



Graph Analytics Workshop

Alison Cossette - Developer Advocate Data Science





Data Sources

The right data for the right question.

github.com/alisoncossette/
Graph-Analytics-Workshop

A Generative AI application
uses an LLM
to provide **responses**
to **user prompts**

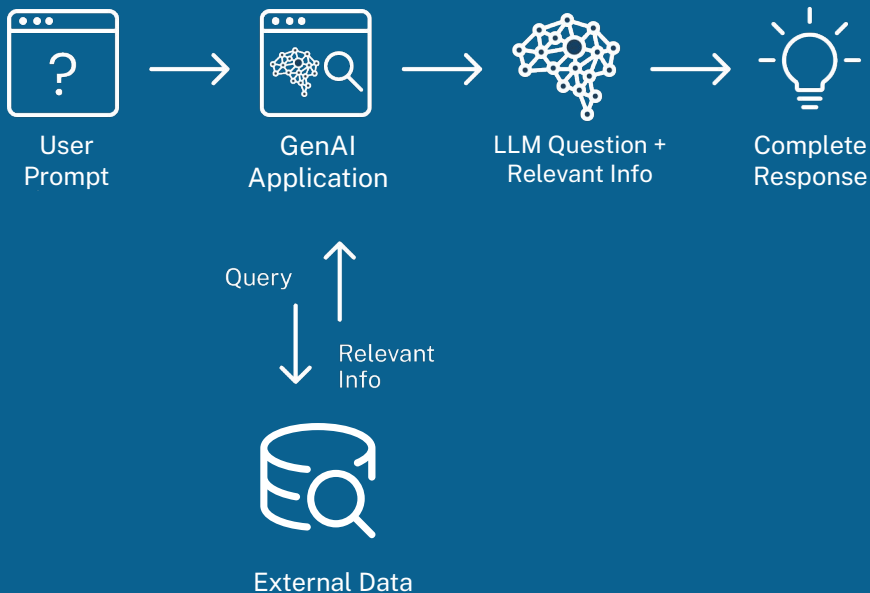
(aka ChatGPT)



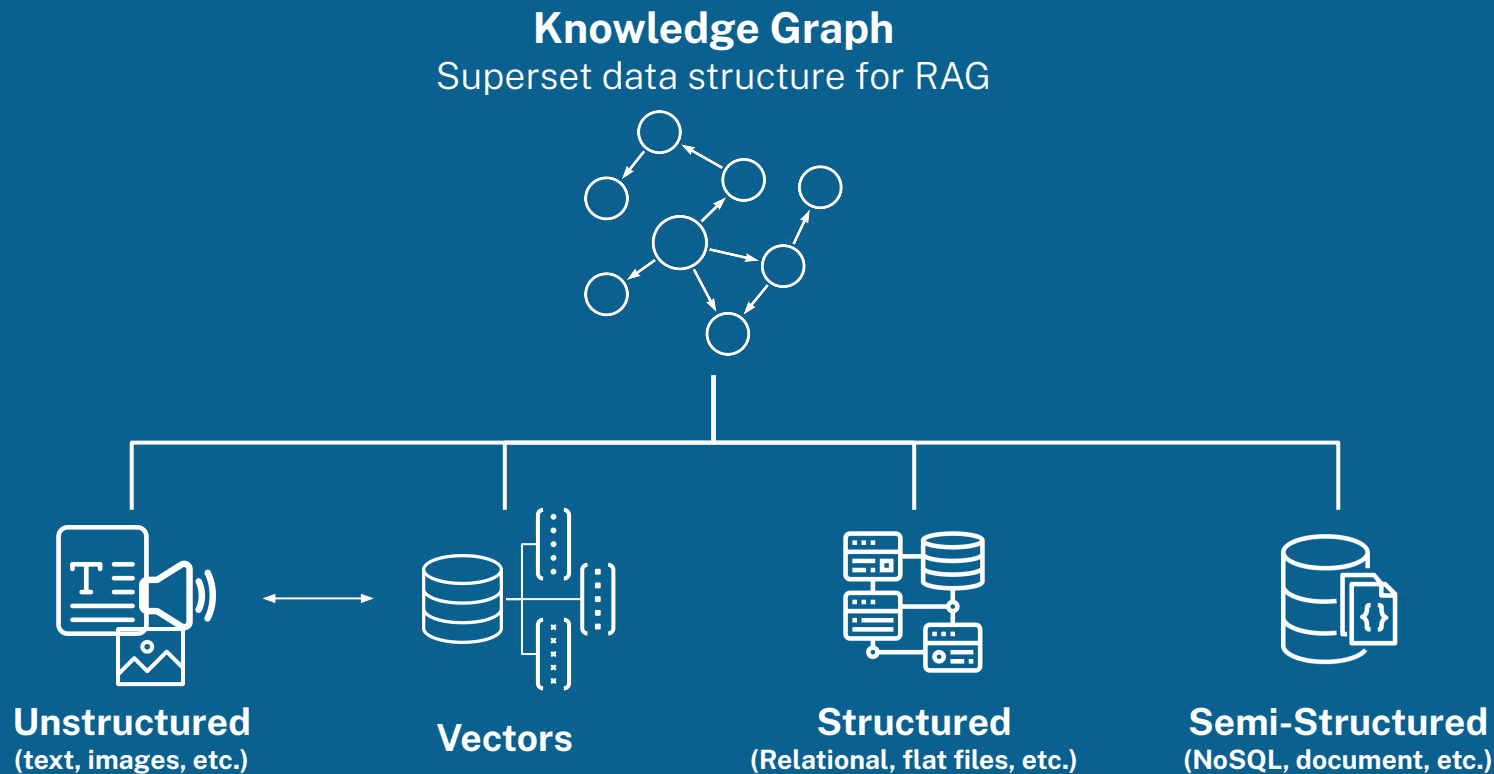
RAG augments the LLM by intercepting a **user's prompt**,

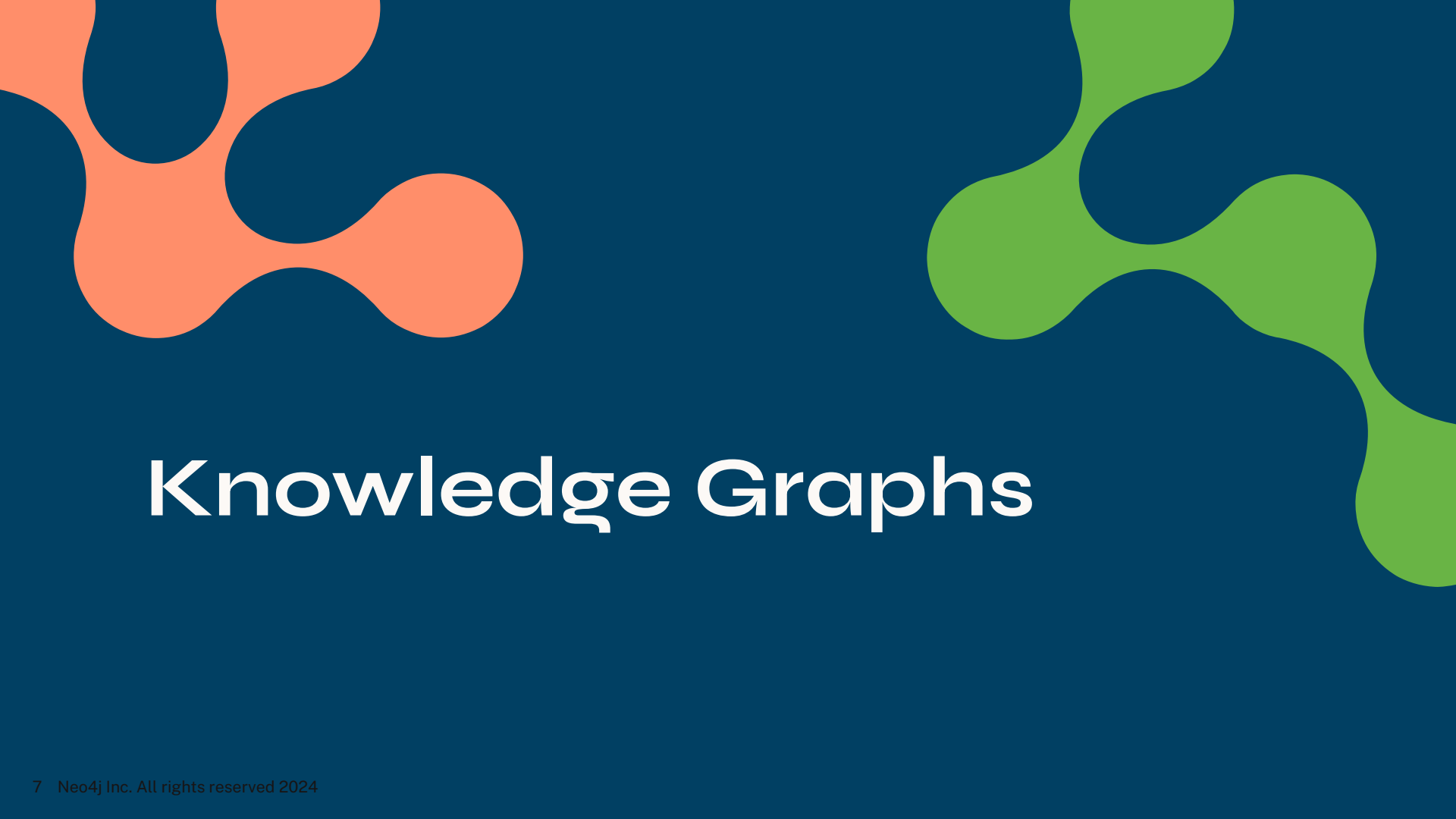
then making a **query to external data**,

then passing relevant results from the query back to the LLM for a **complete, curated response**.



What to use for External Data?





Knowledge Graphs

Neo4j Graph Components

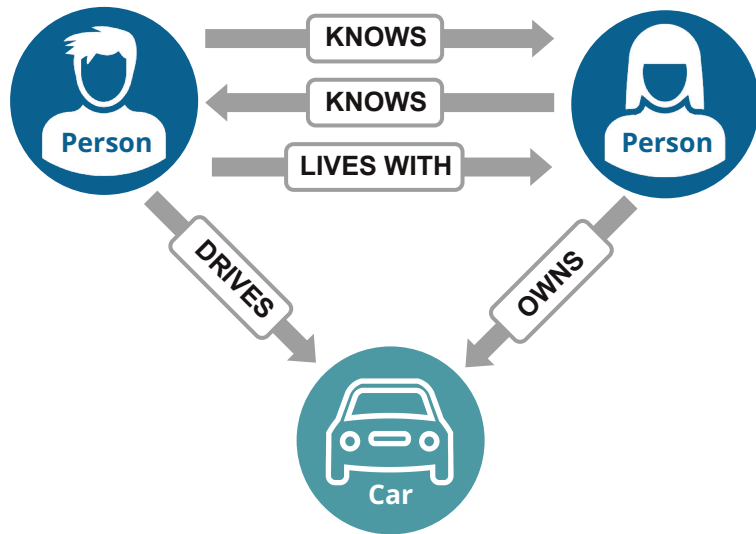
Nodes represent entities in the graph



Neo4j Graph Components

Nodes represent entities in the graph

Relationships represent associations or interactions between nodes



Neo4j Graph Components

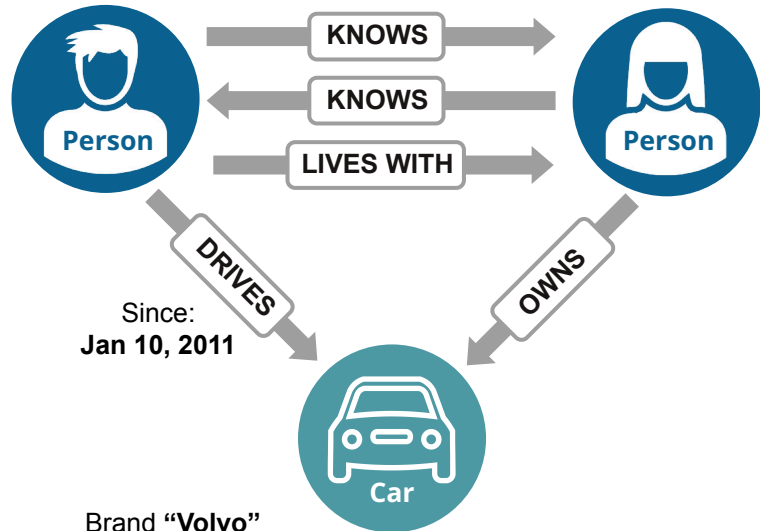
Nodes represent entities in the graph

Relationships represent associations or interactions between nodes

Properties represent attributes of nodes or relationships including vectors

Name: "Andre"
Born: May 29, 1970
Twitter: "@dan"

Name: "Mica"
Born: Dec 5, 1975



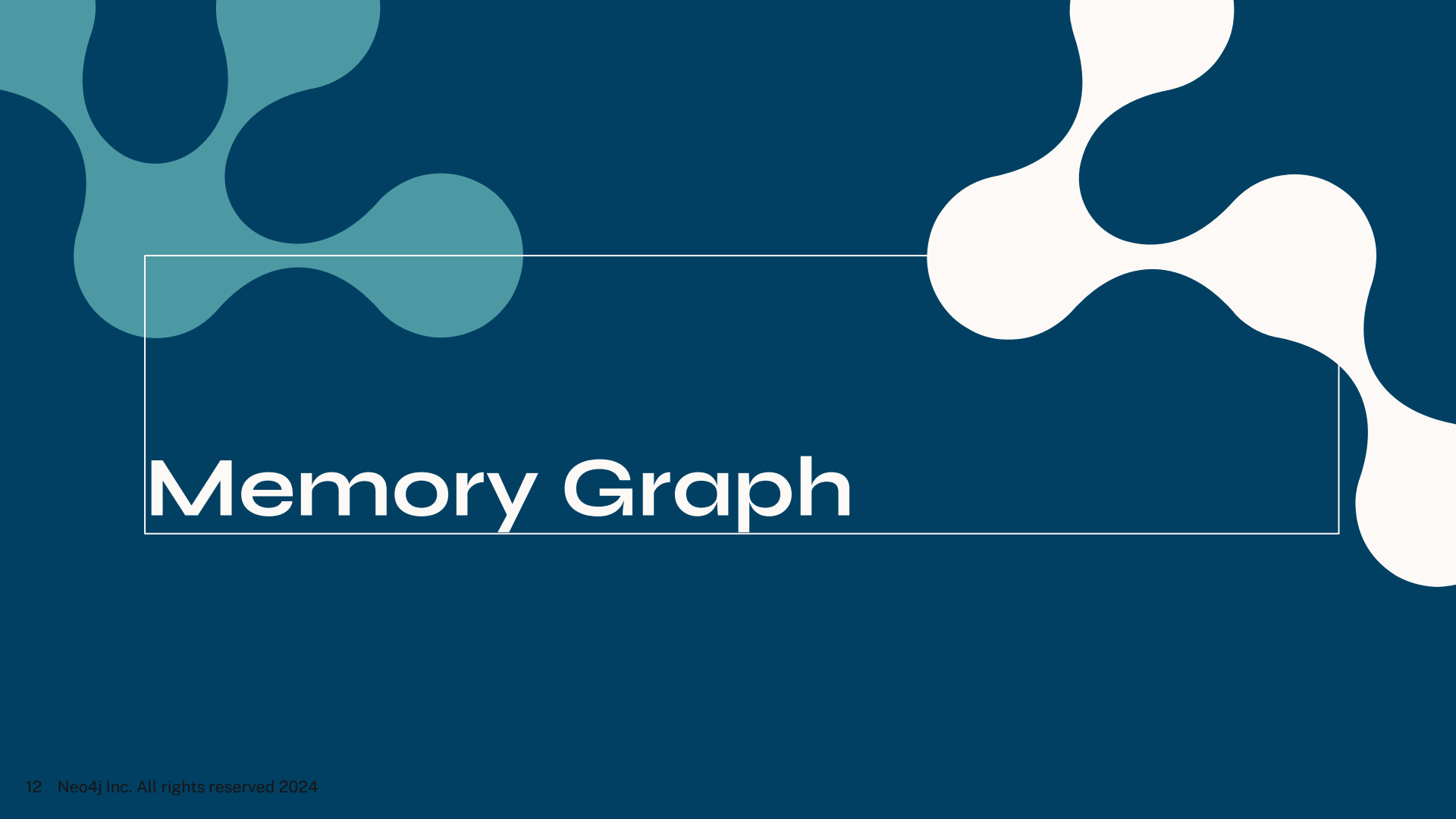
Brand "Volvo"
Model: "V70"
Description: "An executive car manufactured and..."
DescEmbedding: [0.1, -0.3, 0.4, ..., -0.7]
DescSource: "https://en.wikipedia.org/wiki/Volvo_V70"

KNOWLEDGE GRAPH



Source Legend

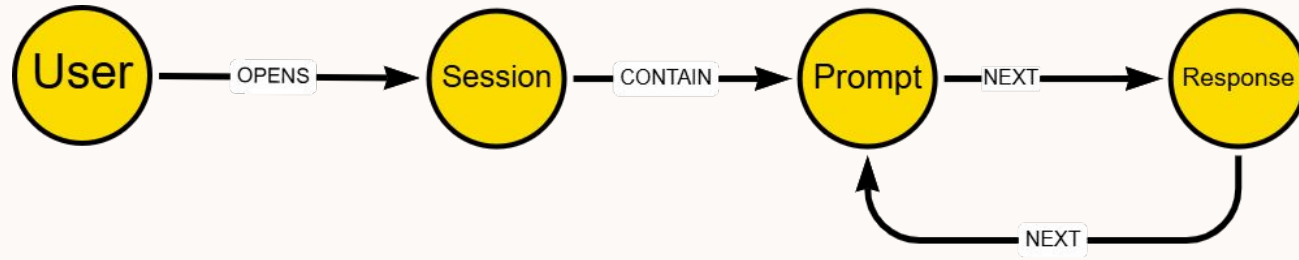
- = Structured Stores
- = Unstructured Data
- = Application



Memory Graph

KNOWLEDGE GRAPH

APPLICATION GRAPH



Source Legend

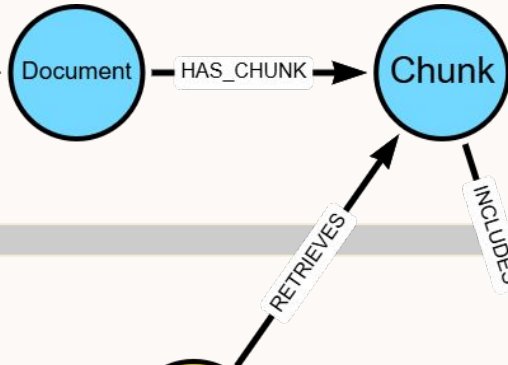
- = Structured Stores
- = Unstructured Data
- = Application

KNOWLEDGE GRAPH

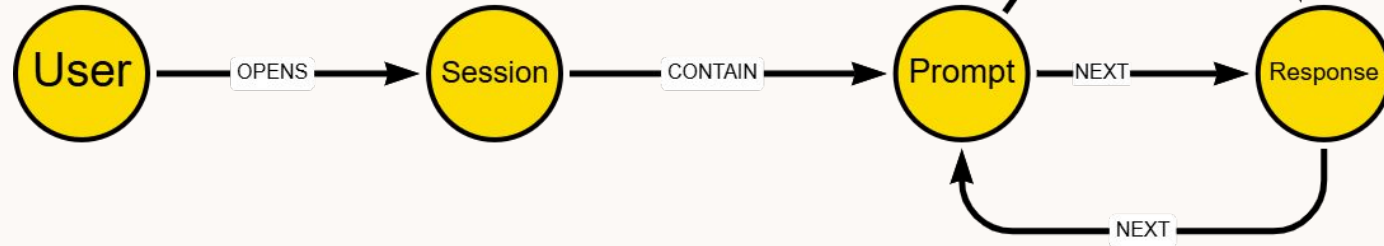
DOMAIN GRAPH



LEXICAL GRAPH



MEMORY GRAPH



Source Legend

- = Structured Stores
- = Unstructured Data
- = Application

KNOWLEDGE GRAPH

DOMAIN GRAPH



LEXICAL GRAPH



MEMORY GRAPH



Source Legend

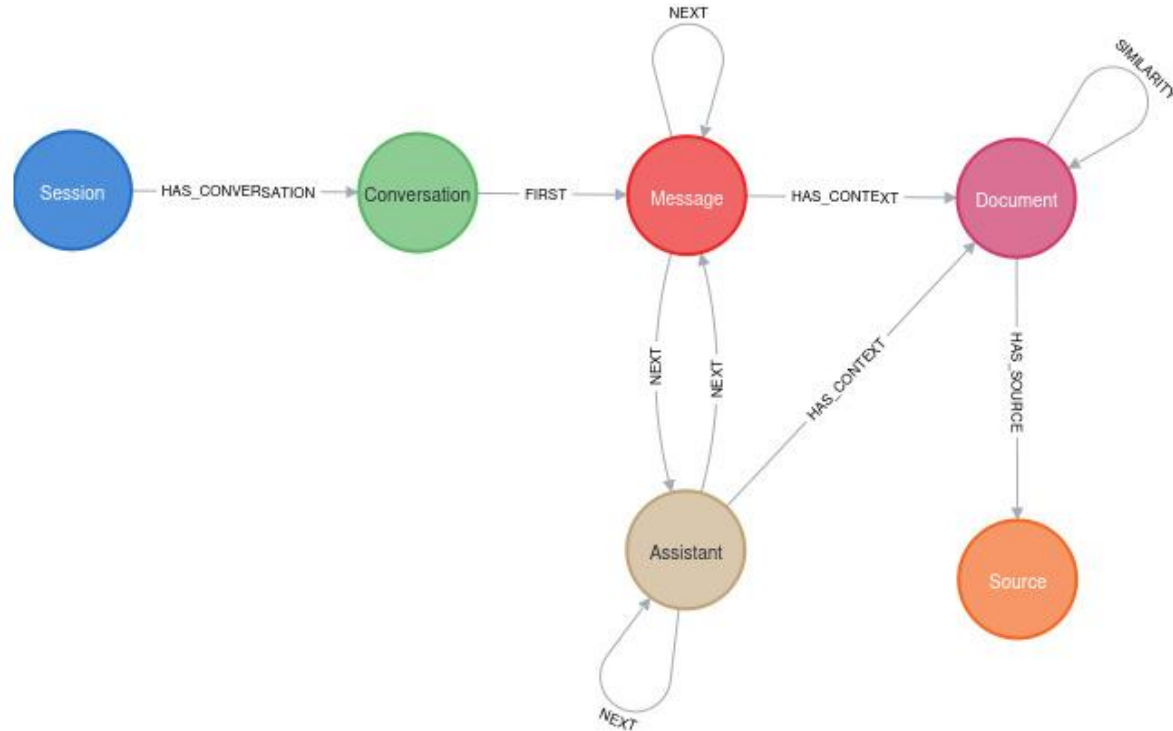
- = Structured Stores
- = Unstructured Data
- = Application



Data Understanding

Exploring your unstructured data

Grounding Data Sources

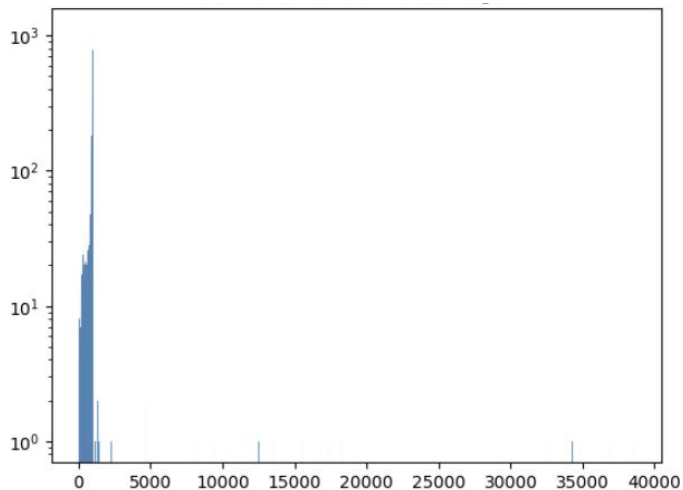


Grounding Data Sources

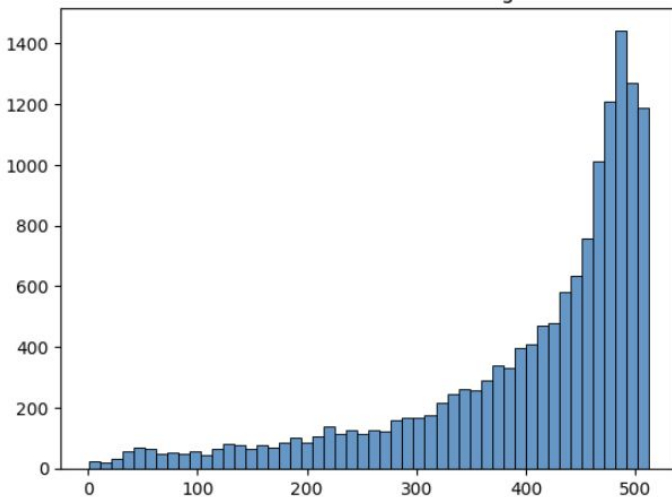
- Dataset (all public data sources)
 - 1,150 documents of official Neo4j documentation
 - Developer blogs
 - Support knowledge base
 - Github
- Split this text into 15,000 embedded text 'chunks'
 - 512 chunk size
 - LangChain Recursive Text Splitter
 - Embeddings via GCP
 - URL of each chunk for LLM citation

EDA on Source Documents

EDA on source documents and document chunks is a critical step before generating embeddings and loading them into the database.



Text length distribution with chunking strategy errors



Text length distribution with corrected chunking strategy

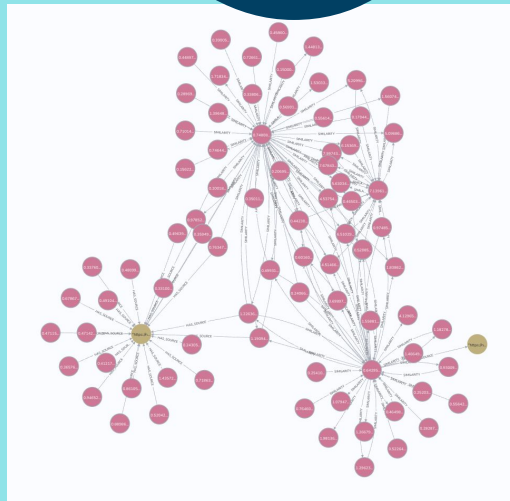
EDA with Graphs and GDS

CONNECT

CLUSTER

CURATE

- **KNN Similarity** create relationships between the most similar chunk
- **Community Detection** and creates clusters based on similarity relationships
- Curate the grounding data set via techniques that work **at scale**



*Similarity Graph of Context Document Chunks (red)
with Source URLs (gold)*

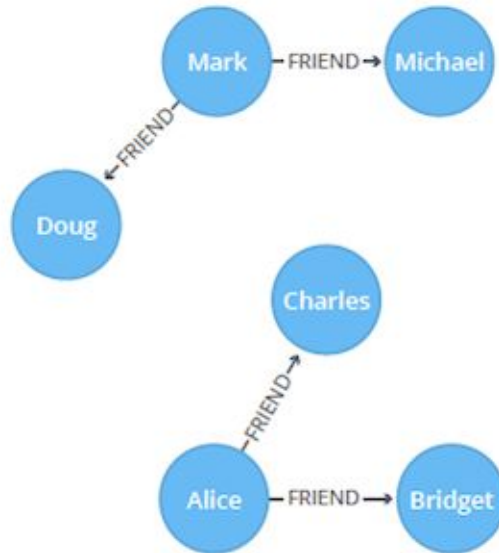
What is Community Detection?

Community Detection algorithms
identify node groupings

How are nodes clustered?

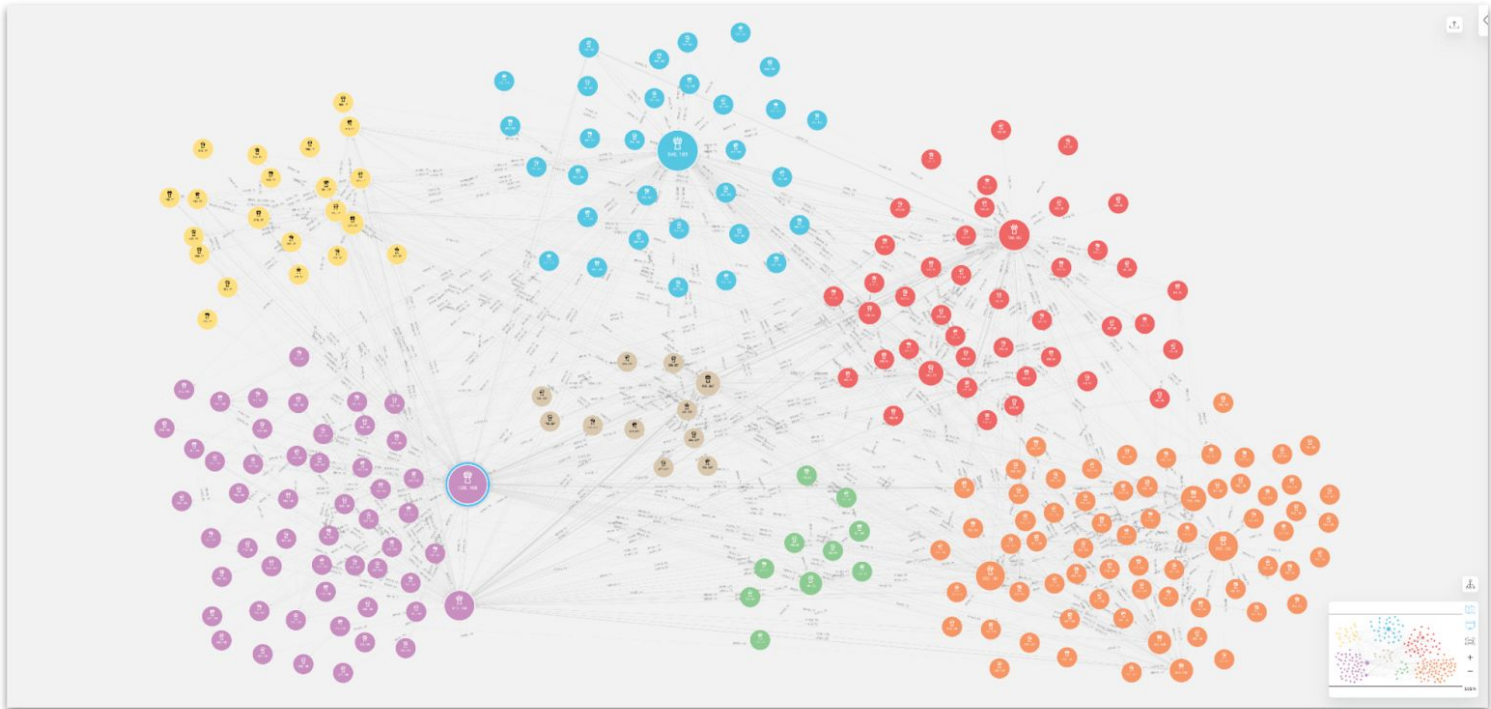
How likely are these groups to strengthen or break apart?

Community Detection

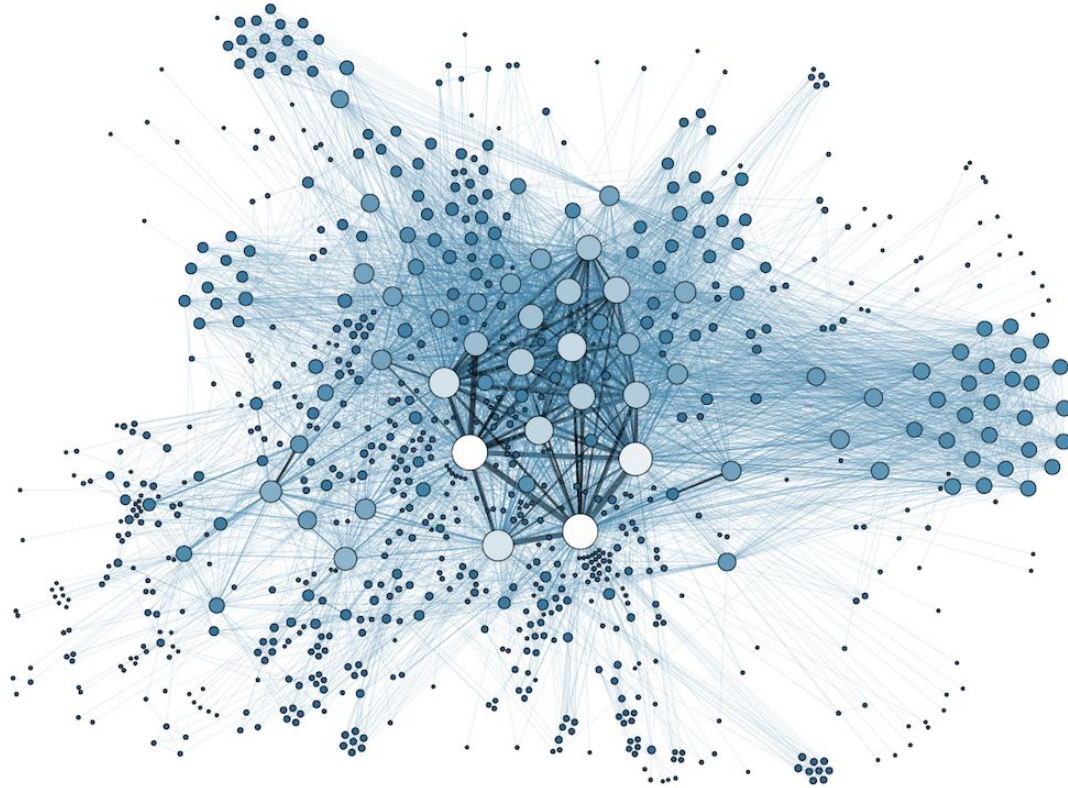


Graph Model

Community Detection



Community Detection



Modularity Optimization

Modularity Optimization

identifies communities based on modularity

A community has high modularity if it is unusually well-connected within itself, and unusually weakly connected to nodes outside the community

The algorithm will explore for every node if its modularity score might increase if it changes its community to one of its neighboring nodes.

Louvain Algorithm

Louvain identifies communities
based on modularity

*A community has high modularity if it is unusually well-connected within itself,
and unusually weakly connected to nodes outside the community*

*The algorithm will explore for every node if its modularity score might increase if it
changes its community to one of its neighboring nodes.*

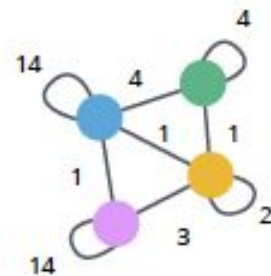
Louvain Algorithm: Pass 1



Step 0

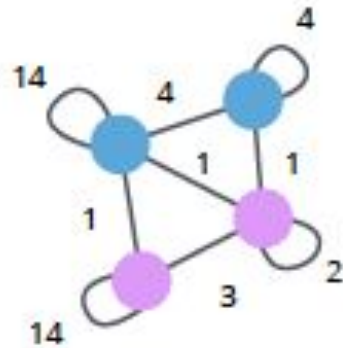


Step 1



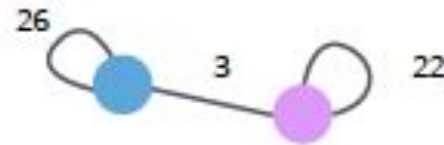
Step 2

Louvain Algorithm: Pass 2



Step 1

Pass 2



Step 2

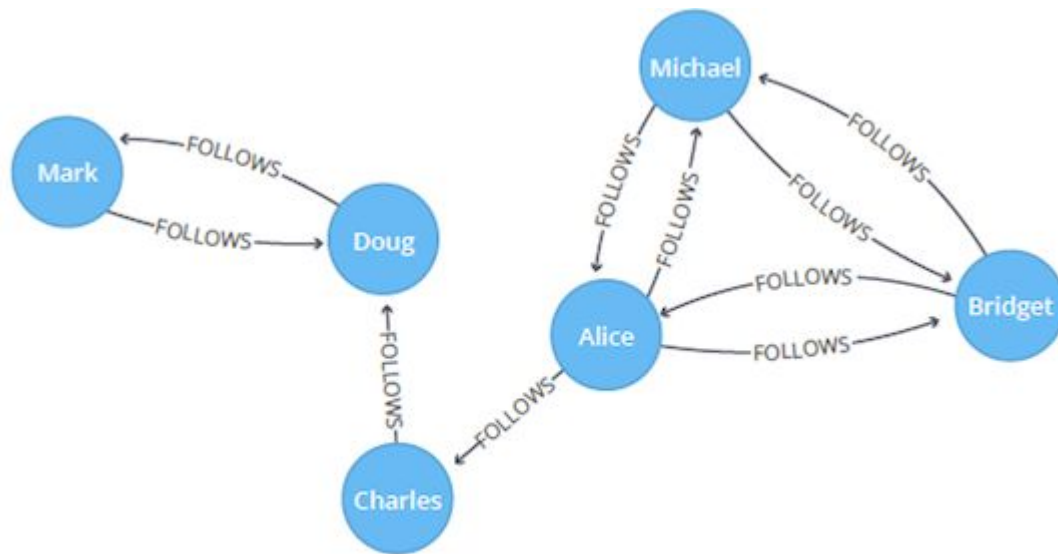
Note:: **Leiden** is similar but work with unconnected graphs. Faster

Label Propagation

Label Propagation identifies communities
by passing message to neighbors

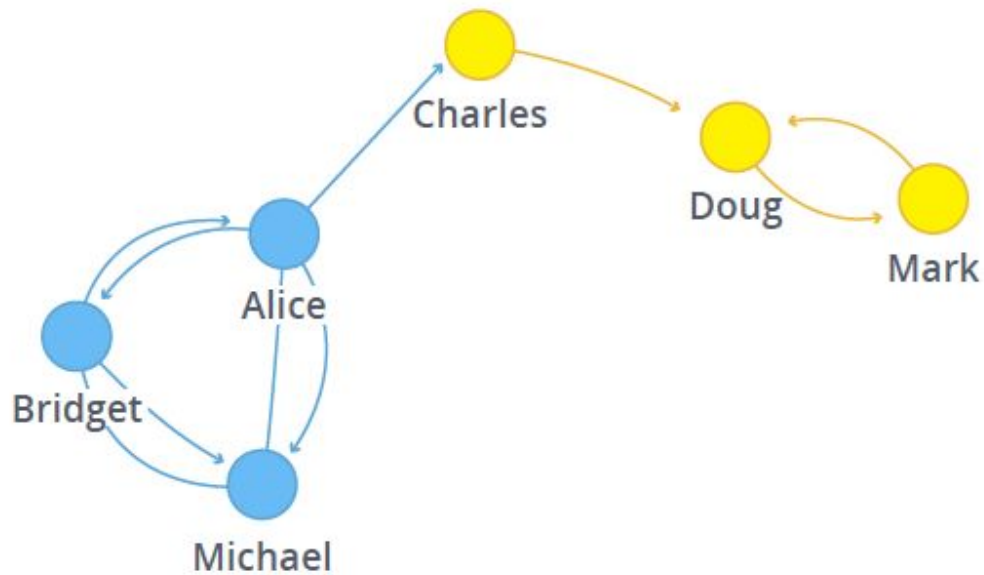
*Nodes acquire the label from the majority of their neighbors
Runs quickly - scales well*

Label Propagation



Graph Model

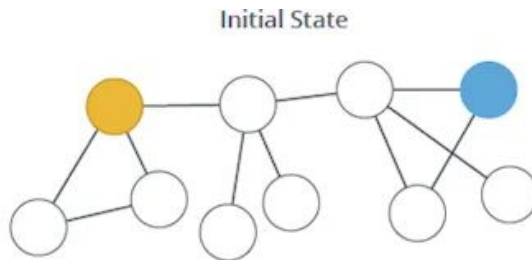
Label Propagation



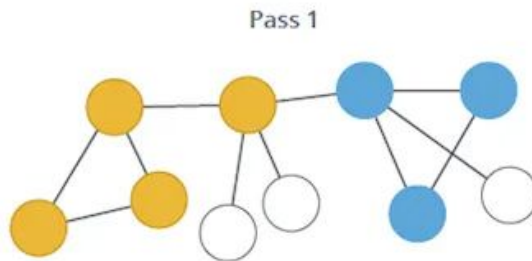
Visualization of Label Propagation

Label Propagation

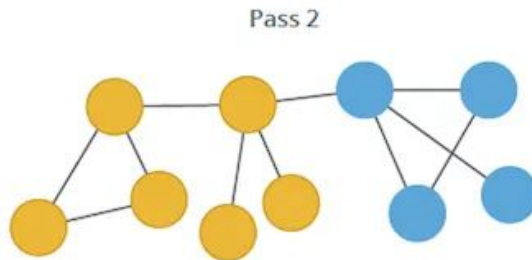
- Fast
- Iterative
- Identifies Density



Some nodes have labels



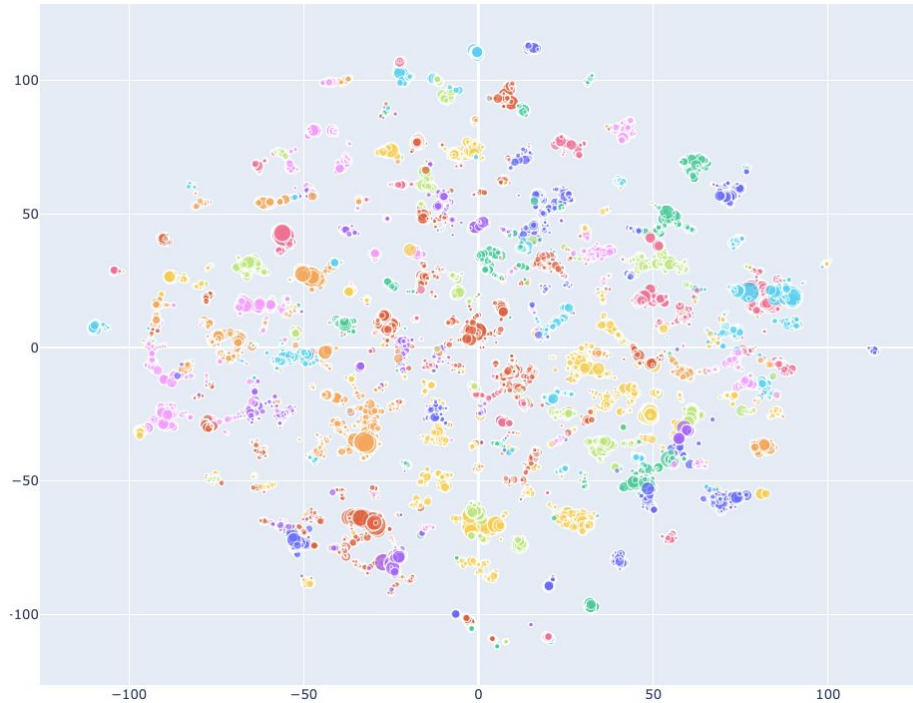
More labels added



Iterations continue until there is convergence on a solution, a set solution range, or a set number of iterations.

Label Propagation Algorithm

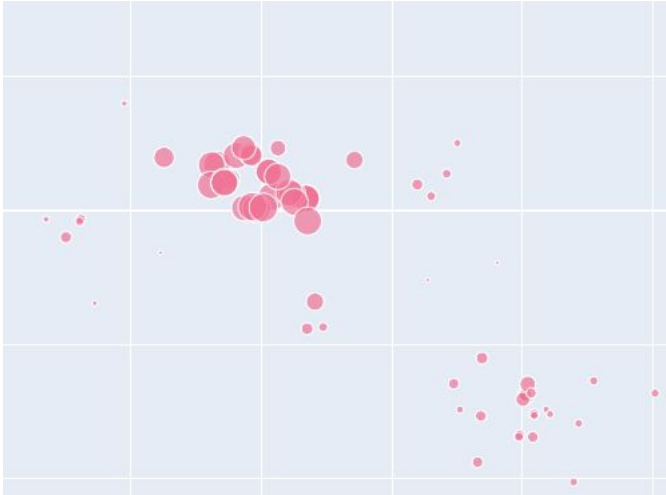
Analyzing Context with Graphs and GDS



2D Visualization of Context Document Node Embeddings

Embedding Visualization Detail

Our grounding document embeddings are generally distinct, but some communities overlap, which may warrant further analysis.



*Single-Community
Document Cluster*



*Cluster of Overlapping
Document Communities*



Data Quality

Curation for AI Success

Elements of High-Quality Grounding Data

Relevant

Related to the problem the LLM is solving and the questions you expect users to ask.

Augmenting

Fills known gaps in the LLM's 'knowledge', due to data being non-public or outside the training window.

Reliable

Contains accurate information, whether from inside or outside of the organization.

Clean

Is generally free of errors or noise, especially if generated from notebooks, websites, repos, etc...

Efficient

Does not contain duplicates, or near-duplicate, 'chunks' that take up valuable context limits.

Identifying Text Errors

Combining graph and traditional statistics helps us identify outliers or data quality issues.

- Traditional: Text length, word count, and word length
- Graph: Community, community size, and PageRank score

	size	med_textLen	med_wordCount	med_avgWordLen	med_pageRank
community					
14015	44	372.0	35.0	8.38	2.236358
755	30	507.0	78.0	5.50	1.893640
7117	51	512.0	1.0	512.00	1.811893
4506	25	479.0	46.0	8.16	1.677685
8299	22	422.5	68.0	5.22	1.603973
12142	27	465.0	61.0	6.71	1.498172
4035	43	407.0	70.0	4.83	1.495224
4701	51	373.0	50.0	6.88	1.468466
1455	22	422.0	56.0	6.77	1.466139
10877	37	421.0	40.0	7.37	1.397775

Graph Communities with additional statistics;
Note: Outlier average word length in community 7117

Investigating Text Chunk Outliers

text text_len word_count avg_word_len community pageRank

```

{"payload":{"allShortcutsEnabled":false,"fileTree":{"":{"items":
  [{"name":"algorithms","path":"algorithms","contentType":"directory"},
  {"name":"embeddings","path":"embeddings","contentType":"directory"},
  {"name":"README.md","path":"README.md","contentType":"file"},
  {"name":"gds-resources.md","path":"gds-
    resources.md","contentType":"file"},{"name":"graph-data-
    modeling.md","path":"graph-data-modeling.md","contentType":"file"},
  {"name":"graph-eda.md","path":"graph-eda.md","contentType":"file"},
  {"name":"graphs-llms.

```

	512	1	512.0	7117	2.415697
--	-----	---	-------	------	----------

```

{"payload":{"allShortcutsEnabled":false,"fileTree":{"":{"items":
  [{"name":"algorithms","path":"algorithms","contentType":"directory"},
  {"name":"embeddings","path":"embeddings","contentType":"directory"},
  {"name":"README.md","path":"README.md","contentType":"file"},
  {"name":"gds-resources.md","path":"gds-
    resources.md","contentType":"file"},{"name":"graph-data-
    modeling.md","path":"graph-data-modeling.md","contentType":"file"},
  {"name":"graph-eda.md","path":"graph-eda.md","contentType":"file"},
  {"name":"graphs-llms.

```

	512	1	512.0	7117	2.413580
--	-----	---	-------	------	----------

```

{"payload":{"allShortcutsEnabled":false,"fileTree":{"":{"items":
  [{"name":"algorithms","path":"algorithms","contentType":"directory"},
  {"name":"embeddings","path":"embeddings","contentType":"directory"},
  {"name":"README.md","path":"README.md","contentType":"file"},
  {"name":"gds-resources.md","path":"gds-
    resources.md","contentType":"file"},{"name":"graph-data-
    modeling.md","path":"graph-data-modeling.md","contentType":"file"},
  {"name":"graph-eda.md","path":"graph-eda.md","contentType":"file"},
  {"name":"graphs-llms.

```

	512	1	512.0	7117	2.288317
--	-----	---	-------	------	----------

Highly-Similar Text Chunks

Because we persisted the KNN Similarity relationships, we can query them just like any other object in the graph:

- Identify all text chunks with at least one 99%+ similarity relationship
- Identify communities with the highest average similarity scores

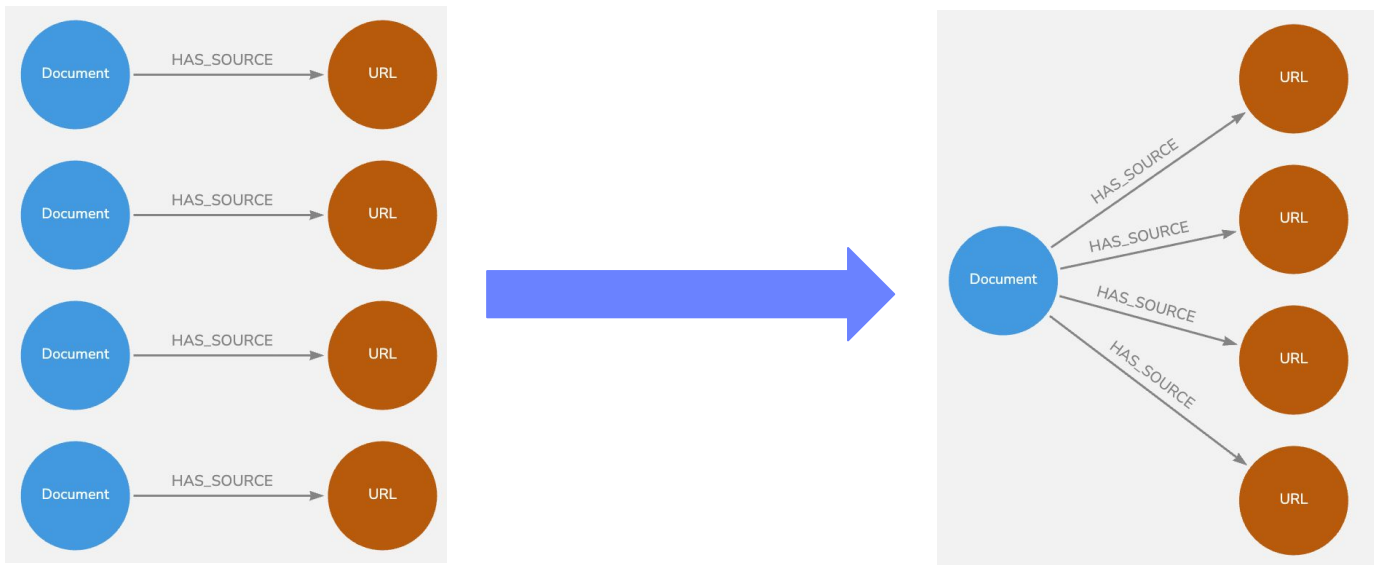
community	average_similarity	node_count	relationship_count
4213	0.991812	17	272
4702	0.986700	49	1207
11148	0.986366	17	272
6979	0.985077	71	1553
11180	0.982547	17	272

Highly-Similar Text Chunks

url	text	community
https://neo4j.com/docs/graph-data-science/current/machine-learning/node-embeddings/graph-sage/	Heterogeneous relationships Heterogeneous relationships fully supported. The algorithm has the ability to distinguish between relationships of different types. Heterogeneous relationships Heterogeneous relationships allowed. The algorithm treats all selected relationships similarly regardless of their type. Weighted relationships Weighted trait. The algorithm supports a relationship property to be used as weight, specified via the relationshipWeightProperty configuration parameter.	4702
https://neo4j.com/docs/graph-data-science/current/algorithms/harmonic-centrality/	Heterogeneous relationships Heterogeneous relationships fully supported. The algorithm has the ability to distinguish between relationships of different types. Heterogeneous relationships Heterogeneous relationships allowed. The algorithm treats all selected relationships similarly regardless of their type. Weighted relationships Weighted trait. The algorithm supports a relationship property to be used as weight, specified via the relationshipWeightProperty configuration parameter.	4702
https://neo4j.com/docs/graph-data-science/current/algorithms/bellman-ford-single-source/	Heterogeneous relationships Heterogeneous relationships fully supported. The algorithm has the ability to distinguish between relationships of different types. Heterogeneous relationships Heterogeneous relationships allowed. The algorithm treats all selected relationships similarly regardless of their type. Weighted relationships Weighted trait. The algorithm supports a relationship property to be used as weight, specified via the relationshipWeightProperty configuration parameter.	4702
https://neo4j.com/docs/graph-data-science/current/algorithms/modularity-optimization/	Heterogeneous relationships Heterogeneous relationships fully supported. The algorithm has the ability to distinguish between relationships of different types. Heterogeneous relationships Heterogeneous relationships allowed. The algorithm treats all selected relationships similarly regardless of their type. Weighted relationships Weighted trait. The algorithm supports a relationship property to be used as weight, specified via the relationshipWeightProperty configuration parameter.	4702
https://neo4j.com/docs/graph-data-science/current/algorithms/closeness-centrality/	Heterogeneous relationships Heterogeneous relationships fully supported. The algorithm has the ability to distinguish between relationships of different types. Heterogeneous relationships Heterogeneous relationships allowed. The algorithm treats all selected relationships similarly regardless of their type. Weighted relationships Weighted trait. The algorithm supports a relationship property to be used as weight, specified via the relationshipWeightProperty configuration parameter.	4702

Collapsing Duplicate Nodes

We can use `apoc.nodes.collapse()` to combine duplicate nodes into a single node with all prior relationships pointing to the single node.





Application Understanding

What your application can teach you.

Graphs Enable Explainable AI with LLMs

- Production
 - How the LLM use grounding documents
 - How they produce answers will become more and more important
- Knowledge Graphs and GDS enable Explainable AI by:
 - Logging user interactions in the same database as the context
 - Visualizing conversations with context
 - Providing tools to analyze LLM performance and identify opportunities for improvement

KNOWLEDGE GRAPH

DOMAIN GRAPH



LEXICAL GRAPH

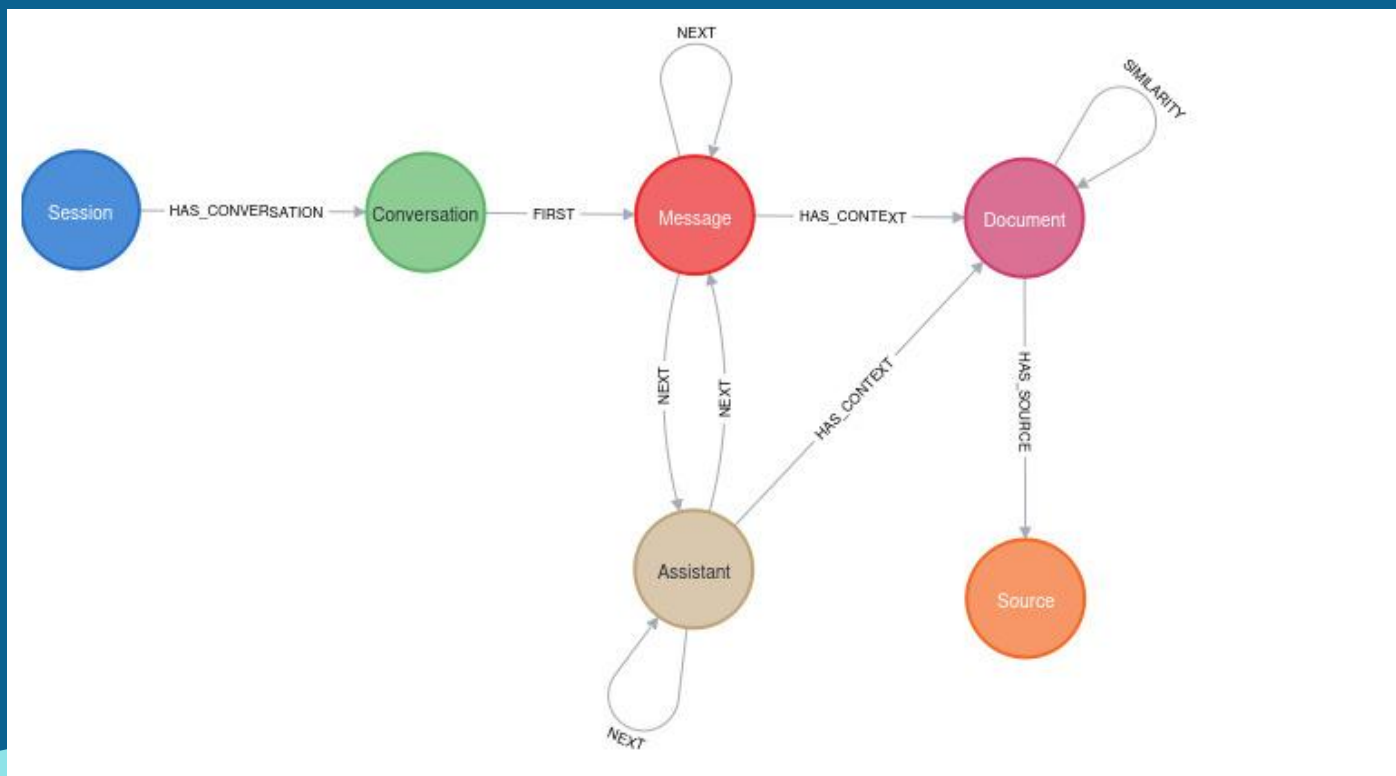


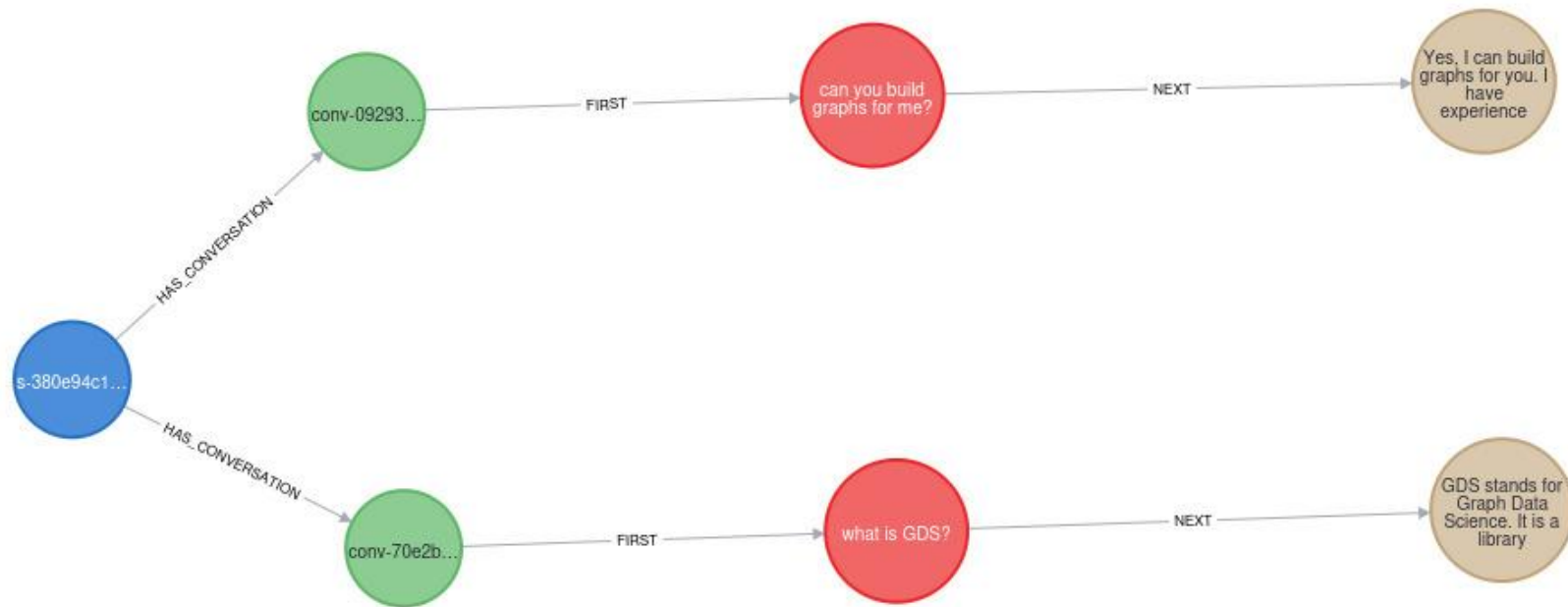
MEMORY GRAPH



Source Legend

- = Structured Stores
- = Unstructured Data
- = Application





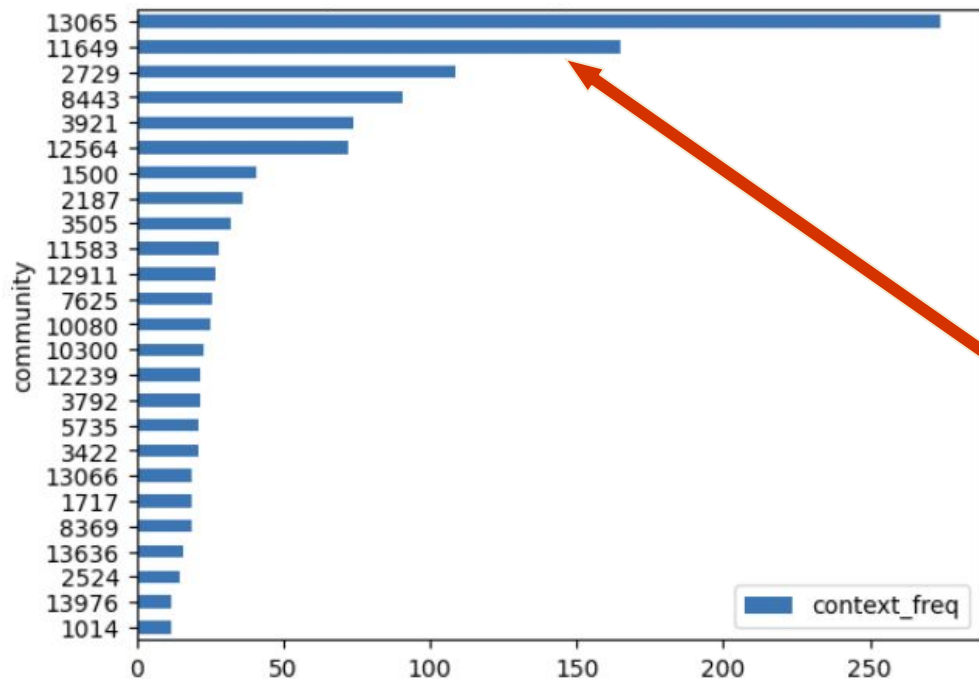
Most Frequently Used Grounding Text

context_count

text

10.0	GDS Degree Centrality algorithm is useful for creating statistics that can support calculating ratios and identifying outliers. The same could also be performed using Cypher (and, if a large graph, apoc.periodic.iterate()). However, one of the benefits of using GDS and Graph Projections is that we can create a single projection and run multiple algorithms on it.
9.0	For example, if we were analyzing a financial transaction network we may want to identify customers who had the most transactions. We
9.0	Neo4j Data Connectors Apache Kafka, Apache Spark, and BI tools Cypher Query Language Powerful, intuitive, and graph-optimized Solutions Use Cases Fraud detection, knowledge graphs and more Generative AI Back your LLMs with a knowledge graph for better business AI Learn More
9.0	want you to act as an experienced graph data scientist who works at Neo4j. A customer asks you how large language models (LLMs) like ChatGPT can assist with graph data science, specifically using Neo4j Graph Data Science algorithms. How would you advise this customer to explore integrating LLMs into their graph data science workflows? What would likely be the easiest or most impactful ways in which an LLM can make them more productive and effective?
9.0	Perhaps you are a data scientist, or you aspire to become one. Graph analytics and data science offer a wide variety of algorithms that can enhance your analytical toolbox and help you find meaningful insights into highly-connected datasets. In this section, I will show how easily you can integrate graph algorithms into your analytical workflows. Neo4j offers a Python client for Neo4j Graph Data Science library that seamlessly allows you to execute graph algorithms using only Python code.

Most Frequent Document Communities



- Combining Document usage frequency with previously identified Document Communities, we can see which of these Communities are the most frequent sources of Context Documents
- The second most frequent Community (11649) comprises text chunks from blogs by Tomaz Bratanic

Modularity Optimization

Modularity Optimization

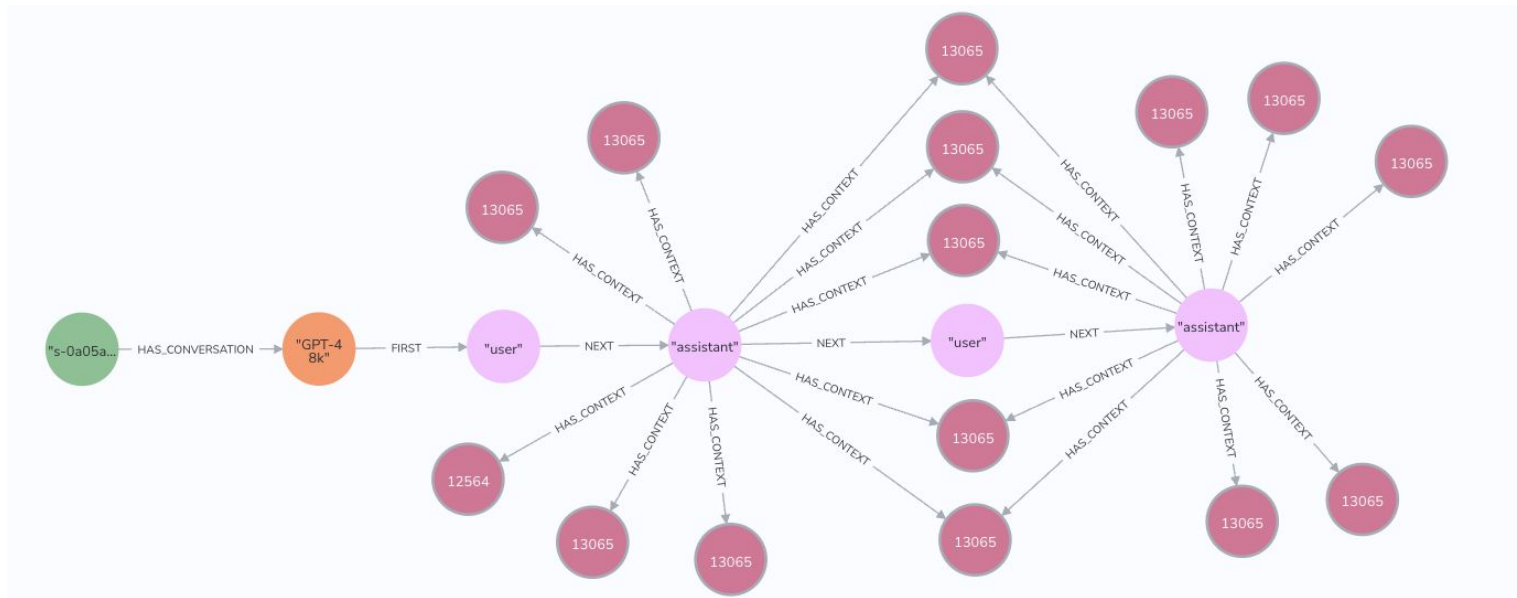
identifies communities based on modularity

A community has high modularity if it is unusually well-connected within itself, and unusually weakly connected to nodes outside the community

The algorithm will explore for every node if its modularity score might increase if it changes its community to one of its neighboring nodes.

Logging and Visualizing Conversations

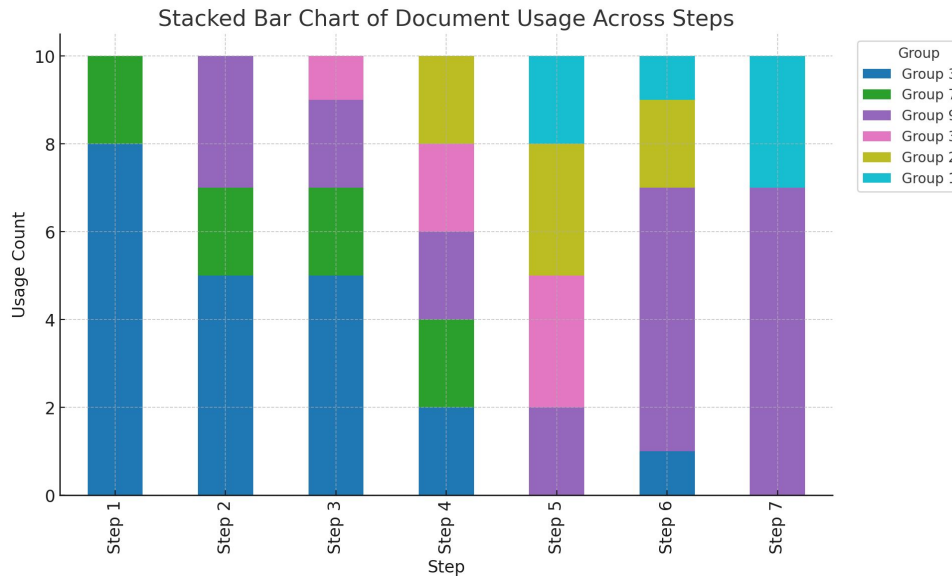
Graphs enable logging of LLM conversations in the same database as the context documents and with defined relationships.



*Graph of an actual conversation between an Agent Neo user and the ChatGPT-4 LLM.
Context Documents are labeled with their GDS Community.*

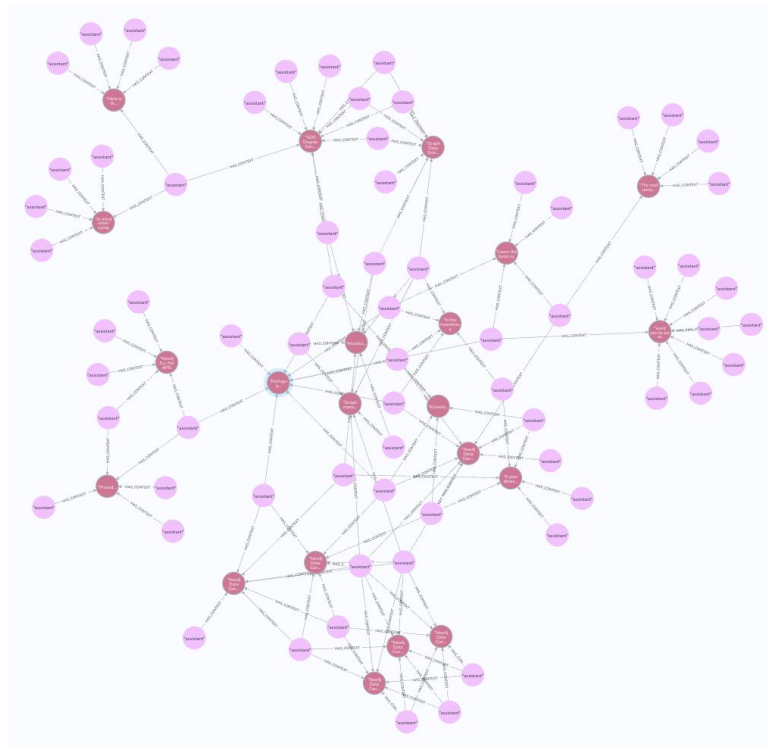
Visualizing Document Usage

- The heat map depicts how frequently documents are used during a single conversation:
 - X-axis represents LLM messages
 - Y-axis represents individual documents
 - Color depicts document use frequency count (1x to 3x)
- Documents are re-used throughout conversations



Visualizing Context Document Usage

- Graphs enable us to visualize the most frequently used context Documents along with the associated LLM responses
- Natural clusters form in the graph even among the most frequently used Documents



*LLM Responses (pink) and
Most Frequently Used Context Documents (red)*

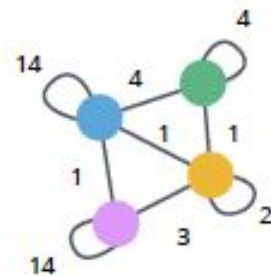
Louvain Algorithm: Pass 1



Step 0

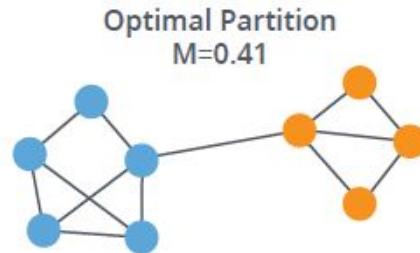
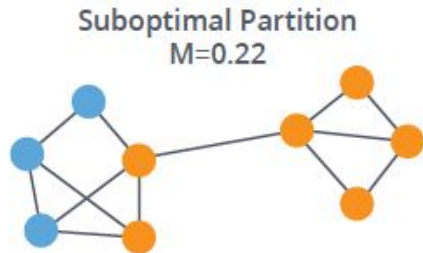
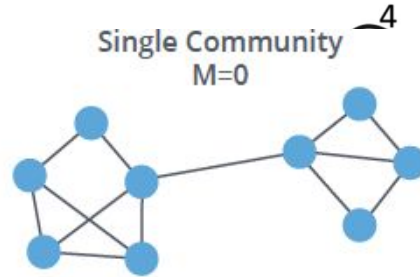
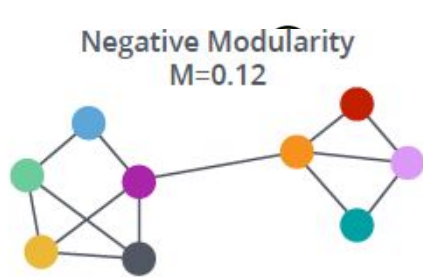


Step 1



Step 2

Modularity



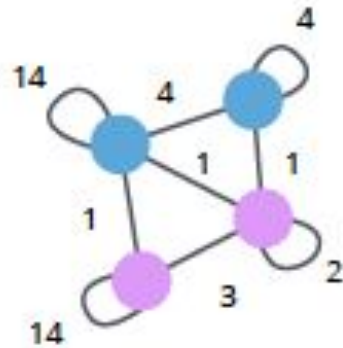
Modularity

Louvain Algorithm

Louvain identifies communities
based on modularity

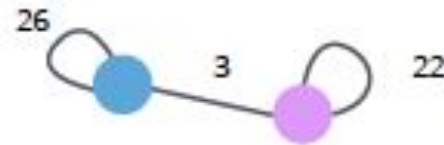
*A community has high modularity if it is unusually well-connected within itself,
and unusually weakly connected to nodes outside the community*

Louvain Algorithm: Pass 2



Step 1

Pass 2



Step 2

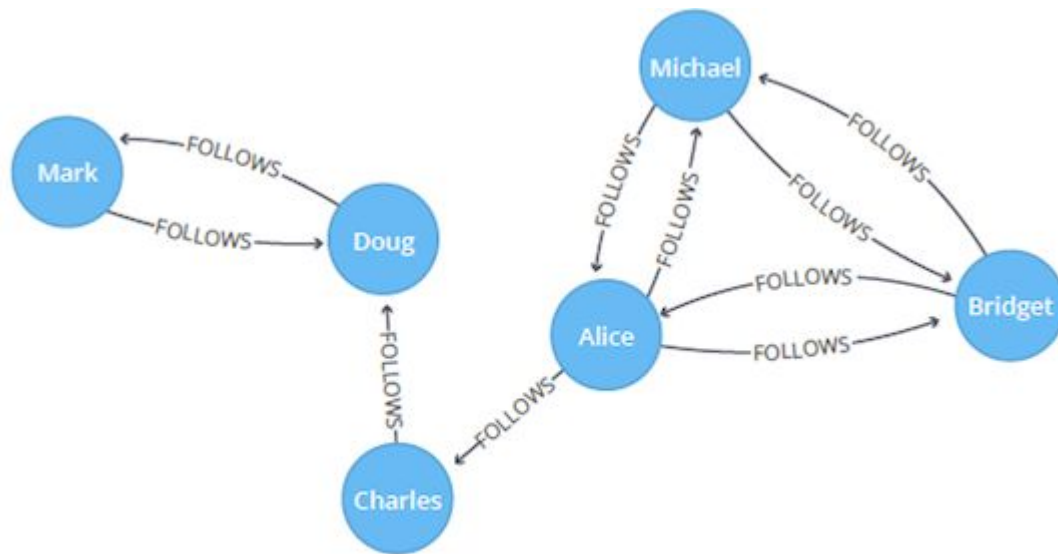
Note:: **Leiden** is similar but work with unconnected graphs. Faster

Label Propagation

Label Propagation identifies communities
by passing message to neighbors

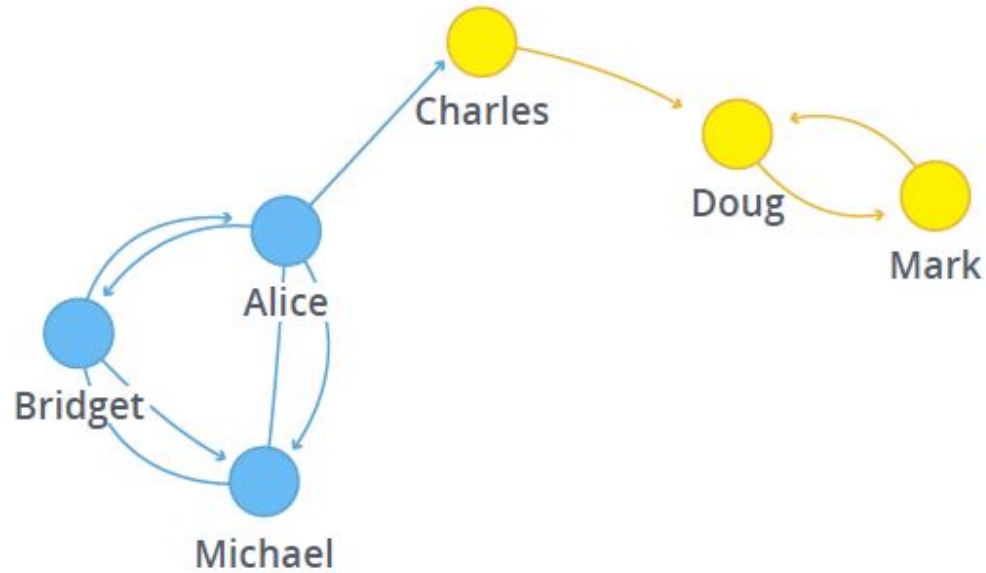
*Nodes acquire the label from the majority of their neighbors
Runs quickly - scales well*

Label Propagation



Graph Model

Label Propagation

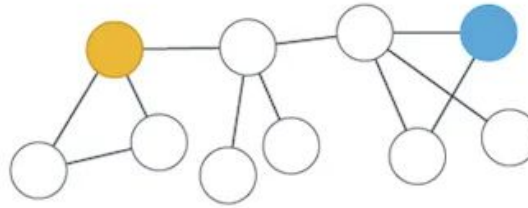


Visualization of Label Propagation

Label Propagation

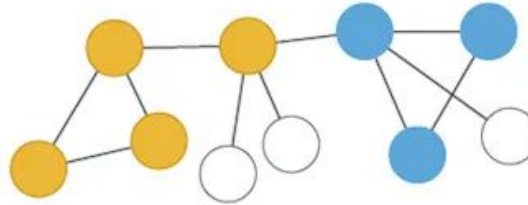
- Fast
- Iterative
- Identifies Density

Initial State



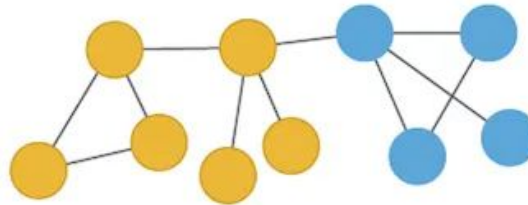
Some nodes have labels

Pass 1



More labels added

Pass 2



Iterations continue until there is convergence on a solution, a set solution range, or a set number of iterations.

Label Propagation Algorithm

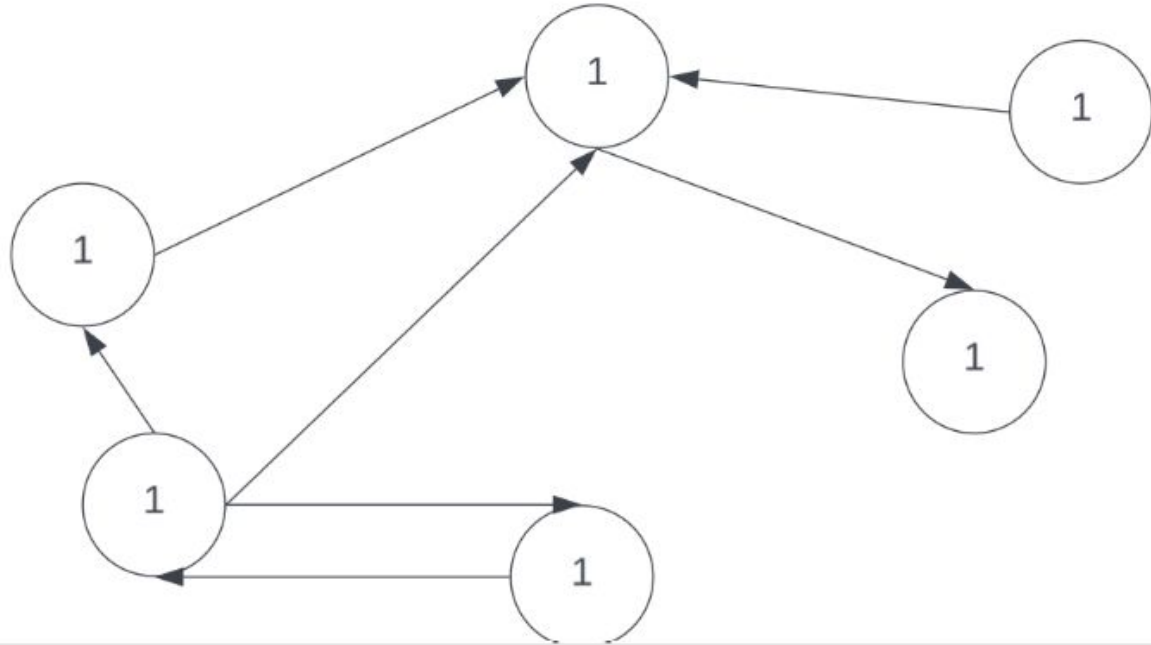
Page Rank

Page Rank

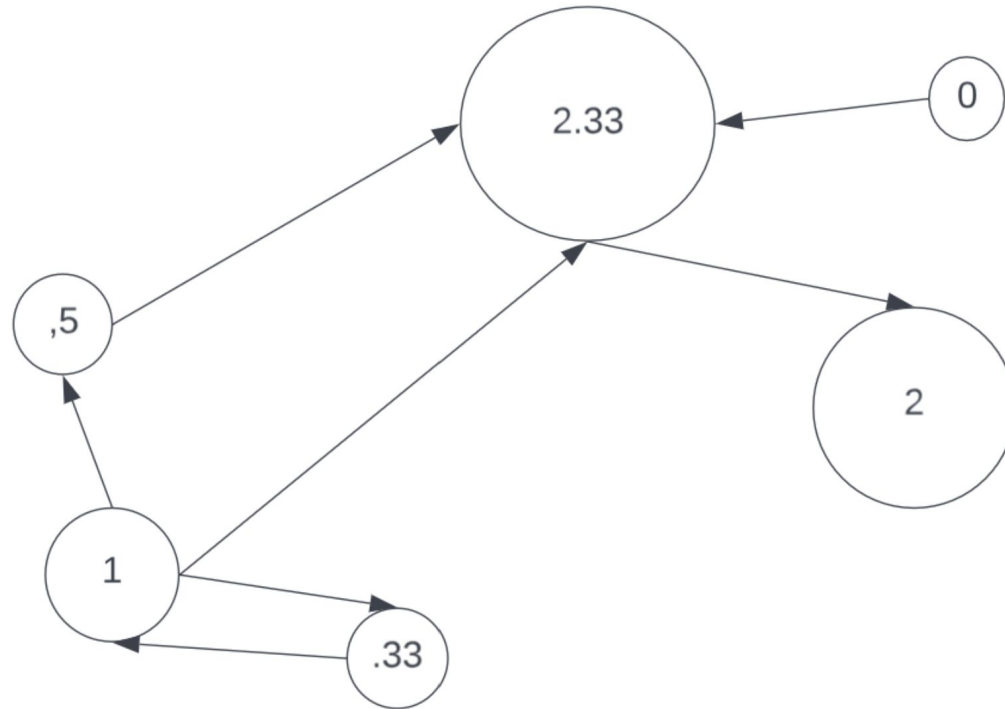
Node is important if many nodes link to it
and/or
important nodes link to it

Nodes share influence with their connections

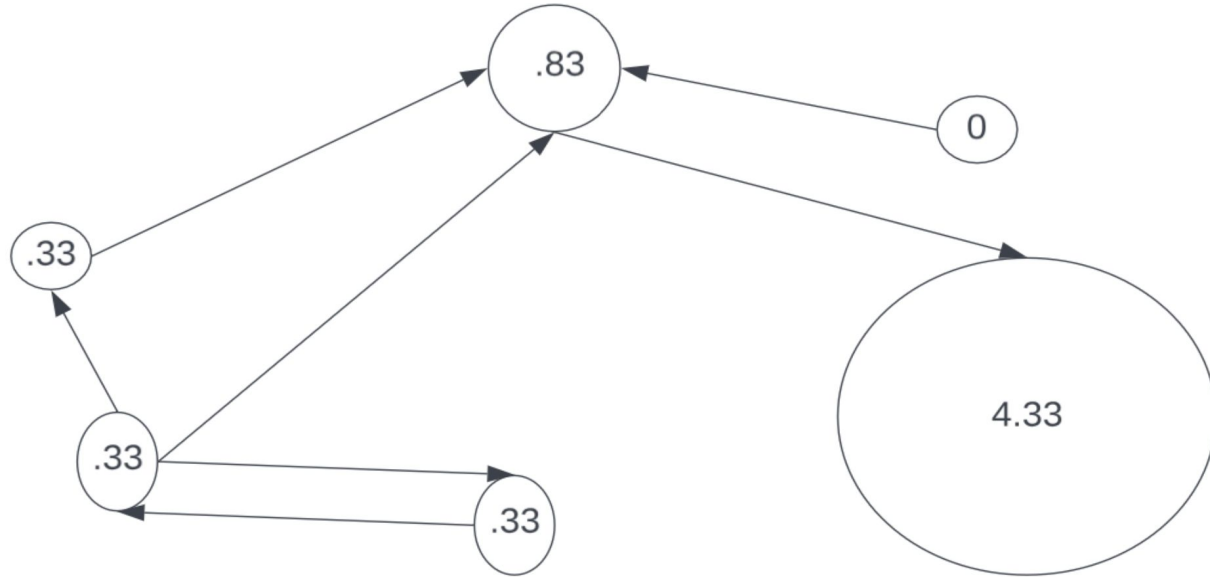
Page Rank



Page Rank



Page Rank



Page Rank

Page Rank

Node is important if many nodes link to it
and/or
important nodes link to it

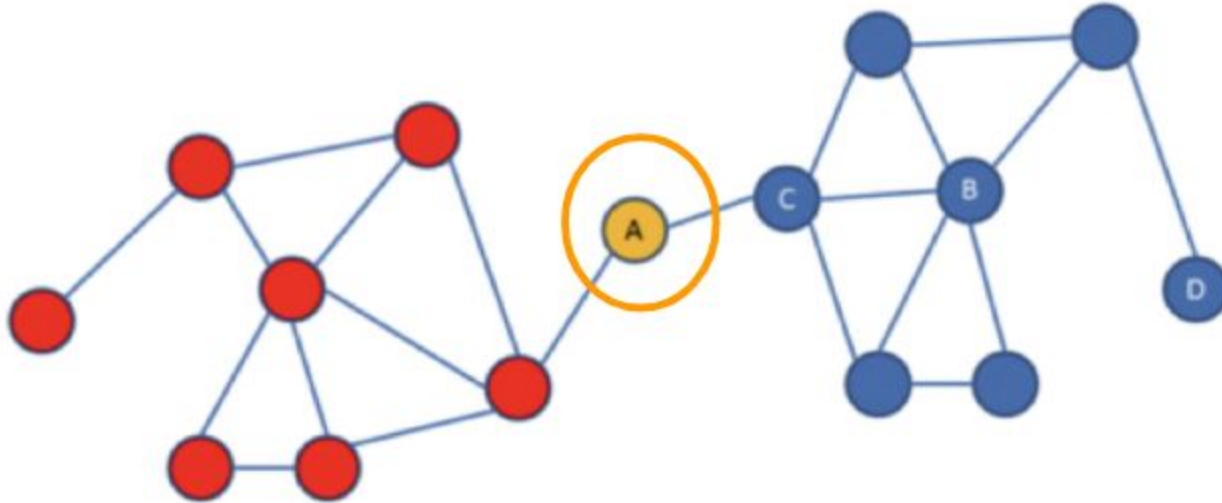
Nodes share influence with their connections

Betweenness Centrality

Betweenness Centrality

Node is important if
a large number of shortest paths
pass through it

Betweenness Centrality



modularity

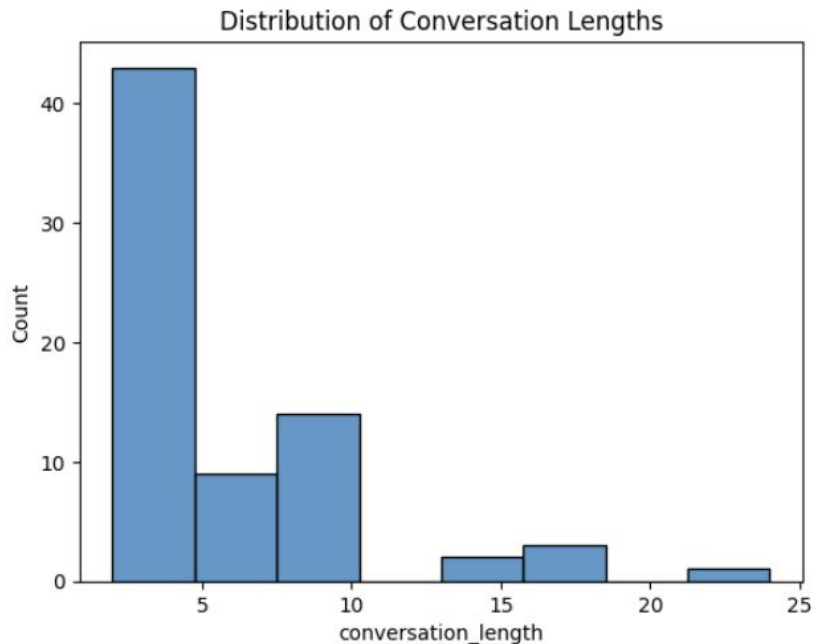
Betweenness Centrality

Methodology

- Calculates the shortest (weighted) path between every pair of nodes in a connected graph
- Uses a the breadth-first search algorithm
- Each node receives a score, based on the number of these shortest paths that pass through the node.

Conversation Lengths

- Because conversations are logged as graphs it is easy to measure the length and store it as a new property on each Conversation node.
- Most users have been experimenting with our tool, so we expect conversation lengths to be shorter.
- As users are more comfortable with these applications, we expect conversation lengths to increase.



Inspect Communities of LLM Responses

We can use similar approaches to inspect each of the LLM response communities via traditional and GDS-based statistics.

	size	med_numDocs	med_pageRank	med_textLen	med_wordCount	ratings_Good	ratings_Bad	NotRated
community								
14725	25	6.0	1.009269	1911.0	246.0	15	2	8
14501	16	5.0	0.950335	2001.0	246.5	4	3	9
14699	14	10.0	0.862450	1778.0	219.5	6	2	6
14818	13	10.0	0.980136	1963.0	244.0	6	1	6
14685	11	10.0	0.800114	2089.0	246.0	5	4	2

Graph enabled Explainable AI



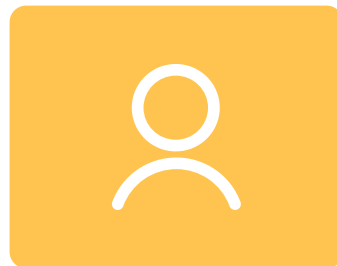
Document
Usage

How the LLM is
using the
various
documents in
your dataset



Application/
Agent Logic

You can track
graph traversals
to explain chain
of actions



User Experience

Visualizing
conversations
with context.



Applications

Analyze
performance
and identify
opportunities for
improvement

GraphRAG Resources



Github repository with
the Intro Workshop



Developer guides and
coded examples

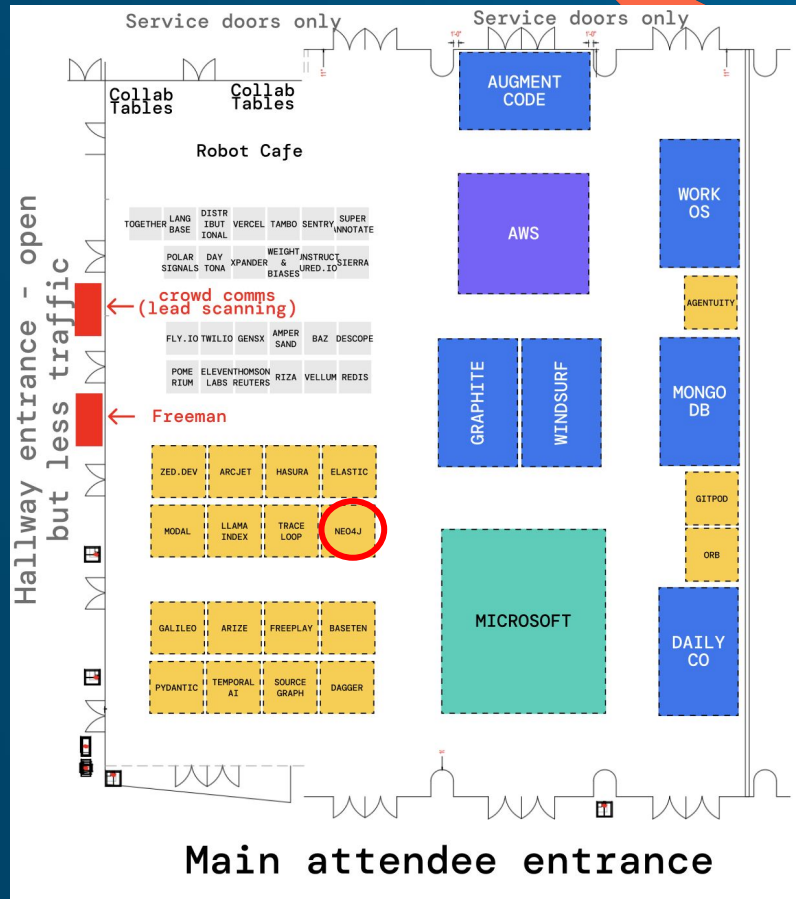


Meetup Today (06/03) @5: AI Agent SF #3
- Agent Protocols (Graphs & MCP) w/
Michael Hunger

Swing By Our Booth!

Booth # G9

- Neo4j MCP Servers
- Google ADK Examples
- Knowledge Graph Construction
- and more!



An aerial photograph of a pod of dolphins swimming in the ocean. The dolphins are dark grey and are scattered across the frame, mostly in the upper right quadrant. The water is a deep blue with visible ripples and waves. In the top left corner, there is a light blue abstract shape. In the bottom left corner, there is a darker blue abstract shape. On the right side, there is a large, solid orange shape that curves along the edge of the image.

Thank you!