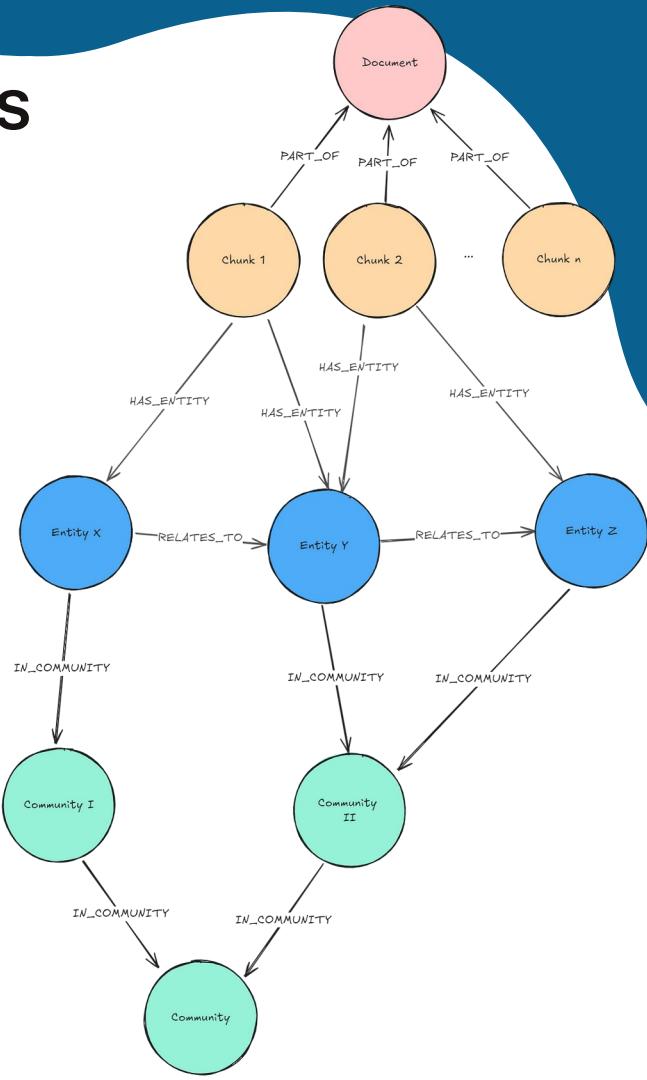
1. Coreference Resolution
2. Use LLM for extraction of Entities & Relationships
 - a. Dedicated Extraction Prompt
 - b. Additional Graph Schema
 - c. Guided Extraction (e.g. via questions)
 - d. Structured Outputs are mandatory
3. Entity Linking & Resolution-Deduplication
4. Additional attributes for Nodes and relationships
 - a. name / id
 - b. description
 - c. score / confidence
 - d. claims / facts
 - e. quantitative / qualitative
5. Generate Entity Embeddings



High Level Community Summaries

Adding Contextual Metadata

1. (Hierarchical) Clustering Algorithm on Entity Graph
 - a. e.g. Leiden or Louvain Algorithm
2. Link Entities to their communities
3. Link to parent communities
4. Generate Community Summaries
 - a. Additional features like size, timeframes etc.
5. Repeat per Level/Rank
6. Generate embeddings per Community Summary



Global Community Summary Retriever

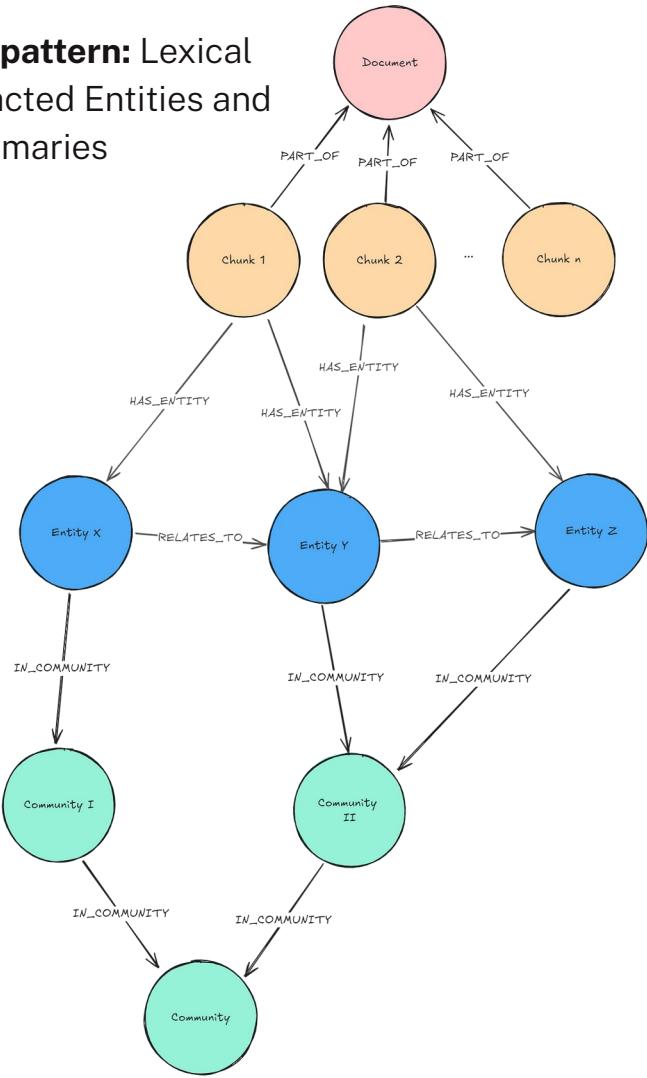
AKA: *Microsoft GraphRAG*, Global Retriever

Context: Certain **questions that can be asked on a whole dataset** do not just relate to things present in some chunks but rather search for an overall message that is overarching in the dataset.

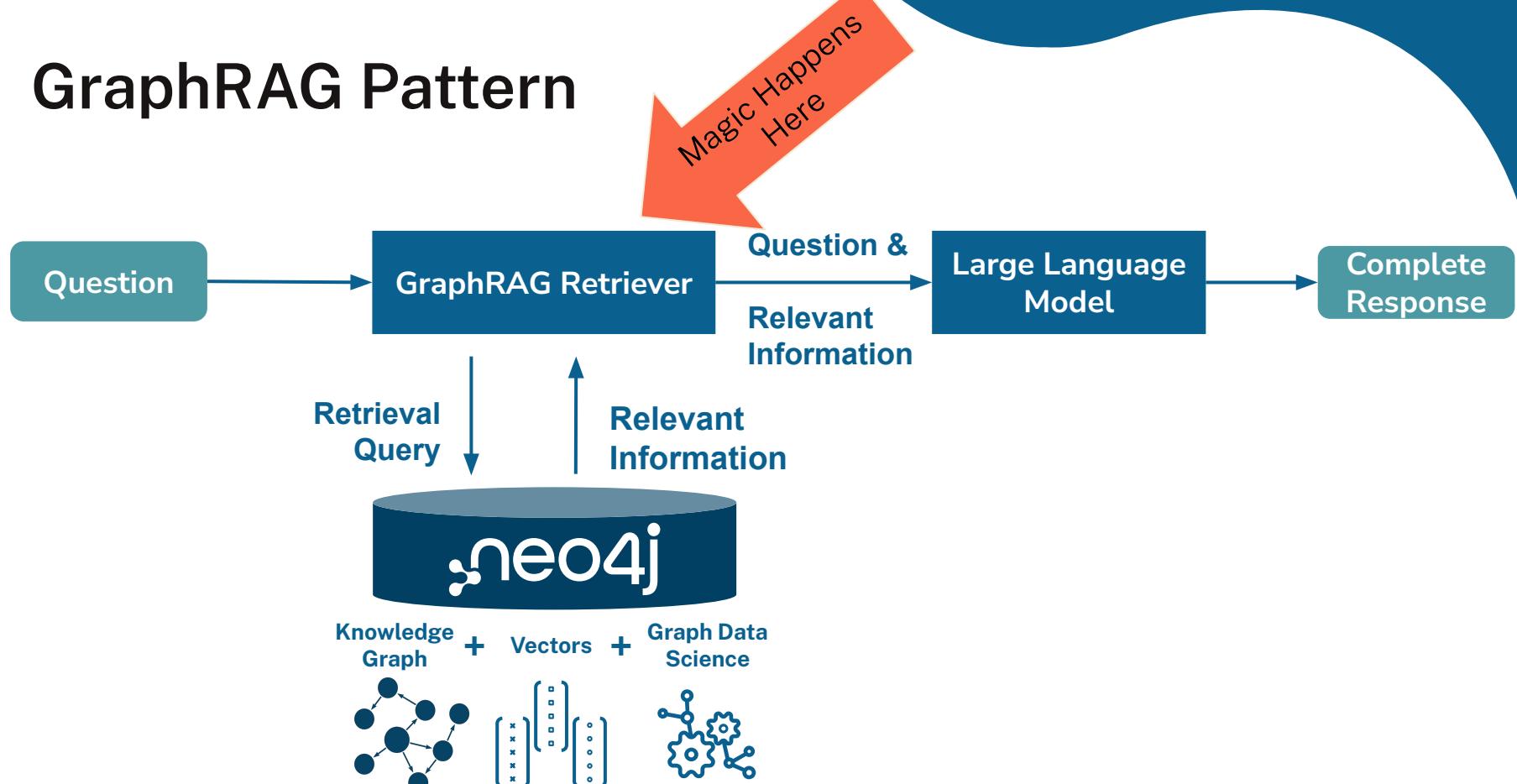
Required pre-processing: In addition to extracting entities and their relationships, we need to form hierarchical communities within the Domain Graph. For every community, an LLM summarizes the entity and relationship information into Community Summaries.

```
MATCH (c:__Community__)
WHERE c.level = $level
RETURN c.full_content AS output
```

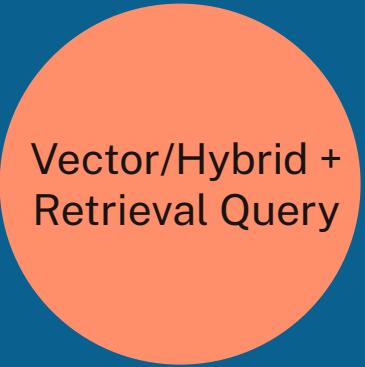
Required graph pattern: Lexical Graph with Extracted Entities and Community Summaries



GraphRAG Pattern

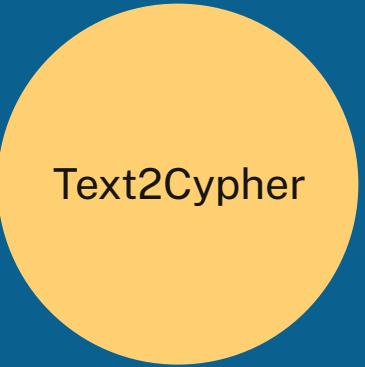


Types of GraphRAG Retrievers



Vector/Hybrid +
Retrieval Query

Generic or SME created
specific queries,
starting at entry points
and retrieving text +
context



Text2Cypher

FineTuned LLM
w/ schema
generates query
for question.



Agentic
Traversal

Iterative navigation
and information
collection across the
graph until question
is answered, set of
appropriate tools

Text2Query - Text2Cypher

- question + (enhanced) schema + few shot -> query
- existing models surprisingly good at text2cypher
- struggle mostly with relationship-direction and complex queries
- CoPilots on neo4j tools
- Neo4j text2cypher project
 - open source training dataset
 - evaluations
 - fine-tuned models Llama3, Gemma, Gemini, OpenAI
- text2sql is a large research area (Spider, numberstation, ...)

Vector Search



- Find nodes using an implicit similarity search in the **vector index*** and **enrich** with additional explicit **relationships** from the **knowledge graph**
- **Hybrid Search** with fulltext search
- Create vectors of **network information using node/graph embeddings**

Now a **top 10 vector database** on LangChain.

Neo4j Indexing and Search

Neo4j makes search efficient through a variety of indexes including:

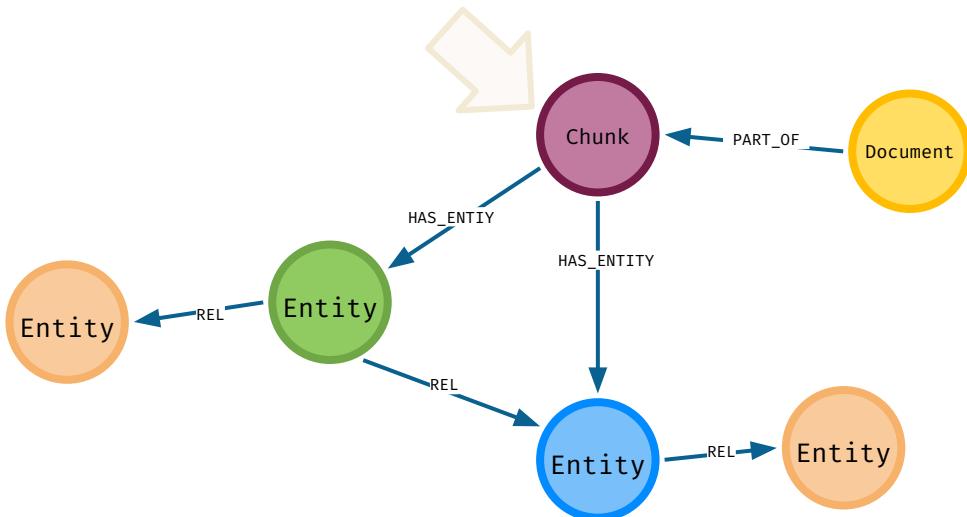
- **Range:** General index for predicates based on equality and range. Numeric, dates, etc.
- **Point:** Predicates on geospatial points like distance bounding boxes, etc.
- **Text:** Predicates on strings like contains, ends with, etc.
- **FullText:** Text search based on tokenization and analyzers
- **Vector:** ANN (Approximate Nearest Neighbor) search on vectors

Graph Enhanced Vector Search

1. Find top-k relevant chunks with (hybrid) search
2. Get containing Document(s)
3. Optionally sibling/similar chunks
4. Expand into relevant entities
 - a. optionally take entity relevancy into account for expansion
5. Context:
 - a. chunk texts
 - b. distinct entity details
 - c. distinct list of triples entity-rel->entity

Enhances basic vector search with context

- lexical context
- entity context
- path length determines context size



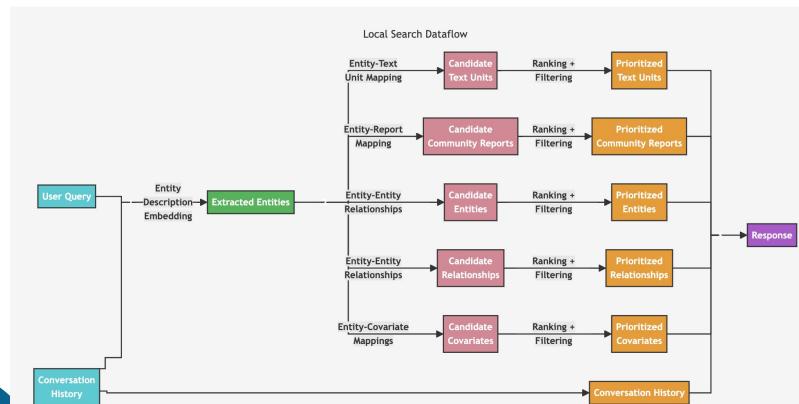
Local/Entity Search - A variant of Graph Search

1. Find top-k relevant entities with (hybrid) search
2. Expand relationships in search results
3. Ranked n-hop neighbours
4. Include relevant
 - a. Communities
 - b. Chunks
 - c. Claims

Best for specific questions on entities and facts.

Additional context from related entities

Supporting facts / information from chunks / summaries / claims.



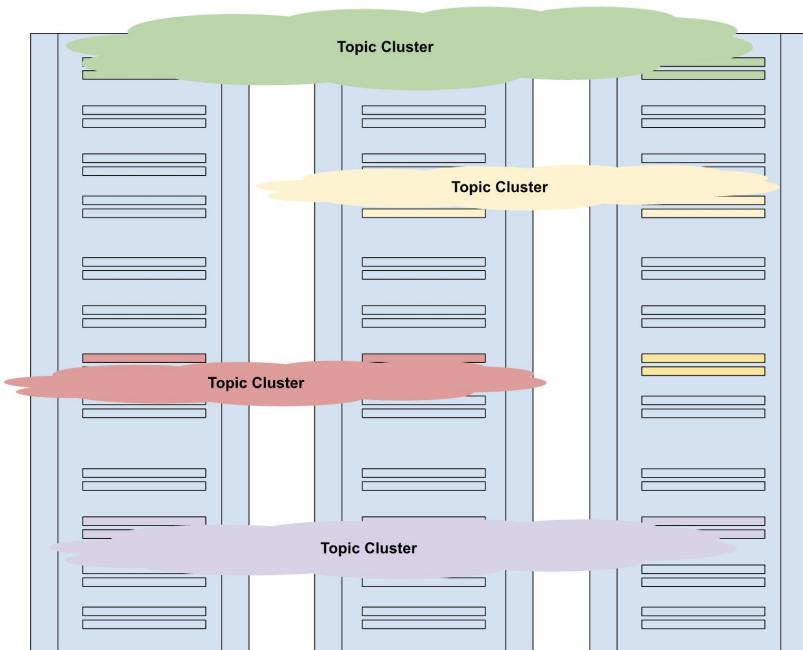
Global Search - Finding High Level Answers

For questions that cover the entirety of a Corpus.

Cross document topic clusters.

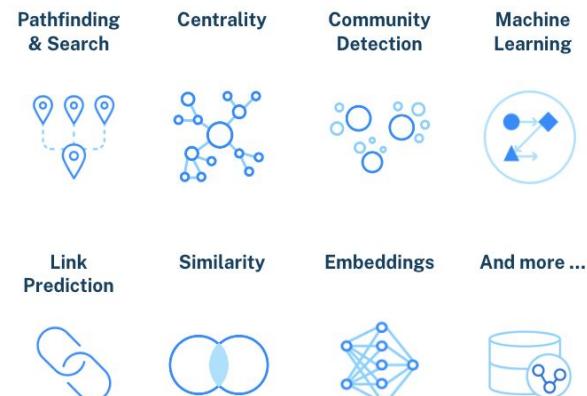
Can use all or most relevant community summaries (vector or hybrid search)

Choose level of granularity.

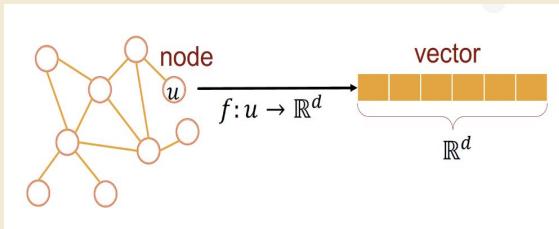


Data Science on Graphs: Graph Data Science...

The Largest Catalog of Graph Algorithms



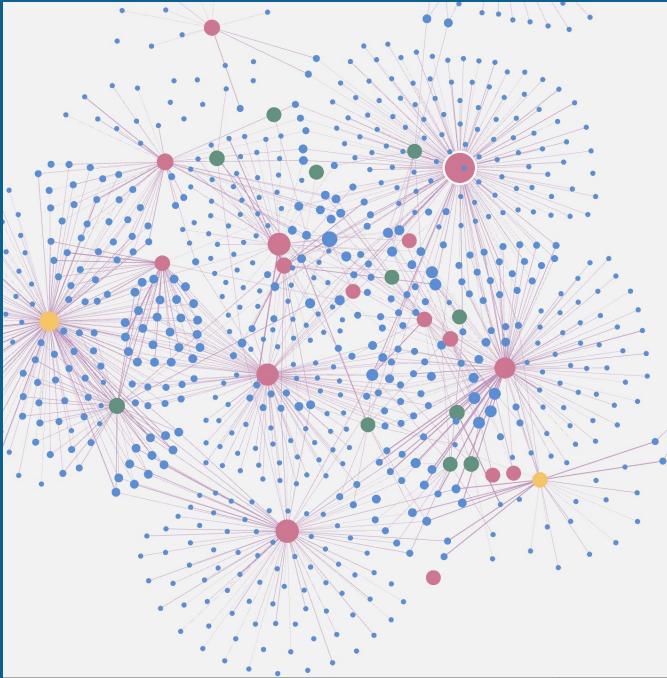
Graph Vector Embeddings for Machine Learning



Graph
Data Science

Bring the context of your connected data into
a format that other pipelines can ingest.

Enrich the KG using Graph Algorithms

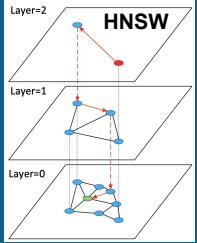


Enrich the measure of **relevancy** using graph algorithms.

- **Page Rank** to understand the importance of parts of documents
- **Link Prediction** to find hidden relationships that further contextualise the results
- **Community Detection** to group related parts of documents for more focused knowledge retrieval

Semantic Search Journey

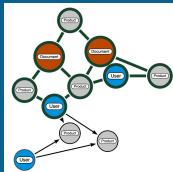
Vector Similarity Search



Find relevant documents and content for user queries.

Vector Search

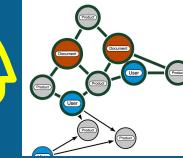
Vector Similarity + Local Graph Traversals



Find people, places, and things associated to content. Identify patterns in connected data.

Graph Database

Vector Similarity + GDS-Based Graph Traversals



Further improve search relevance using graph algorithms and ML to discover new relationships, entities, and groups.

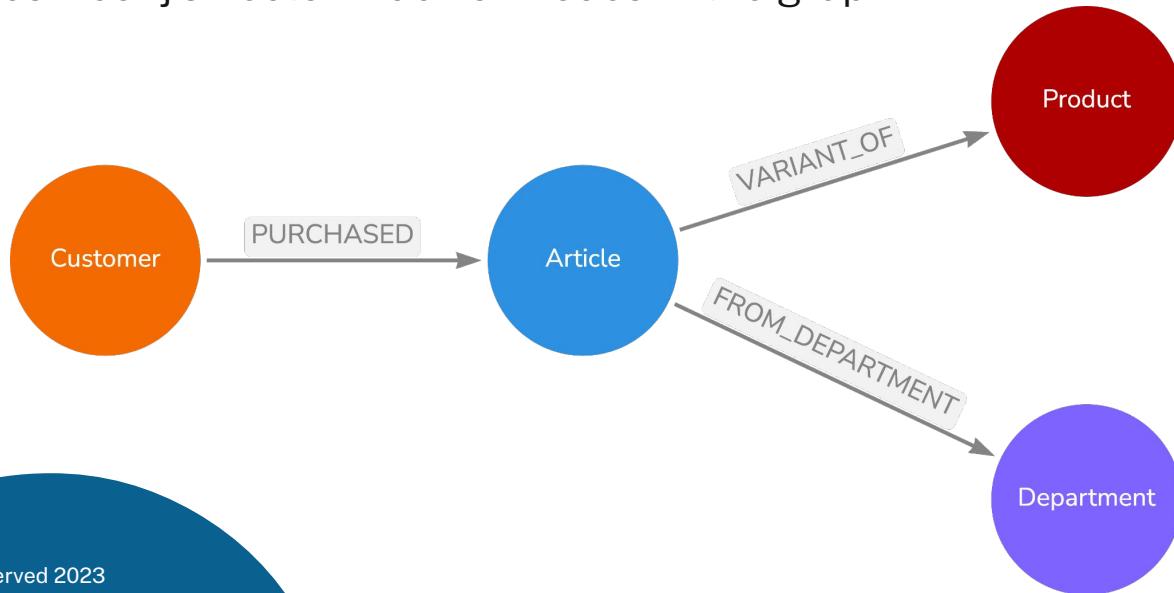
Graph Data Science

neo4j

Advanced GraphRAG Example

Example Overview

- Real-world data from the Kaggle H&M Personalized Fashion Recommendations Dataset
- Combines structured and unstructured data about articles of clothing and customer purchases
- Leverages Neo4j's Vector Index on nodes in the graph

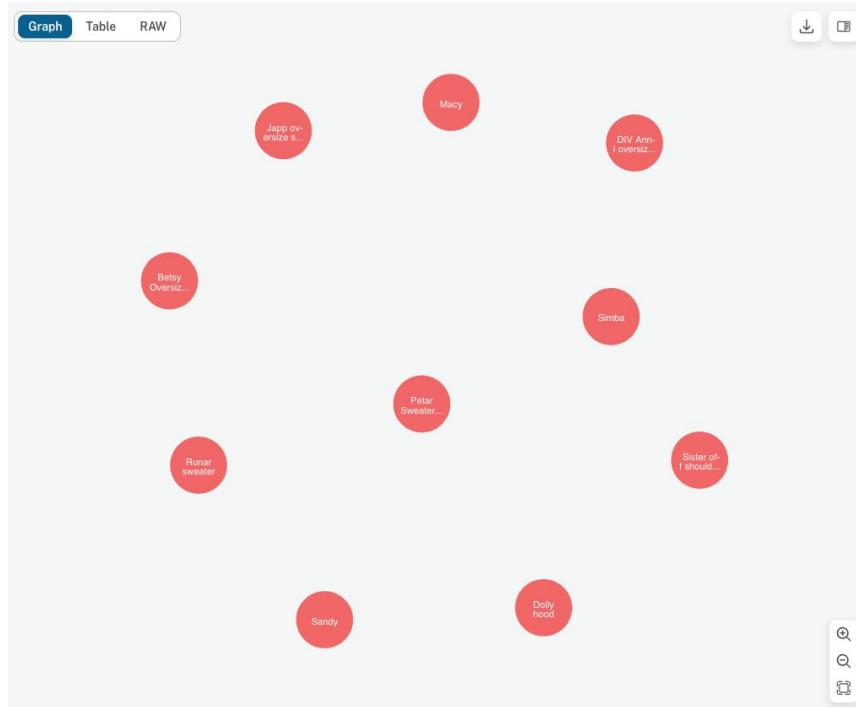


Vector Similarity Search Only

Starts with an indexed vector embedding on each node

Uses Neo4j as if it was a vector database:

- Natural language search phrase
- Performs a vector similarity search (i.e., cosine similarity)
- Return the top N results
- Each returned node is an individual chunk of data



Visualization of results from vector search-only approach

Results: Vector Search Only

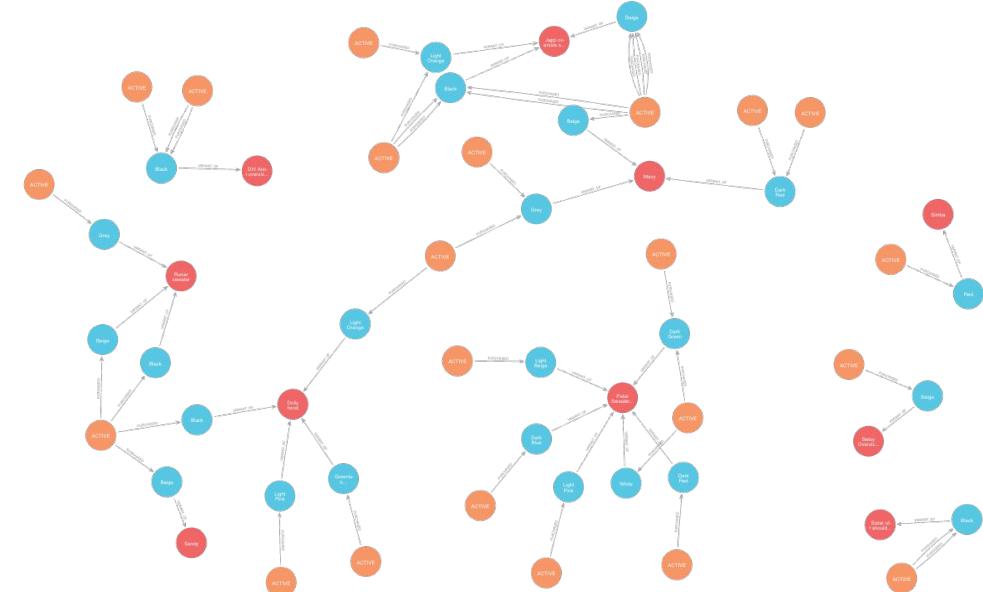
| Product Code | Product Type | Document | Similarity Score |
|--------------|--------------|--|------------------|
| 842001 | Sweater | Product-- Name: Betsy Oversized Type: Sweater... | 0.945937 |
| 817392 | Sweater | Product-- Name: Japp oversize sweater Type: Sweater... | 0.944246 |
| 709418 | Sweater | Product-- Name: DIV Anni oversize hood Type: Sweater... | 0.932580 |
| 860833 | Sweater | Product-- Name: Runar sweater Type: Sweater... | 0.931568 |
| 893141 | Sweater | Product-- Name: Sandy Type: Sweater... | 0.930025 |
| 812167 | Sweater | Product-- Name: Macy Type: Sweater... | 0.929781 |
| 690623 | Sweater | Product-- Name: Simba Type: Sweater... | 0.928991 |
| 557247 | Sweater | Product-- Name: Petar Sweater(1) Type: Sweater... | 0.928751 |
| 538283 | Sweater | Product-- Name: TOR Sweater Type: Sweater... | 0.927127 |
| 687934 | Sweater | Product-- Name: Sister off shoulder Type: Sweater... | 0.927100 |

Search Term: “oversized sweater”

Vector Similarity + Local Graph Traversal

Augments vector similarity search
with information already encoded in
the knowledge graph

- Start with a vector similarity search
 - Performs a **local graph traversal** on each matching node
 - Return the additional context *that would not be available from vector similarity search alone*



Visualization of results from vector similarity + local traversal approach

Results: Vector Similarity + Local Traversal

| Product Code | Product Type | Document | Search Score | Purchase Score | Vector-Only Rank |
|---------------|----------------|---|-----------------|----------------|------------------|
| 677930 | Sweater | Product-- Name: Queen Sweater Type: Sweater... | 0.922999 | 6 | NaN |
| 516712 | Top | Product-- Name: Jess oversize LS Type: Top... | 0.922911 | 5 | NaN |
| 557247 | Sweater | Product-- Name: Petar Sweater(1) Type: Sweater... | 0.928751 | 4 | 7.0 |
| 675408 | Sweater | Product-- Name: Mother Type: Sweater... | 0.920846 | 4 | NaN |
| 669682 | Sweater | Product-- Name: Irma sweater Type: Sweater... | 0.921362 | 2 | NaN |
| 640755 | Sweater | Product-- Name: Allen Sweater Type: Sweater... | 0.926152 | 1 | NaN |
| 687948 | Hoodie | Product-- Name: Annie Oversized Hood Type: Hoodie... | 0.925855 | 1 | NaN |
| 709991 | Sweater | Product-- Name: SISTER OL Type: Sweater... | 0.924914 | 1 | NaN |
| 687856 | Jacket | Product-- Name: Jacket Oversize Type: Jacket... | 0.924428 | 1 | NaN |
| 674826 | Sweater | Product-- Name: Fine knit Type: Sweater... | 0.921296 | 1 | NaN |

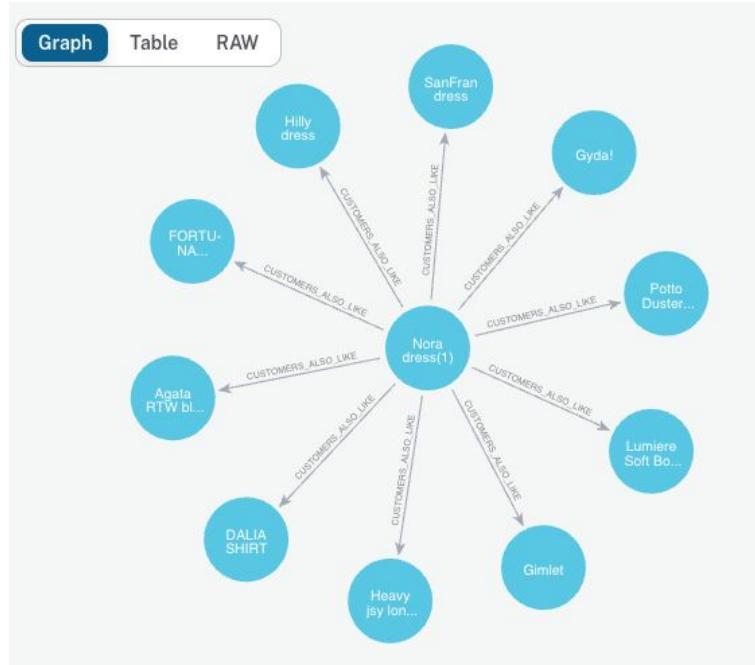
Only repeat result

Search Term: “oversized sweater”

Vector Similarity + GDS-Based Graph Traversal

Use Graph Data Science (GDS) library algorithms to encode knowledge **from the entire graph** into new relationships

- Capture product co-purchase behavior *from all customers* into FastRP node embeddings
- Apply K-Nearest Neighbors to encode this information into new relationships
- Run an updated graph traversal using new relationships as part of the RAG workflow



Visualization of results from vector similarity + GDS-based traversal approach

Results: Vector Similarity + GDS Traversal

| Product Code | Product Type | Document | Recommender Score | Vector-Only Rank |
|---------------|----------------|---|-------------------|------------------|
| 562252 | Trousers | Product-- Name: Space 5 pkt tregging Type: Trousers... | 5.50 | NaN |
| 658030 | Trousers | Product-- Name: Push Up Jegging L.W Type: Trousers... | 3.68 | NaN |
| 607347 | T-shirt | Product-- Name: Beck L/S Type: T-shirt... | 3.68 | NaN |
| 863561 | Bra | Product-- Name: Alexis seamless top Rio Opt1 Type: Bra... | 2.78 | NaN |
| 647684 | T-shirt | Product-- Name: GABBE Type: T-shirt... | 1.89 | NaN |
| 860833 | Sweater | Product-- Name: Runar sweater Type: Sweater... | 1.86 | 4 |
| 657159 | Flat shoe | Product-- Name: OL ALFONS PQ Espadrille Type: Flat shoe... | 1.86 | NaN |
| 867240 | Cardigan | Product-- Name: OKLAHOMA OVERSHIRT Type: Cardigan... | 1.86 | NaN |
| 661417 | Vest top | Product-- Name: BAE top with inner bra Type: Vest top... | 1.85 | NaN |
| 674606 | Skirt | Product-- Name: CHARLIE SKIRT Type: Skirt... | 1.85 | NaN |

Only repeat result

Search Term: “oversized sweater”

Summary

- Vector similarity search alone does not provide complete information
 - Vector similarity returns information explicitly about the search term
 - i.e., Search for “sweater” and receive 10 sweaters in return
 - Might be accurate, but ***is not complete***
- Graphs provide greater context than would not be available using vector search alone, resulting in ***accurate and complete*** information
 - Relationships explicitly define how items or concepts are related
 - Graph Data Science algorithms enable you to encode knowledge ***from the entire graph*** into easily navigable and understandable relationships

Evaluation

LinkedIn Customer Support

Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering

Zhentao Xu
zhxu@linkedin.com
LinkedIn Corporation
Sunnyvale, CA, USA

Tie Wang
tiewang@linkedin.com
LinkedIn Corporation
Sunnyvale, CA, USA

Mark Jerome Cruz
marcruz@linkedin.com
LinkedIn Corporation
Sunnyvale, CA, USA

Manasi Deshpande
madeshpande@linkedin.com
LinkedIn Corporation
Sunnyvale, CA, USA

Matthew Guevara
mguevara@linkedin.com
LinkedIn Corporation
Sunnyvale, CA, USA

Xiaofeng Wang
xiaofwang@linkedin.com
LinkedIn Corporation
Sunnyvale, CA, USA

Zheng Li
zeli@linkedin.com
LinkedIn Corporation
Sunnyvale, CA, USA

KEYWORDS

Large Language Model, Knowledge Graph, Question Answering,
Retrieval-Augmented Generation

ABSTRACT

In customer service technical support, swiftly and accurately retrieving relevant past issues is critical for efficiently resolving cus-

in BLEU. Our method has been deployed within LinkedIn's customer service team for approximately six months and has reduced the median per-issue resolution time by 28.6%.

GraphRAG Higher Accuracy (data.world)

3X

A Knowledge Graph improves the accuracy of LLM responses by 54.2%, an average of 3x better than SQL alone

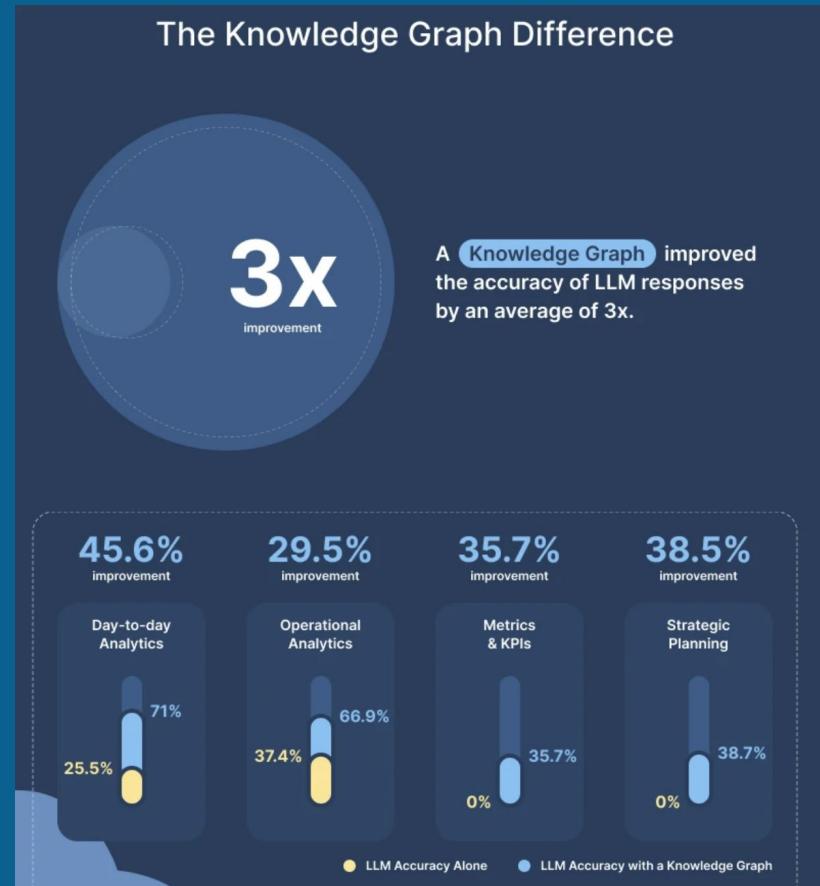
More Validating Research @ graphrag.com/appendices/research



data.world

Knowledge Graphs remove a critical barrier standing in the way of enterprises unlocking new capabilities with AI. This benchmark underscores the significant impact a Knowledge Graph can have on LLM accuracy in enterprise settings.

The implications are enormous for businesses: Making LLMs a viable means for making data-driven decision-making accessible to more people, enabling faster time-to-value with data and analytics, and surfacing new ways to use data to drive ROI, just to name a few.



Neo4j GraphRAG Eval Project

Analysis of RAG Evaluation

TLDR

- The goal of the RAG-Eval project is to implement the main steps of RAG (Retrieval Augmented Generation) and its evaluation, presented in Figure 1.
- We used the RAGAS framework for evaluation, since it offers core metrics with little to no human intervention.
- For the evaluation:
 - 3 different retrieval approaches are implemented: (i) Vector Similarity-based, (ii) Augmented Vector search (iii) Text2Cypher-based
 - 2 new Q/A datasets are created: (i) Movies (ii) Products
- The results reveal that:
 - The graph-based strategies outperform the vector search strategy.
 - Augmented vector search performance highly depends on the input cypher retrieval query.
 - Text2Cypher is relatively the most consistent retriever for question variations.
 - Text2Cypher performed better than other retrieval strategies on all complexity levels.

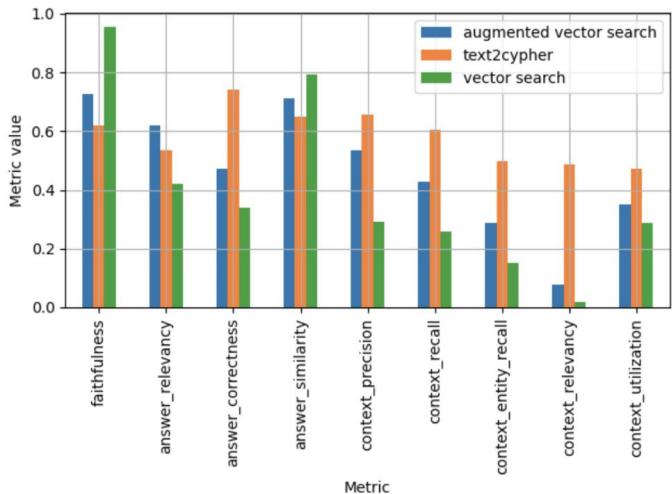


Figure 6. Average RAGAS metric values per RAG strategy on the products dataset.

GraphRAG Impact

Query Knowledge Graph within your GenAI application

Rich Context

graph entities
graph algorithms for
enhancement &
clustering



:neo4j

Accurate & relevant answers

avoid hallucinations
deprioritize LLM training data
only use language skills



Orchestration & Flow

only fetch relevant
information from reliable
datasources and integrate in
architecture



Vector + Graph

Combine vector and
graph search for retrieval
and filtering

GraphRAG Challenges

- Upfront Effort
- data modeling / graph construction
- data volume / path lengths
- supernodes
- graph representation to LLM

neo4j

Practical Examples

Tools, libraries, models

Knowledge Graph Construction from unstructured sources



Rapidly Build a Knowledge Graph for AI Use Cases



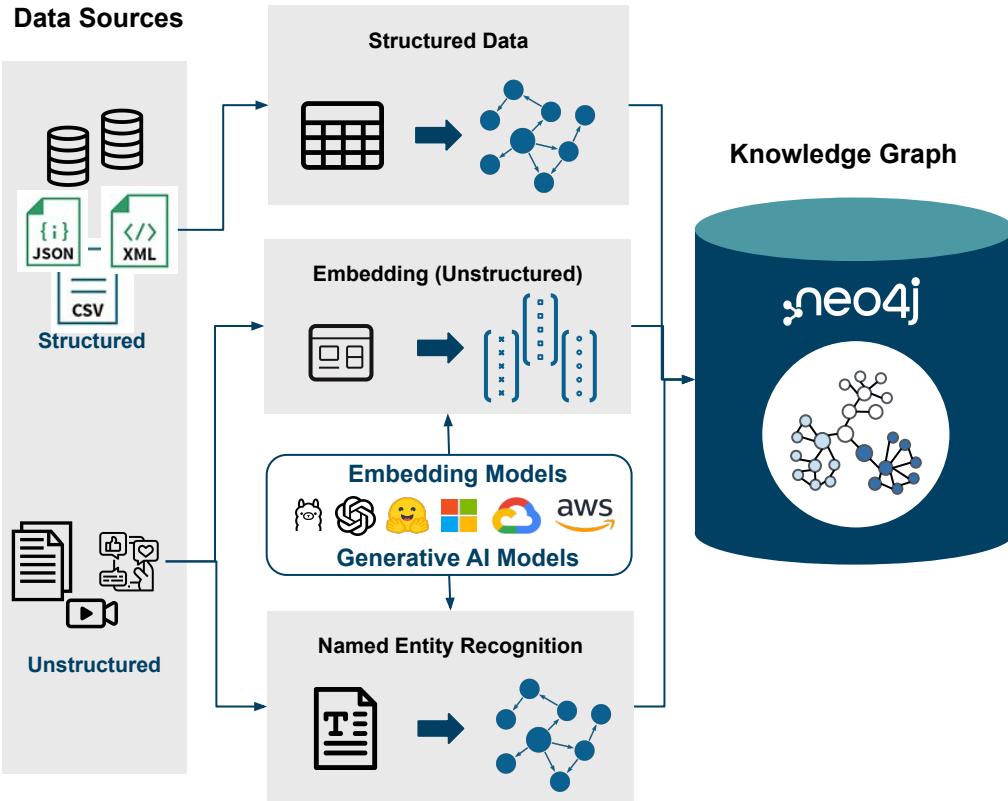
Jumpstart knowledge graph creation from unstructured data with named entity recognition



Rapidly model structured data as nodes and relationships in a knowledge graph



Ingest text, audio, and other content embeddings as node vector properties and index them for use in RAG



Automatically Build a KG for GenAI

3 Simple Steps

1

Connect to Neo4j



2

Upload Files

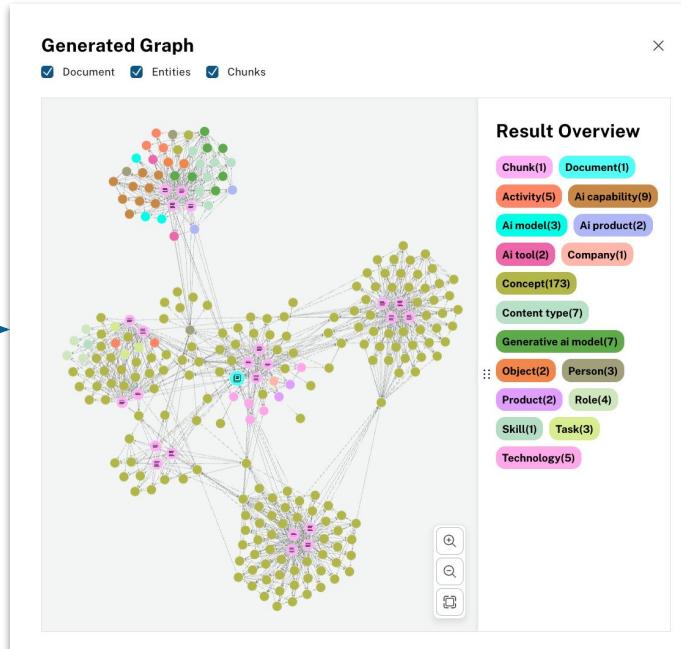
pdf, YouTube,
cloud storage,
wikipedia

3

Generate Graph



View & Explore Your Graph



Power GraphRAG

Welcome to the Neo4j Knowledge Graph Chat. You can ask questions related to documents which have been completely processed.

6/3/2024 11:44:58 AM

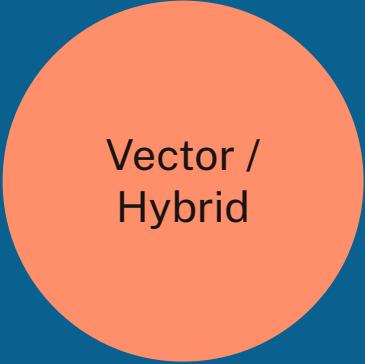


What can Generative AI do?
6/3/2024 12:05:58 PM

Generative AI can perform a wide range of tasks, including language translation, text generation, image synthesis, music composition, and even video creation. It has the ability to generate new content based on patterns and examples it has learned from training data.

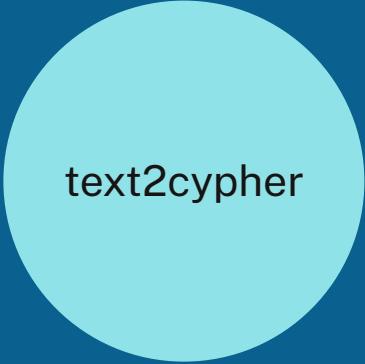
6/3/2024 12:06:18 PM

Supported Retrievers



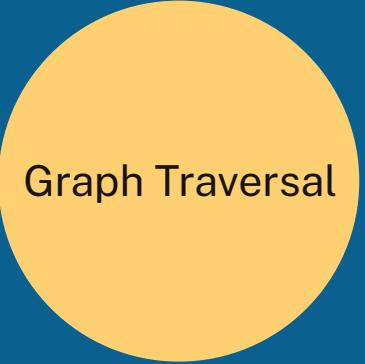
Vector /
Hybrid

embedding +
fulltext search on
document chunks



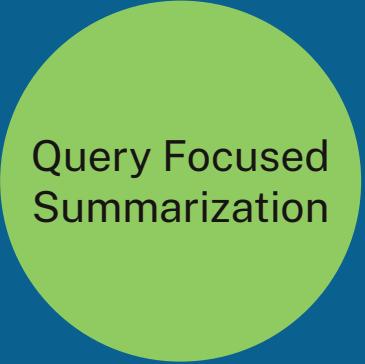
text2cypher

generate query
with schema and
question



Graph Traversal

Generic graph
traversals
starting at
documents or
entities and
expanding
patterns



Query Focused
Summarization

Using vector
search on
clustered
Community
Summaries for
global questions

Retrievers + Eval

I don't have that information right now. Is there something else I can help you with?
12.11.2024, 23:27:08

Details ↗ ⓘ Global ↙ ⓘ

Precision medicine is applied to lupus in the following ways:
Genetic Profiling: Identifying genetic markers and risks.
Biomarker Utilization: Using specific biomarkers (e.g., treatment responses).
Targeted Therapies: Developing and using drugs that target B-cell depletion therapies (e.g., Rituximab).
Individualized Treatment Plans: Customizing treatment based on organ involvement and symptom severity.
Monitoring Disease Activity: Employing scoring systems to track treatment accordingly.
Patient Reported Outcomes: Incorporating patient feedback into treatment decisions.
Longitudinal Studies: Conducting research to understand disease progression in diverse populations.
12.11.2024, 23:28:38

Details ↗ ⓘ Global ↙ ⓘ

Retrieval information

To generate this response, the process took **6.83 seconds**, utilizing **5001 tokens** with the model **gpt-4o-mini-2024-07-18** in **graph & vector & fulltext mode**.

| Sources used | Chunks | Top Entities used | Evaluation Metrics |
|---|------------------|-------------------|--------------------|
| We use several key metrics to assess the quality of our chat responses. Click the button below to view detailed scores for this interaction using ragas framework . These scores help us continuously improve the accuracy and helpfulness of our chatbots. This usually takes about 20 seconds . You'll see detailed scores shortly. | | | |
| Faithfulness: Determines How accurately the answer reflects the provided information Answer Relevancy: Determines How well the answer addresses the user's question. | | | |
| Mode | Answer Relevancy | Faithfulness | |
| Vector | 0.91 | 0.61 | |
| Graph | 0.88 | 0.82 | |
| Graph Vector Fulltext | 0.93 | 0.80 | |
| Entity Vector | 0.84 | 0.27 | |
| Global Vector | 0.00 | 0.00 | |

Showing 1-5 of 5 results Show 10 ↘

Vector

Performs semantic similarity search on text chunks using vector indexing.

● Selected

Graph

Translates text to Cypher queries for precise data retrieval from a graph database.

● Selected

Graph+Vector

Combines vector indexing and graph connections for contextually enhanced semantic search.

Fulltext

Conducts fast, keyword-based search using full-text indexing on text chunks.

Graph+Vector+Fulltext

Integrates vector, graph, and full-text indexing for comprehensive search results.

● Selected

Entity search+Vector

Uses vector indexing on entity nodes for highly relevant entity-based search.

● Selected

Global search+Vector+Fulltext

Use vector and full-text indexing on community nodes to provide accurate, context-aware answers globally.

● Selected

LLM KG Builder Demo

dev.neo4j.com/kg-builder



Neo4j connection

neo4j+s://5de0c35d.databases.neo4j.io:7687

Disconnect Show files with status New

| Name | Status | Upload |
|-----------------|-------------|----------|
| AlphaFold re... | ● Completed | ● Upload |
| Deepmind | ● Completed | ● Upload |
| Google's new... | ● Completed | ● Upload |



Retrieval information

To generate this response, in **9.43 seconds** we used **1988** tokens with the model **gpt-4o**.

Sources usedTop Entities used

Organization (16) Google Deepmind, Ai For Science Team, Francis Crick Institute

Concept (15) Digital Biology, Fusion, Quantum Chemistry

Person (13) Richard Evans, Kathryn Tunyasuvunakool, Rosana Kapeller

Technology (8) Alphafold, Dali, Foldseek

Publication (1) Science

Biological_structure (1) Nuclear Pore Complex

Ai system (1) Alphafold

Graph Schema used for Entity Extraction

Showing 1-3 of 3 results

Show16

OpenAI GP...

Generate GraphShow GraphOpen Graph with BloomDelete FilesQ&A ChatSubmit

related to documents which have been completely processed.

3.6.2024 19:13:41

What did Deepmind work on?

3.6.2024 19:15:08



DeepMind has worked on several notable projects, including: 1. **Neural Turing Machines**: Neural networks that can access external memory, resembling short-term memory in the human brain. 2.

AlphaGo: A program that beat world champion Go player Lee Sedol in 2016. 3. **AlphaZero**: A more general program that defeated the most powerful programs in Go, chess, and shogi. 4.

AlphaFold: A system that predicts protein structures, aiding in biological research. 5.

Streams: A clinical task management app co-developed with the Royal Free NHS Trust. Additionally, DeepMind has developed the Gemini project, which includes Gemini Nano, Gemini Pro, and Gemini Ultra.

3.6.2024 19:15:24



LLM Graph Builder Application

Interactive application to construct knowledge graphs from PDF, youtube transcripts etc.

Hosted on

llm-graph-builder.neo4jlabs.com

Documentation

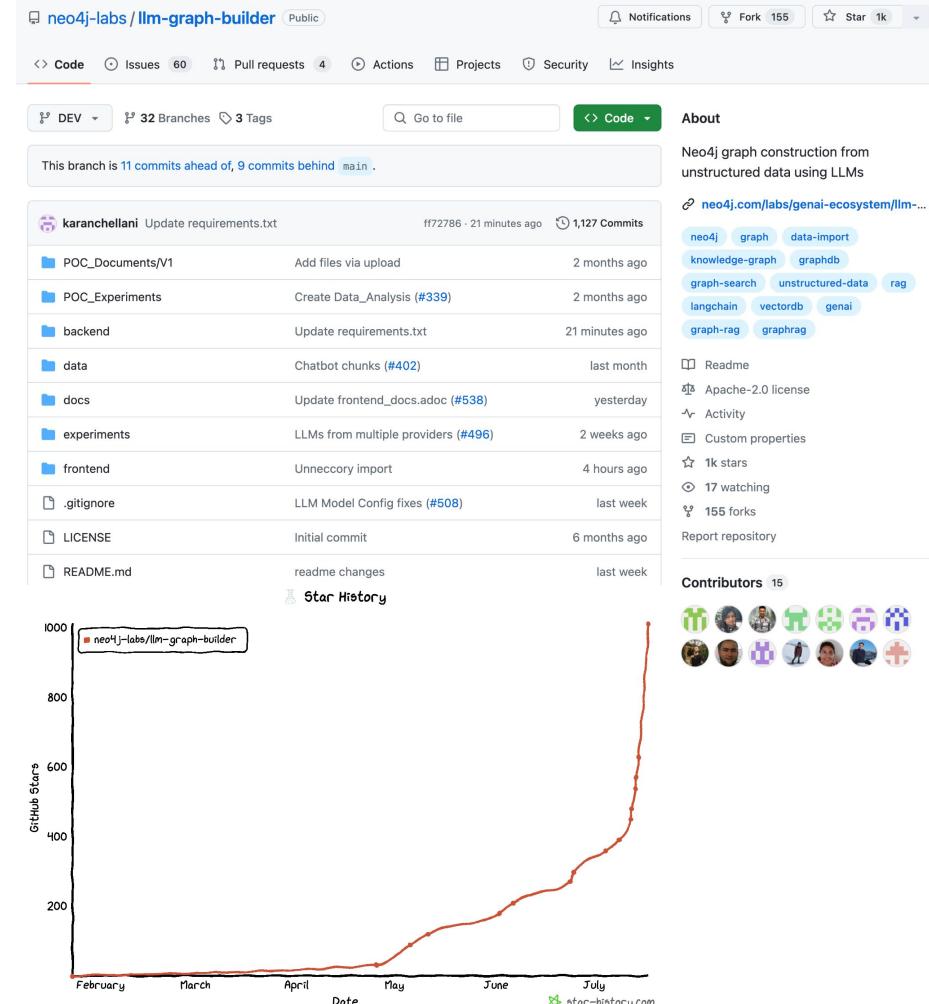
neo4j.com/labs/genai-ecosystem/llm-graph-builder

The screenshot displays the LLM Graph Builder application interface. At the top, there's a header with the Neo4j logo and a disconnect button. On the left, a sidebar features a 'Drag & Drop or browse' area with a 'Youtube' icon and an 'Amazon S3' icon. To the right of the sidebar is a table showing three completed files: 'NKc8Tr5_L3w' (5809.09 KB, PDF), 'Patrick Pichette wan...' (386.59 KB, PDF), and 'Patrick Pichette - Wi...' (0.00 KB, TEXT). Below the table is a message box asking 'What is the connection from Google to Bombardier?'. A detailed knowledge graph is centered, showing numerous nodes (represented by colored circles) and directed edges (arrows) connecting them. The graph includes nodes for Google, Bombardier, Patrick Pichette, and others. A sidebar on the right contains a 'Generate Graph' button, an 'Open Graph' button, and a text input field with a 'Submit' button. The bottom of the screen shows a footer with the Neo4j logo and the text 'Neo4j Inc. All rights reserved 2024'.

Open Source Project

github.com/neo4j-labs/llm-graph-builder

- React Front End
 - connect to Neo4j
 - load local PDF, S3 URL, YT Video
 - Wikipedia Enrichment
 - GraphRAG Chat
- Python FastAPI Back End
 - store data in neo4j
 - chunking
 - create embedding (Emb Model)
 - entity extraction (LLM, Diffbot)
 - kNN graph
- Docker(Compose) images

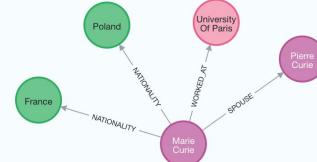


LLM Graph Extraction Library

LLM-Graph Transformer in LangChain

1. any LLM with structured output
2. use chunked documents
3. provide optional schema
4. outputs structured nodes & relationships
5. store as KG in Neo4j

[Blog Post](#) | [Docs](#)



```
from langchain_experimental.graph_transformers import LLMGraphTransformer
from langchain_openai import ChatOpenAI
from langchain_community.graphs import Neo4jGraph

graph = Neo4jGraph()
llm = ChatOpenAI(temperature=0,
                 model_name="gpt-4o-mini")

kg_transformer = LLMGraphTransformer(
    llm=llm,
    allowed_nodes=
        ["Person", "Country", "Company"],
    allowed_relationships=
        ["LOCATED_IN", "WORKED_AT"])
results =
kg_transformer.convert_to_graph_documents(
    documents)
graph.add_graph_documents(results)
```



NeoConverse

Agentic Retrieval

NeoConverse

Define Tools for Agentic Retrieval (Cypher, API, ...)

Also structured graph queries via
text2cypher using schema information from
existing graph

Supports different LLMs

Responses rendered as text or charts

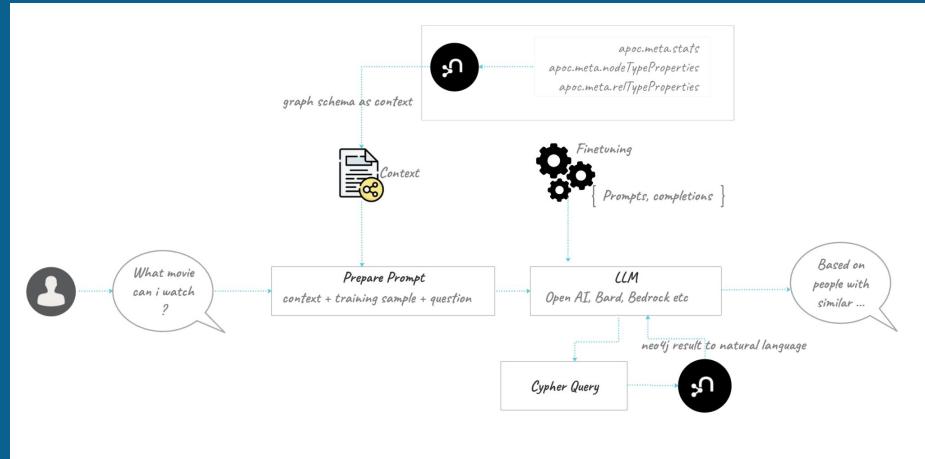
Hosted on

neoconverse.graphapp.io

Documentation

neo4j.com/labs/genai-ecosystem/neoconverse

Neo4j Inc. All rights reserved 2024

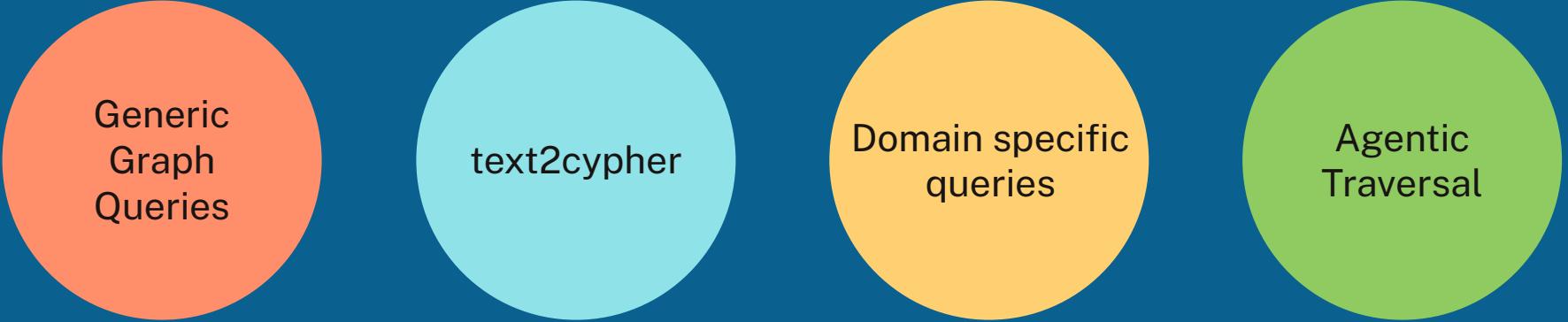


Agentic systems

- Agents give access to sets of **Tools**
- Planning for tool use, reflection
- Mix of collaborating retrieval and other agents
- Capture collected information in current state
- Evaluate current state fitness for completing task or continuing with agent loop
- Tools allow for federation across data systems
- Represent agent systems as executable graphs



Agentic GraphRAG Tools



Generic
Graph
Queries

n-hop,
neighborhood,
shortest path,
communities/clus-
ters

text2cypher

generate query
with schema and
question

Domain specific
queries

Domain queries
by SME for
specific agents

Agentic
Traversal

Agentic traversal
incrementally
collect
information, see
GraphReader

Tools for Diffbot Companies Dataset

Industries Property Search

List of Industry category names

 EDIT

 REMOVE

Companies Full text Search

List of Companies returns (id, name, summary) by fulltext search
by 'industry' name

 EDIT

 REMOVE

companies_in_industry Graph Traversal

Companies (id, name, summary) in a given industry by
`industry` name

 EDIT

 REMOVE

articles_in_month Property Search

List of Articles (id, author, title, date, sentiment) in a 30
day timeframe from the given `date` (yyyy-mm-dd)

 EDIT

 REMOVE

article Property Search

Single Article details (id, author, title, date, sentiment,
site, summary, content) by article id

 EDIT

 REMOVE

people_at_company Property Search

People (name, role) associated with a company by
company id

 EDIT

 REMOVE

fuzzy_articles_search Vector Search

Search for articles by a similar text using text embedding
search

 EDIT

 REMOVE

Name
`companies_in_articles`

Tags
`Graph Traversal`

Description

Companies (id, name, summary) mentioned in articles by list of article ids

Parameters Type

object

Property Name

`article_ids`

Property Type

`string`

Property Description

`Comma separated string of artic`

New Property Name

New Property Type

New Property Description

Required Fields

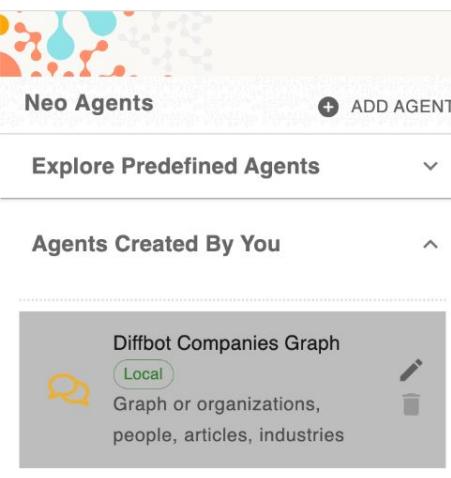
`article_ids`

Category

Cypher Execution

```
1 WITH [id in split('${article_ids}',',') | trim(id)] as ids
2 MATCH (:Article)-[:MENTIONS]->(c)
3 WHERE a.id in ids AND not exists { (c)<-[:HAS_SUBSIDIARY]-() }
4 RETURN c.id as id, c.name as name, c.summary as summary
5
```

Give me top 3 articles by sentiment in January 2023 and the companies mentioned in those articles and the people working at these in which roles?



Neo Agents + ADD AGENT

Explore Predefined Agents ▼

Agents Created By You ^

Diffbot Companies Graph Local

Graph or organizations, people, articles, industries



neo4j NeoConverse 🔍
A GenAI copilot to converse with your neo4j graph data

Chat

Hey there! NeoConverse uses generative AI to help you communicate with neo4j database using natural language. If you encounter any inaccurate responses, please report them using the report icon. Let's start chatting!

Give me top 3 articles by sentiment in January 2023 and the companies mentioned in those articles and the people working at these in which roles?

Here are the top 3 articles by sentiment in January 2023, the companies mentioned in those articles, and the people working at these companies in their respective roles:

Article: "Industry Veteran Jay Detting Joins Hero Digital as CEO to Accelerate the Company's Next Phase of Growth"

Sentiment: 0.972
Company: Hero Digital
People:

- Jay Detting - CEO
- Timothy T. Hall - Board Member
- Joost Thesseling - Board Member
- Fred Iseman - Board Member

Article: "Hero Digital Acquires Omnichannel Commerce, a Leader in Salesforce Commerce"

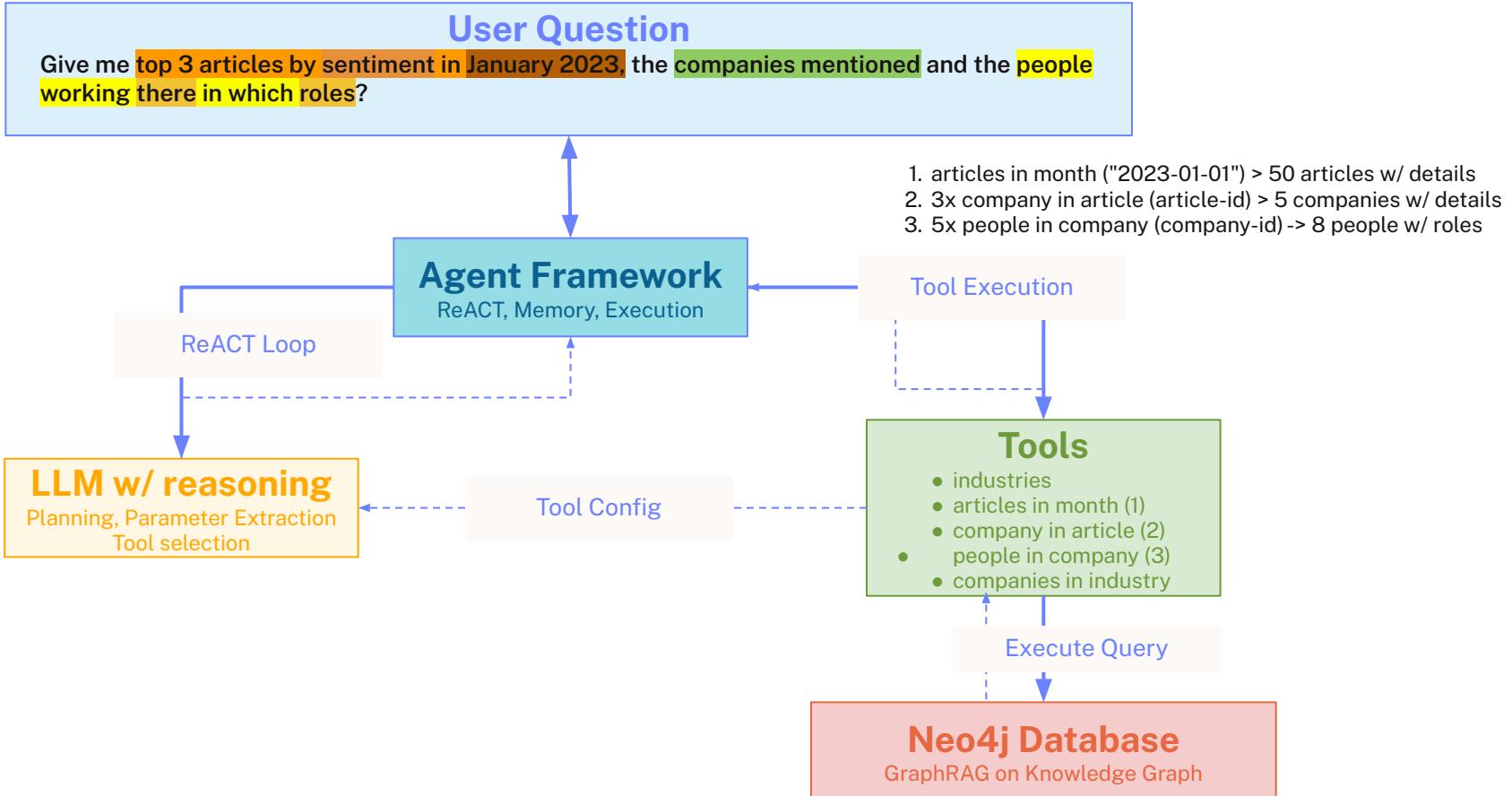
Sentiment: 0.968
Company: Hero Digital
People:

- Jay Detting - CEO
- Timothy T. Hall - Board Member
- Joost Thesseling - Board Member
- Fred Iseman - Board Member

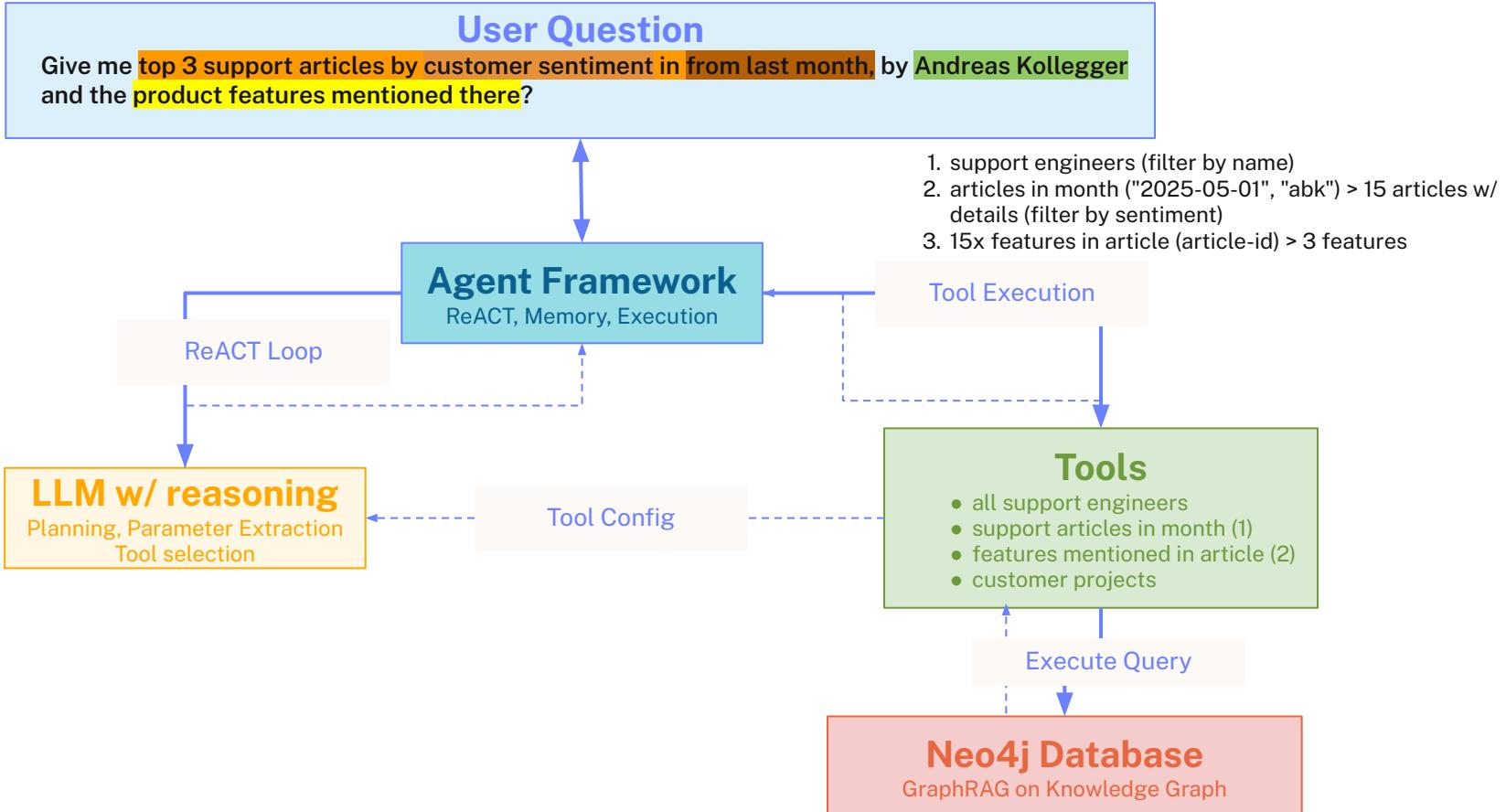
Article: "Amax Engineering: Scalable HPC and GPU Reference Design Solutions with 4th Gen Xeon Scalable Processors"

Sentiment: 0.906
Company: AMAX
People:

Agent Loop with Tool Calling



Agent Loop with Tool Calling



Behind the scenes - Tool calling

```
chat-history for message: ▶ Array(2)
```

```
tool.categorical_value ▶ Array(1)
```

```
query from template conversion:    match (a:Article)
    where date('2023-01-01') <= date(a.date) < date('2023-01-01') + duration('P1M')
    return a.id as id, a.author as author, a.title as title, toString(a.date) as date, a.sentiment as sentiment
    limit 25
```

```
chat-history for tool: ▶ (4) [...], [...], [...], [...]
```

```
tool.categorical_value ▶ [...]
```

```
query from template conversion: WITH [id in split('ART222983431438,ART234375685382,ART206570973340','|') | trim(id)] as ids
MATCH (a:Article)-[:MENTIONS]->(c)
WHERE a.id in ids AND not exists { (c)<-[:HAS_SUBSIDARY]-() }
RETURN c.id as id, c.name as name, c.summary as summary
```

```
chat-history for tool: ▶ (6) [...], [...], [...], [...], [...], [...]
```

```
tool.categorical_value ▶ [...]
```

```
query from template conversion:    MATCH (c:Organization)-[role]-(p:Person) WHERE c.id = 'E10mxYzHiMTu8cMp8UNGvVw'
    RETURN replace(type(role),"HAS_","",) as role, p.name as name
```

```
tool.categorical_value ▶ [...]
```

```
query from template conversion:    MATCH (c:Organization)-[role]-(p:Person) WHERE c.id = 'E1KSTKSP70wGMm0x94h6pxA'
    RETURN replace(type(role),"HAS_","",) as role, p.name as name
```

```
chat-history for tool: ▶ (9) [...], [...], [...], [...], [...], [...], [...], [...], [...]
```

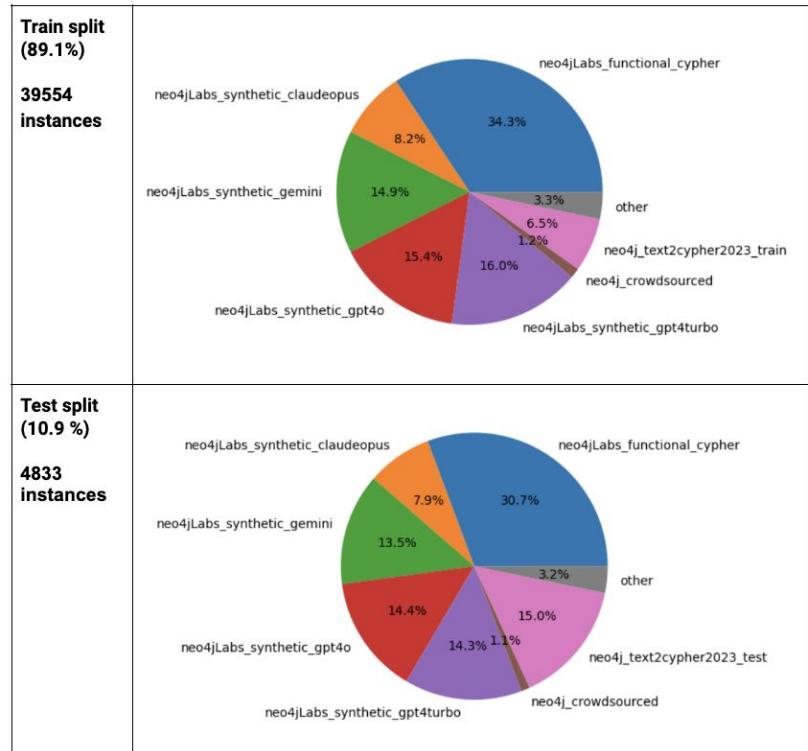
Text2Cypher

Graph Query Generation

Neo4j-Text2Cypher Dataset'24

- Overall combined dataset
 - Total: 44387 instances
 - Training set: 39554 instances
 - Test set: 4833 instances
- # Instances with database access
 - Used in evaluation
 - Total: 24564 instances (55.34%)
 - Training set: 22093 instances (55.85%)
 - Test set: 2471 instances (51.12%)

Distribution of data by datasource



Benchmarking using Neo4j-Text2Cypher Dataset'24

- Used **14 models** for benchmarking
 - 4 Finetuned models
 - 10 Foundational models
 - 4 Open models
 - 6 Closed models

| Index | Type | Name | Baseline-model | Dataset (if fine-tuned) | |
|----------------------------|----------------------|--|--------------------------------------|-------------------------------------|--|
| Fine-tuned models | | | | | |
| 1 | HF | hf_finetuned_lakkeo_stable_cypher_instruct3B | Stability AI/Stable-code-instruct-3b | neo4jLabs_synthetic_claudeopus | |
| 2 | HF | hf_finetuned_tomasonjo_text2cypher | Meta/Llama-3-8b-Instruct | neo4jLabs_synthetic_gpt4o (cleaned) | |
| 3 | HF | ** hf_finetuned_neo4j_text2cypher_23_codellama | Meta/Codellama-13B | neo4j_text2cypher20_23_train | |
| 4 | OpenAI | ** openai_finetuned_neo4j_text2cypher_23_gpt3_5 | OpenAI/GPT-3.5 | neo4j_text2cypher20_23_train | |
| Foundational models | | | | | |
| | Open models | | | N/A | |
| 5 | HF | hf_foundational_gemma2_9B_it | Google/Gemma-2-9B-it | | |
| 6 | HF | hf_foundational_codegemma_7B_it | Google/CodeGemma-7B-it | | |
| 7 | HF | hf_foundational_meta_llama3_1_8B_instruct | Meta/LLama-3.1-8B-instruct | | |
| 8 | HF | hf_foundational_codeLlama_7B_instruct_hf | Meta/CodeLLama-7B-instruct | | |
| | Closed models | | | | |
| 9 | OpenAI | openai_gpt3_5 | OpenAI/GPT-3.5 | | |
| 10 | OpenAI | openai_gpt4_o | OpenAI/GPT-4o | | |
| 11 | OpenAI | openai_gpt4_o_mini | OpenAI/GPT-4o-mini | | |
| 12 | Google | gemini-1.0-pro-002 | VertexAi/Gemini-1.0-Pro | | |
| 13 | Google | gemini-1.5-flash-001 | Google/Gemini-1.5-Flash | | |
| 14 | Google | gemini-1.5-pro-001 | Google/Gemini-1.5-Pro | | |

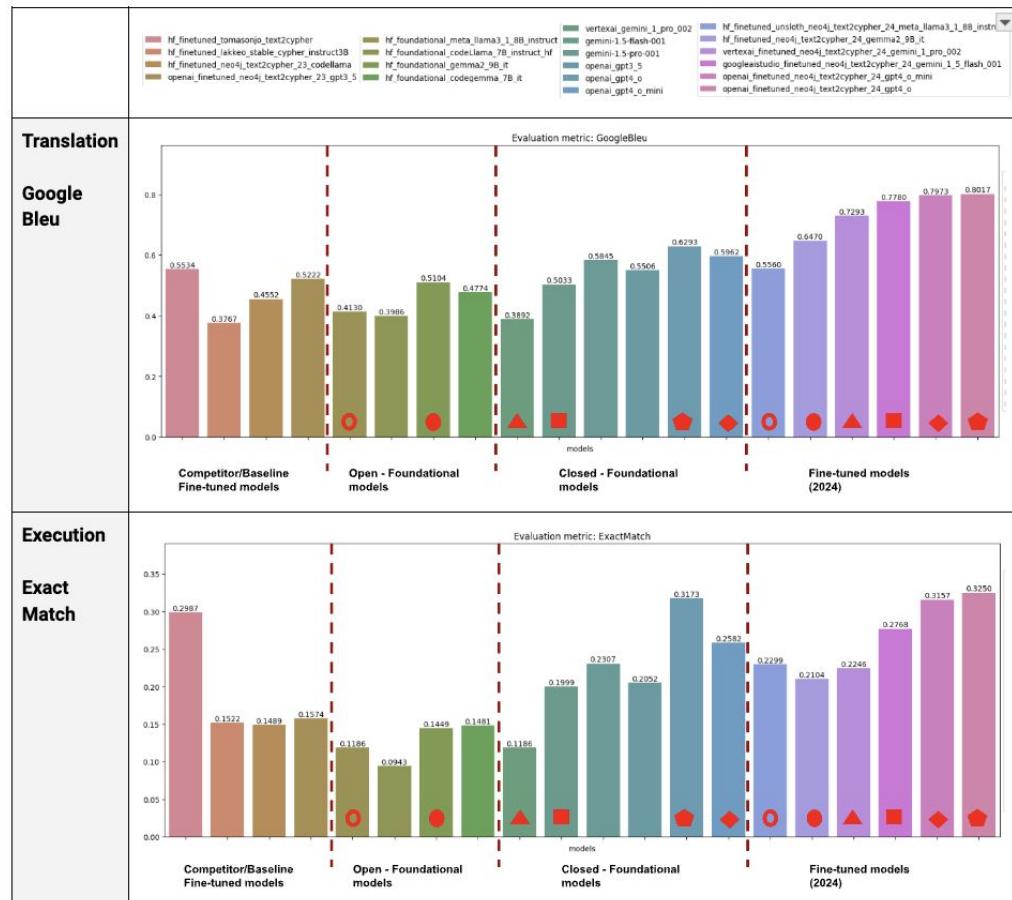
- The models identified with (**) are Neo4j models.

neo4j.com/developer-blog/benchmarking-neo4j-text2cypher-dataset/

Neo4j-Text2Cypher Finetuned Models'24

Best performing models:

- **Translation-based evaluation**
 - Finetuned-OpenAI/**Gpt4o**
 - Finetuned-OpenAI/**Gpt4o-mini**
 - Finetuned-GoogleAIStudio/**Gemini-1.5-Flash-001**
- **Execution-based evaluation**
 - Finetuned-OpenAI/**Gpt4o**
 - Finetuned-OpenAI/**Gpt4o-mini**



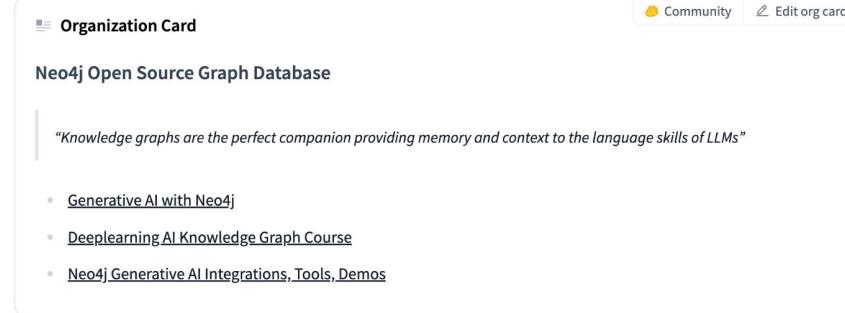
Dataset & Models on HuggingFace

 **Neo4j** Company Verified
<https://neo4j.com> [neo4j](#) <https://github.com/neo4j> · Upgrade to [Enterprise](#)

+ New Activity Feed Organization settings Following 35


AI & ML interests
Graphs, Graph database, Knowledge Graphs, Graph Data Science


Team members 26

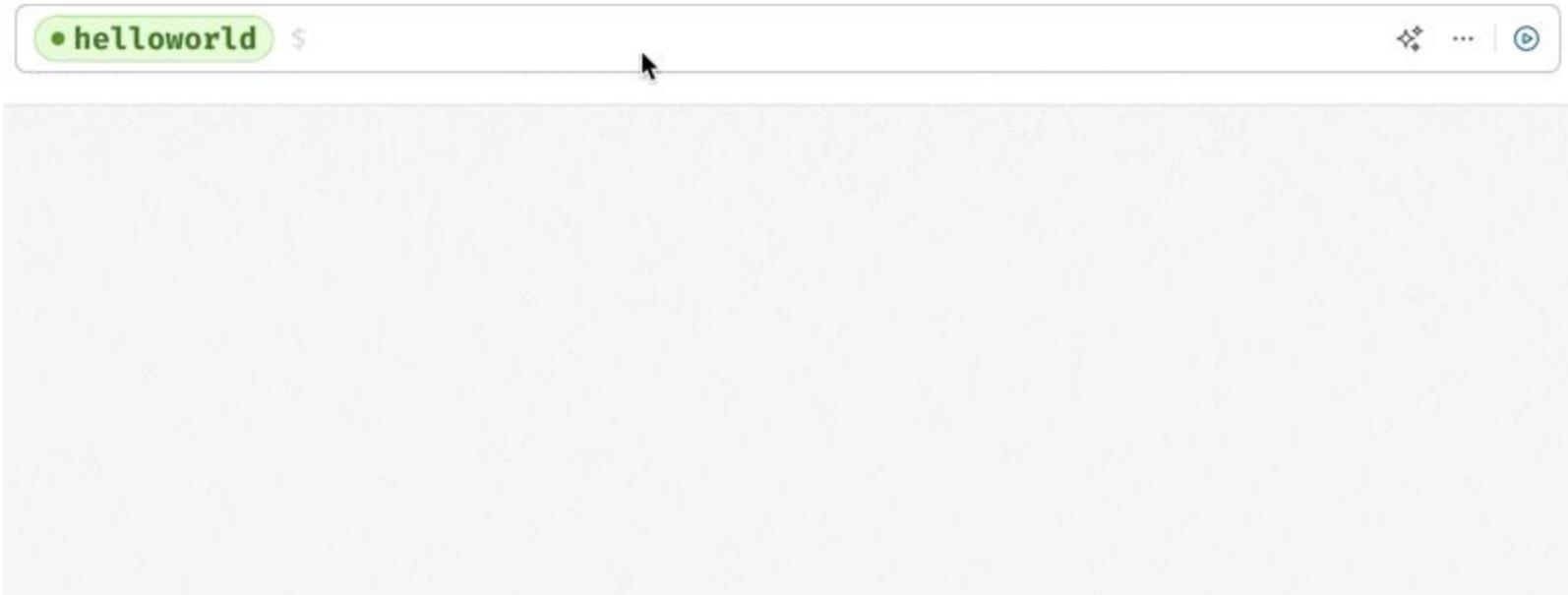

Organization Card
Neo4j Open Source Graph Database
"Knowledge graphs are the perfect companion providing memory and context to the language skills of LLMs"
• [Generative AI with Neo4j](#)
• [DeepLearning AI Knowledge Graph Course](#)
• [Neo4j Generative AI Integrations, Tools, Demos](#)


Models 3 Sort: Recently updated
neo4j/neo4j_llama318b_finetuned_mer... private
Text Generation · Updated 19 days ago · ↴ 2
neo4j/text2cypher-llama3_1_8B_instr... private
Text2Text Generation · Updated 27 days ago · ↴ 2
neo4j/text2cypher-gemma-2-9b-it-fin... private
Text2Text Generation · Updated Oct 9 · ↴ 2


Datasets 1
neo4j/text2cypher-2024v1
Viewer · Updated 5 days ago · ↴ 25 · ❤ 2

neo4j.com/developer-blog/introducing-neo4j-text2cypher-dataset/

Powering Neo4j CoPilots



neo4j-graphrag package

Construction + Retrieval

GraphRAG Python package

End-to-end workflows for faster, simpler, and more effective GenAI-KG development, supported by Neo4j

Build Knowledge Graphs

transform documents into GenAI-ready knowledge graphs with vectors

Implement KG Retrievers

Easily implement powerful knowledge graph retrievers using graph traversals and vector search and hybrid search

Create GraphRAG Pipelines

Combine entire RAG process with retrieval, prompt & LLM using just a few lines of code

GraphRAG Python package Workflow

```
import neo4j
from neo4j_graphrag.llm import OpenAILLM as LLM
from neo4j_graphrag.embeddings.openai import OpenAIEmbeddings as Embeddings
from neo4j_graphrag.experimental.pipeline.kg_builder import SimpleKGPipeline
from neo4j_graphrag.retrievers import VectorRetriever
from neo4j_graphrag.generation.graphrag import GraphRAG

# 0. Connect to Neo4j Database
neo4j_driver = neo4j.GraphDatabase.driver(NEO4J_URI,
                                         auth=(NEO4J_USERNAME, NEO4J_PASSWORD))

# 1. Build KG and Store in Neo4j Database
ex_llm=LLM(
    model_name="gpt-4o-mini",
    model_params={
        "response_format": {"type": "json_object"},
        "temperature": 0
    })

embedder = Embeddings()

kg_builder_pdf = SimpleKGPipeline(
    llm=ex_llm,
    driver=neo4j_driver,
    embedder=embedder,
    from_pdf=True
)
await kg_builder_pdf.run_async(file_path='precision-med-for-lupus.pdf')
```

```
# 2. Create KG Retriever & GraphRAG Class
vector_retriever = VectorRetriever(
    neo4j_driver,
    index_name="text_embeddings",
    embedder=embedder
)

llm = LLM(model_name="gpt-4o")
rag = GraphRAG(llm=llm, retriever=vector_retriever)

# 3. Run
response = rag.search("How is precision medicine applied to Lupus?")
print(response.answer)

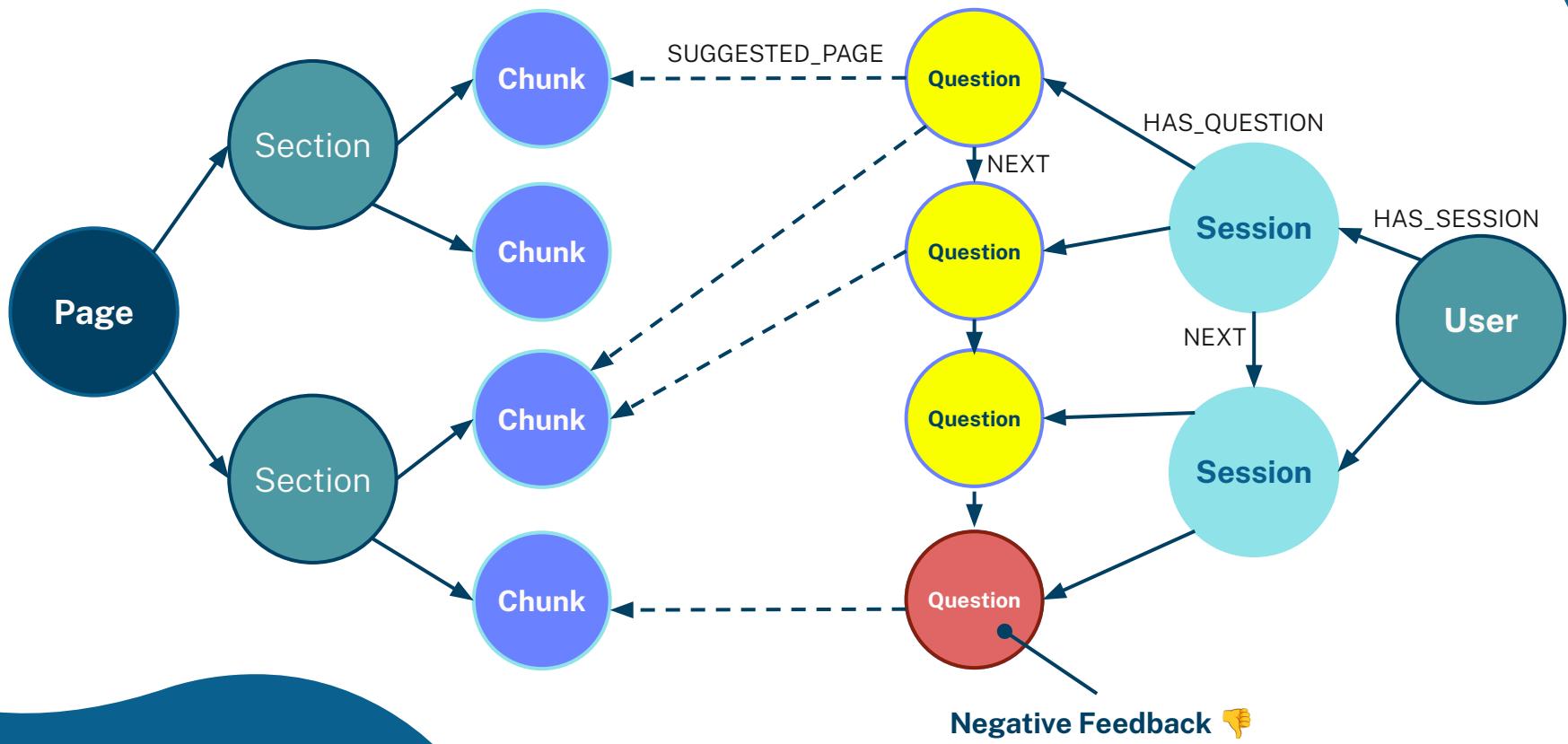
—
Precision medicine in systemic lupus erythematosus (SLE) is an evolving approach
that aims to tailor treatment based on individual genetic, epigenetic, and
pathophysiological characteristics....
```

Resources

- **Demo Code:** github.com/neo4j-product-examples/graphrag-python-examples
- **Free Course:** graphacademy.neo4j.com/courses/genai-workshop-graphrag/
- **Docs:** neo4j.com/docs/neo4j-graphrag-python
- **GenAI Learning & Building Resources:** neo4j.com/generativeai/#resources

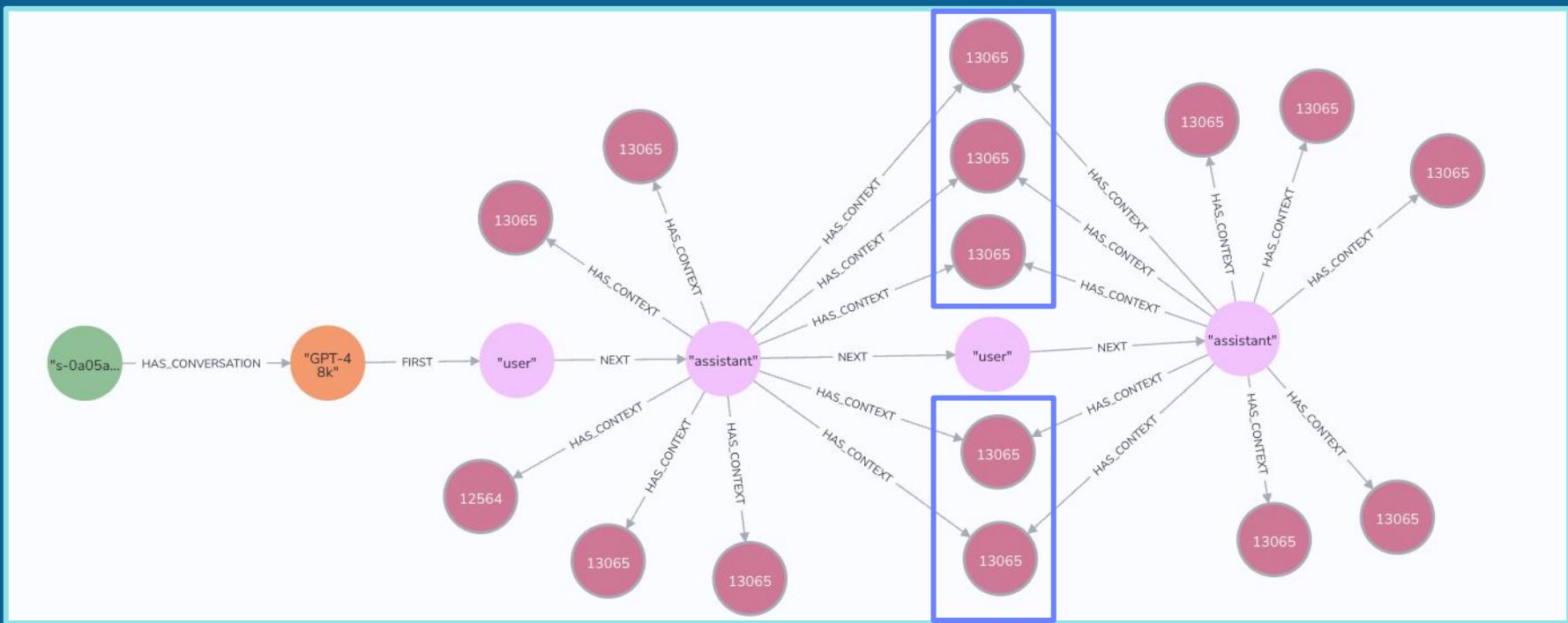
Conversation Memory

Conversation History as a Graph



Grounded LLM Conversations are Graphs

Graphs enable logging of LLM conversations ***in the same database*** as the context documents and with defined relationships.



Impact of Knowledge Graphs in GenAI apps

Connect



Context rich, connected view of your data that enables **easier decision making**

Enrich



Enhance your data with **graph data science**, text embeddings, and additional derived context

Consume



Ground responses with information and context in the graph

Improve search relevance combining vector search and graph traversals

Capture



Log, visualize, and analyze LLM interactions to improve application deployments

Graphs capture context



Context rich, connected view of your data that enables easier decision making



Enrich your data with graph data science, text embeddings, and additional derived context



Ground responses with information and context in the graph

Improve **search relevance** combining vector search and graph traversals

How do you represent and access relevant context beyond text fragments?

Graphs Enable Explainable AI

How do you ensure a high-quality production environment with LLMs?



Logging user interactions in the same database as the context



Visualizing conversations within context



Analysing LLM performance and identifying opportunities for improvement

LLM Courses on Neo4j GraphAcademy

 neo4j graphacademy

Neo4j & Large Language Model (LLM) Fundamentals

Learn how to use Neo4j with Large Language Models



r.neo4j.com/ga-llm-fun



 neo4j graphacademy

Build an Neo4j-backed Chatbot using Python

Build a chatbot using Neo4j, Langchain and Streamlit



r.neo4j.com/ga-llm-py



Your Takeaway - graphrag.com

- Graph Models
- GraphRAG Retrieval Patterns
- Papers
- Open for Contribution

GraphRAG with a Knowledge Graph

Connect the dots for better answers

What is GraphRAG? →

What is a Knowledge Graph? →

