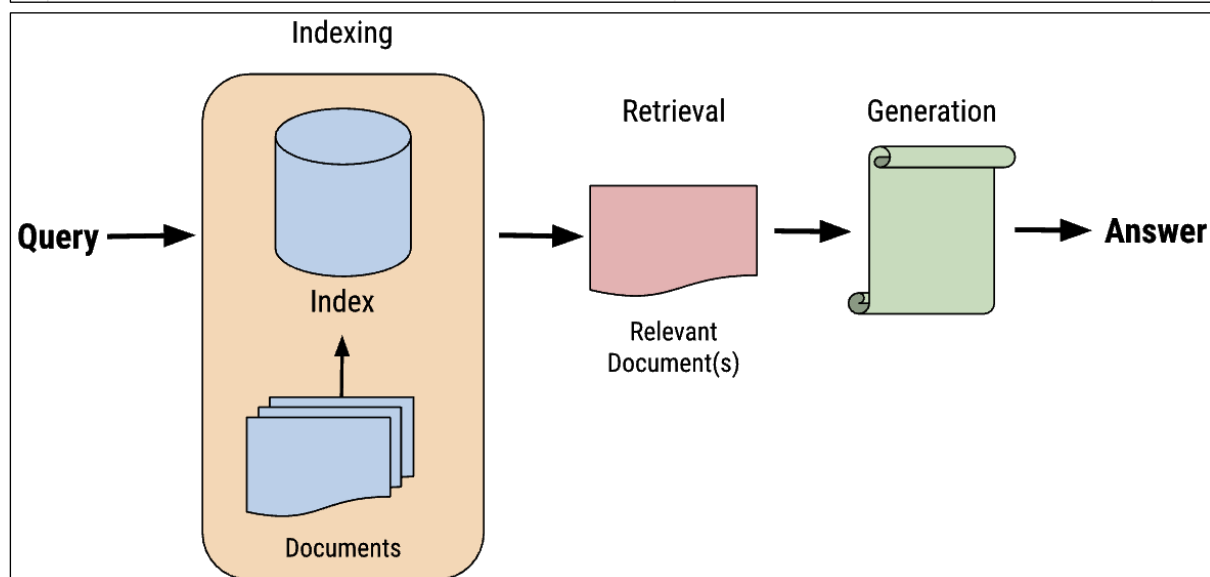
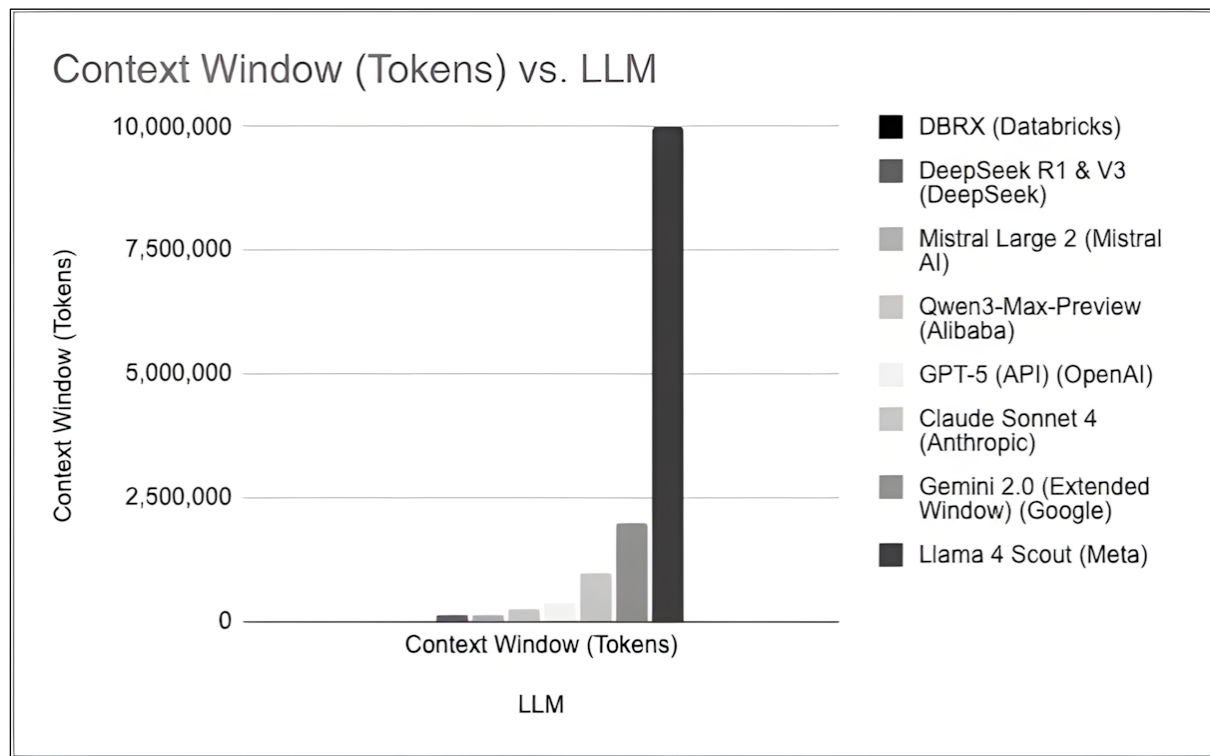


Chapter 1: What Is Retrieval-Augmented Generation?



Chapter 2: Code Lab: An Entire RAG Pipeline

Introduction to Retrieval Augmented Generation (RAG)

Date: March 10, 2024 | Estimated Reading Time: 15 min | Author: Keith Bourne

In the rapidly evolving field of artificial intelligence, Retrieval-Augmented Generation (RAG) is emerging as a significant addition to the Generative AI toolkit. RAG harnesses the strengths of Large Language Models (LLMs) and integrates them with internal data, offering a method to enhance organizational operations significantly. This book delves into the essential aspects of RAG, examining its role in augmenting the capabilities of LLMs and leveraging internal corporate data for strategic advantage.

As it progresses, the book outlines the potential of RAG in business, suggesting how it can make AI applications smarter, more responsive, and aligned with organizational objectives. RAG is positioned as a key facilitator of customized, efficient, and insightful AI solutions, bridging the gap between Generative AI's potential and specific business needs. This exploration of RAG encourages readers to unlock the full potential of their corporate data, paving the way for an era of AI-driven innovation.

What You Can Expect to Learn

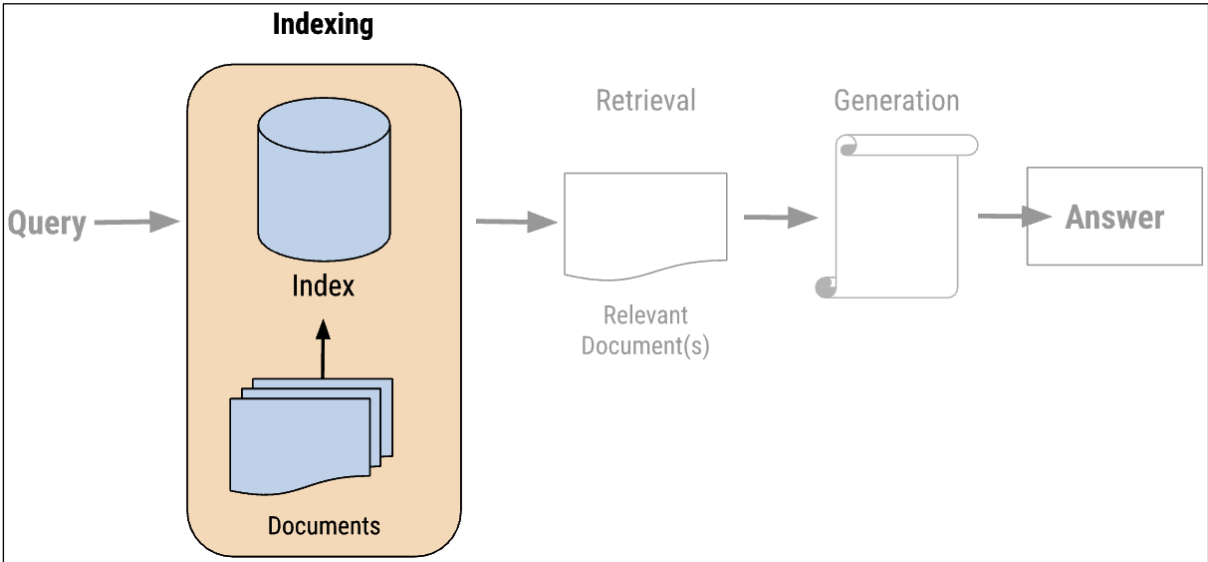
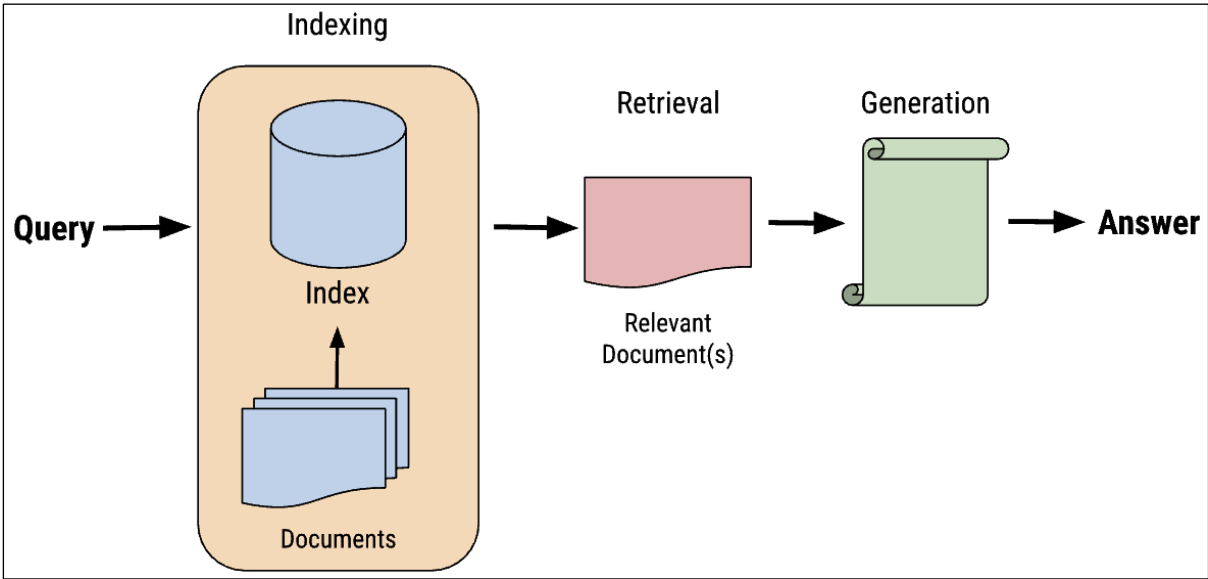
Expect to launch a comprehensive journey to understand and effectively incorporate Retrieval Augmented Generation (RAG) into AI systems. You'll explore a broad spectrum of essential topics, including vector databases, the vectorization process, vector search techniques, prompt engineering and design, and the use of AI agents for RAG applications, alongside methods for evaluating and visualizing RAG outcomes. Through practical, working code examples utilizing the latest tools and technologies like LangChain and Chroma's vector database, you'll gain hands-on experience in implementing RAG in your projects.

At the outset, you'll delve into the core principles of RAG, appreciating its significance in the broader landscape of Generative AI. This foundational knowledge equips you with the perspective needed to discern how RAG applications are designed and why they succeed, paving the way for innovative solution development and problem-solving in AI.

You'll discover the symbiosis between Large Language Models (LLMs) and internal data to bolster organizational operations. By learning about the intricacies of this integration, particularly the process of vectorization, including the creation and management of vector databases for efficient information retrieval, you'll gain crucial skills for navigating and harnessing vast data landscapes effectively in today's data-driven environments.

Gain expertise in vector search techniques, an essential skill set for identifying pertinent data within extensive datasets. Coupled with this, you'll learn strategies for prompt engineering and design, ensuring that you can craft queries that elicit precise and relevant AI responses.

Chapter 4: Components of a RAG System



Introduction to Retrieval Augmented Generation (RAG)

Date: March 10, 2024 | Estimated Reading Time: 15 min | Author: Keith Bourne

In the rapidly evolving field of artificial intelligence, Retrieval-Augmented Generation (RAG) is emerging as a significant addition to the Generative AI toolkit. RAG harnesses the strengths of Large Language Models (LLMs) and integrates them with internal data, offering a method to enhance organizational operations significantly. This book delves into the essential aspects of RAG, examining its role in augmenting the capabilities of LLMs and leveraging internal corporate data for strategic advantage.

As it progresses, the book outlines the potential of RAG in business, suggesting how it can make AI applications smarter, more responsive, and aligned with organizational objectives. RAG is positioned as a key facilitator of customized, efficient, and insightful AI solutions, bridging the gap between Generative AI's potential and specific business needs. This exploration of RAG encourages readers to unlock the full potential of their corporate data, paving the way for an era of AI-driven innovation.

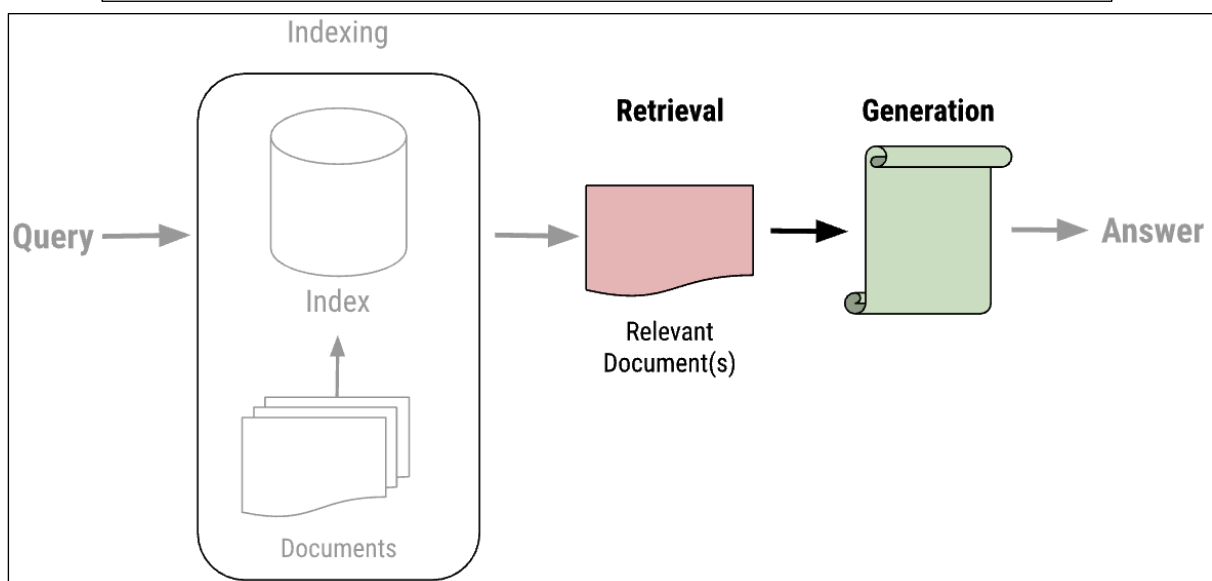
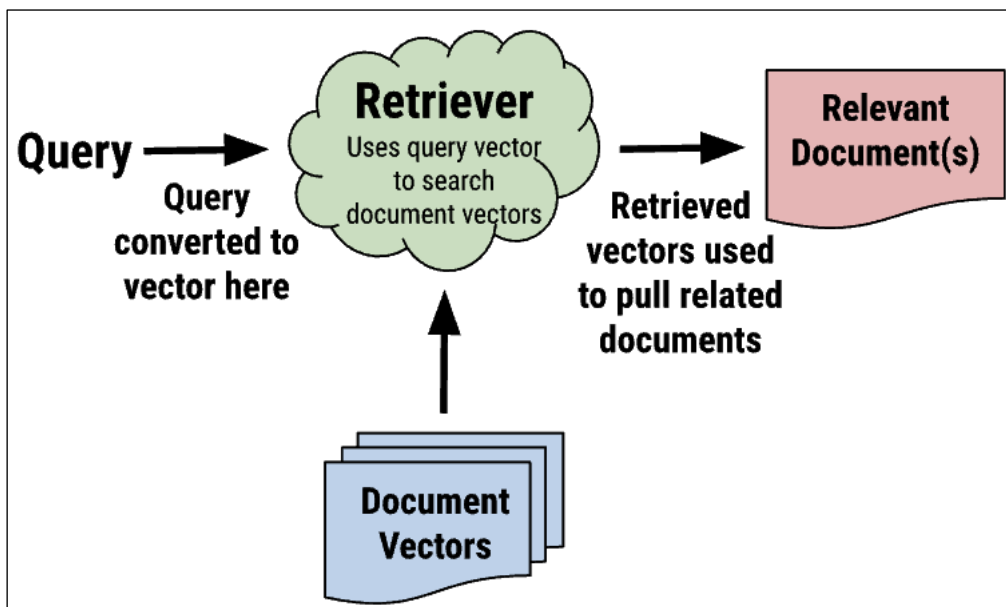
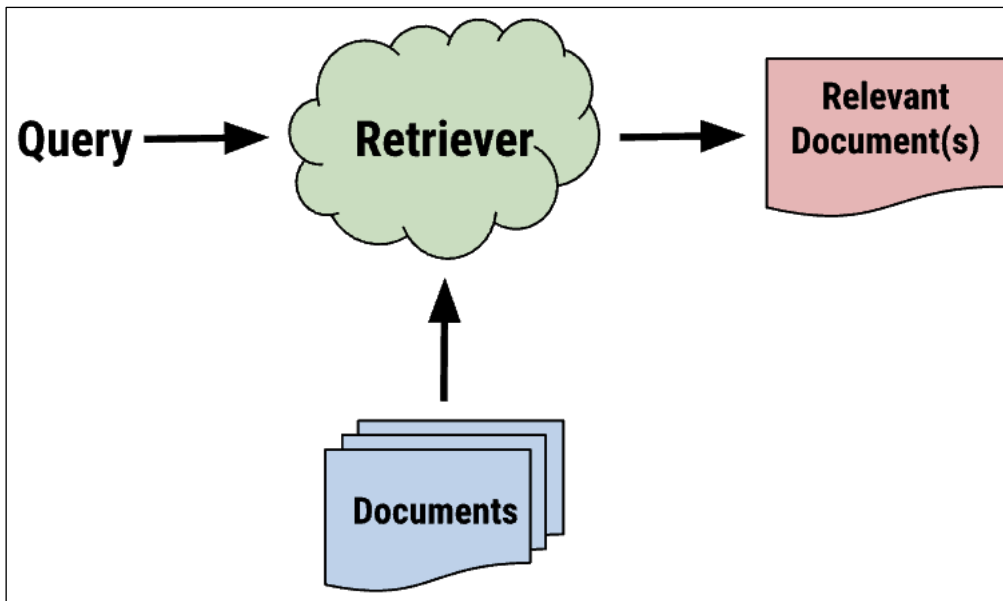
What You Can Expect to Learn

Expect to launch a comprehensive journey to understand and effectively incorporate Retrieval Augmented Generation (RAG) into AI systems. You'll explore a broad spectrum of essential topics, including vector databases, the vectorization process, vector search techniques, prompt engineering and design, and the use of AI agents for RAG applications, alongside methods for evaluating and visualizing RAG outcomes. Through practical, working code examples utilizing the latest tools and technologies like LangChain and Chroma's vector database, you'll gain hands-on experience in implementing RAG in your projects.

At the outset, you'll delve into the core principles of RAG, appreciating its significance in the broader landscape of Generative AI. This foundational knowledge equips you with the perspective needed to discern how RAG applications are designed and why they succeed, paving the way for innovative solution development and problem-solving in AI.

You'll discover the symbiosis between Large Language Models (LLMs) and internal data to bolster organizational operations. By learning about the intricacies of this integration, particularly the process of vectorization, including the creation and management of vector databases for efficient information retrieval, you'll gain crucial skills for navigating and harnessing vast data landscapes effectively in today's data-driven environments.

Gain expertise in vector search techniques, an essential skill set for identifying pertinent data within extensive datasets. Coupled with this, you'll learn strategies for prompt engineering and design, ensuring that you can craft queries that elicit precise and relevant AI responses.





ChatGPT 4 ▾



You

What are the Advantages of using RAG?



ChatGPT

Retrieval-Augmented Generation (RAG) models offer a range of advantages, particularly in the context of natural language processing and generation tasks. These models combine the strengths of pre-trained language models with the vast information available in external data sources. Here are some of the key advantages of using RAG ↓



Can you explain #2 more?



ChatGPT can make mistakes. Consider checking important information.

Chapter 6: Interfacing with RAG and Gradio

RAG Question Answering

Enter a question and get the relevance score, final answer, and sources from RAG.

Enter your question

What are the advantages of using RAG?

ClearSubmit

Relevance Score

Final Answer

Sources

Flag

RAG Question Answering

Enter a question and get the relevance score, final answer, and sources from RAG.

Enter your question

What are the advantages of using RAG?

ClearSubmit

Relevance Score

5

Final Answer

The advantages of using Retrieval-Augmented Generation (RAG) include:

1. **Improved Accuracy and Relevance**: RAG enhances the accuracy and relevance of responses generated by large language models (LLMs) by fetching and incorporating specific, up-to-date information from databases or datasets in real-time. This ensures the output is based on both the model's pre-existing knowledge and the most current and relevant data.

2. **Customization and Flexibility**: RAG allows for responses tailored to domain-specific needs by integrating a company's internal databases into the model's response generation process. This customization is invaluable for creating personalized experiences and for applications requiring a high degree of specificity and detail.

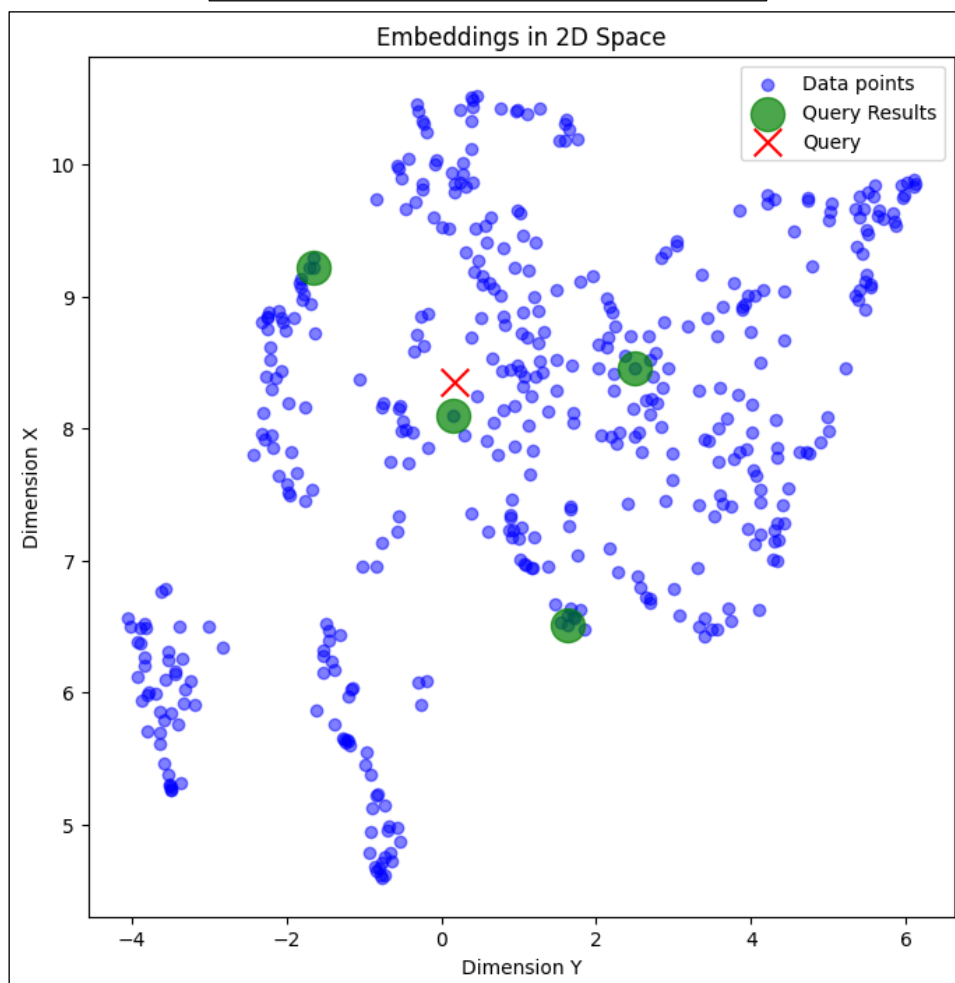
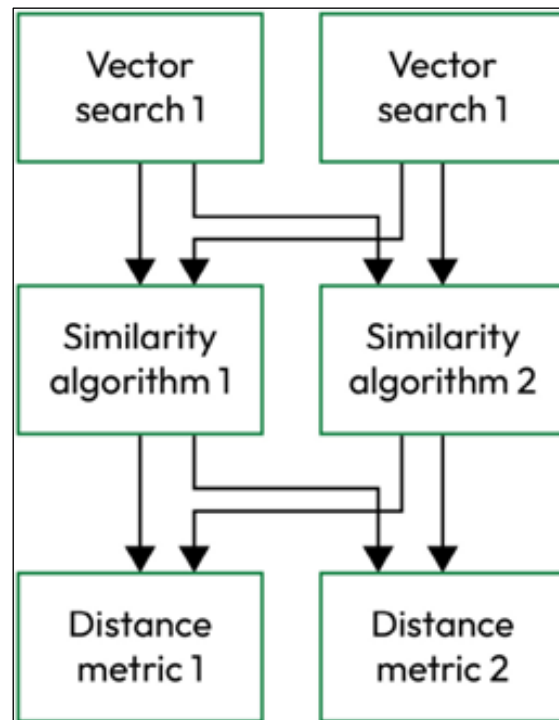
3. **Expanding Model Knowledge Beyond Training Data**: RAG enables models to access and utilize information that was not included in their initial training sets. This expands the knowledge base of the model without needing retraining, making LLMs more versatile and adaptable to new domains or rapidly evolving topics.

These advantages collectively help organizations unlock the full potential of their internal data, making AI applications smarter, more responsive, and better aligned with specific business needs.

Sources

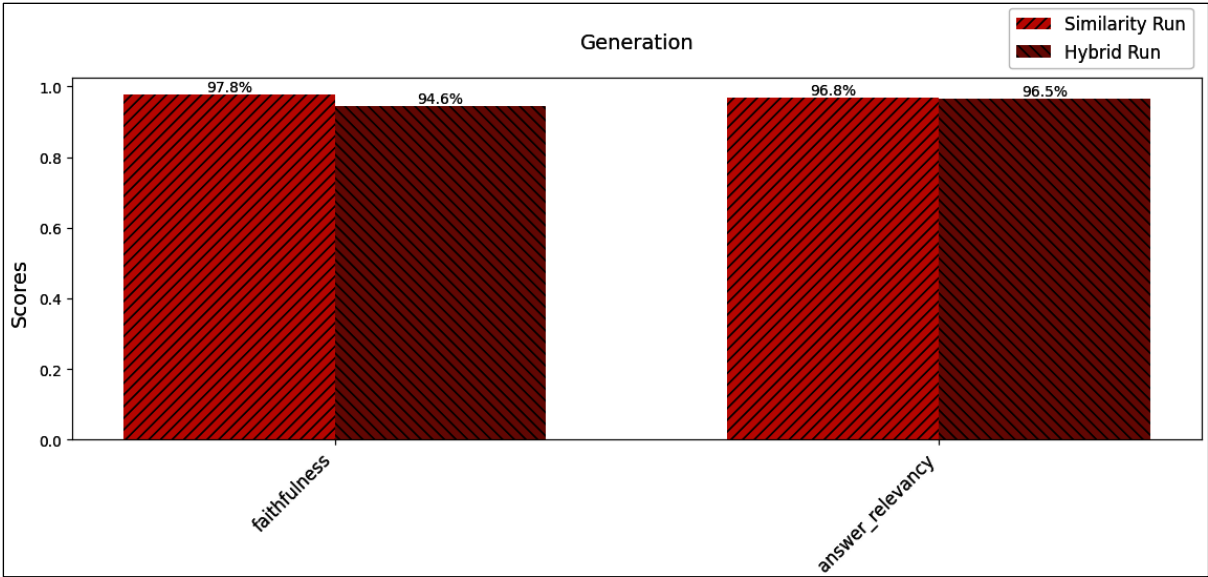
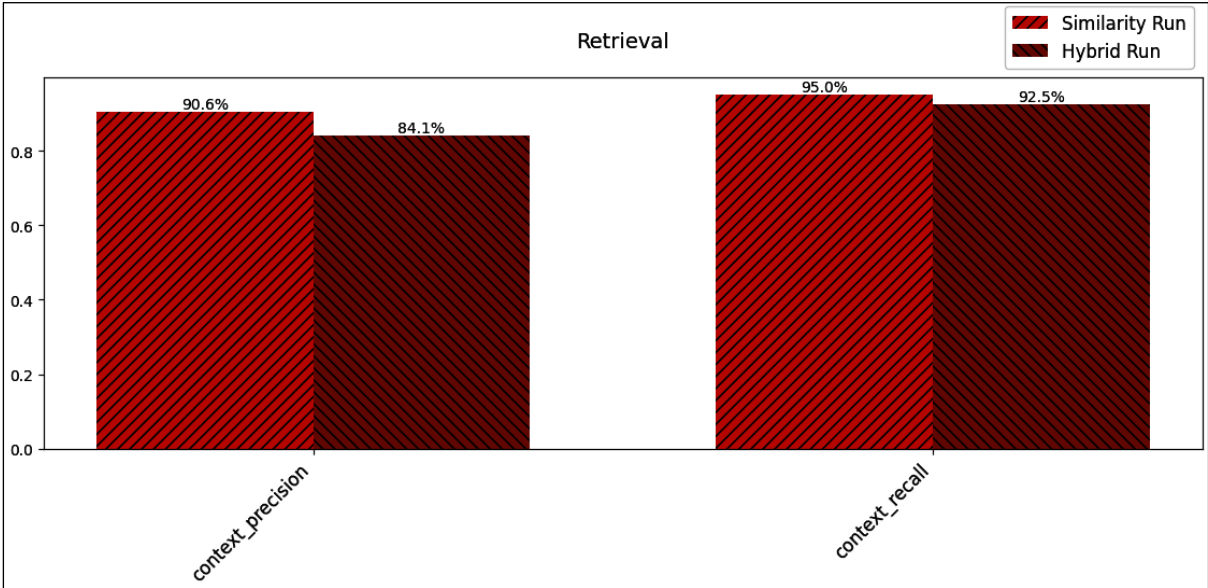
https://kbourne.github.io/chapter1.html, https://kbourne.github.io/chapter1.html, https://kbourne.github.io/chapter1.html

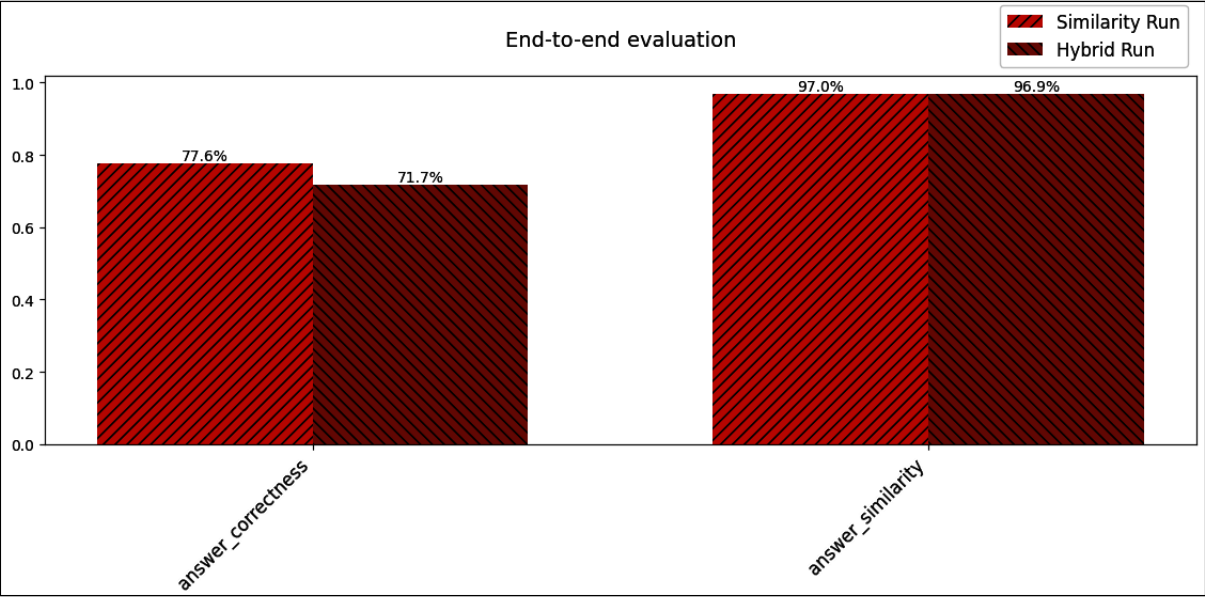
Chapter 8: Similarity Searching with Vectors



Chapter 9: Evaluating RAG Quantitatively and with Visualizations

	question	contexts	ground_truth	evolution_type	metadata	episode_done
0	How does Google Flights provide carbon emissio...	[When individuals search in Google Flights, t...	Google Flights provides carbon emissions estim...	simple	{0}	True
1	How is Google supporting the Nature Conservanc...	[iMasons Climate AccordGoogle is a founding m...	Google supported three of the Nature Conservan...	simple	{0}	True
2	How has Google demonstrated its support for st...	[We've consistently supported strong climate ...	Google has consistently supported strong clima...	simple	{0}	True
3	What innovative water stewardship solutions ar...	[multiple offices around the world, we've ach...	The new Bay View campus, opened in 2022, is on...	simple	{0}	True
4	How much water have Google's contracted waters...	[In addition to focusing on responsible water...	271 million gallons of water	simple	{0}	True





Chapter 11: Using LangChain to Get More from RAG

Upload .txt

Splitter: Recursive Character Text Splitter 

Chunk Size: 

Chunk Overlap: 

Total Characters: 2658
Number of chunks: 9
Average chunk size: 295.3

One of the most important things I didn't understand about the world when I was a child is the degree to which the returns for performance are superlinear.

Teachers and coaches implicitly told us the returns were linear. "You get out," I heard a thousand times, "what you put in." They meant well, but this is rarely true. If your product is only half as good as your competitor's, you don't get half as many customers. You get no customers, and you go out of business.

It's obviously true that the returns for performance are superlinear in business. Some think this is a flaw of capitalism, and that if we changed the rules it would stop being true. But superlinear returns for performance are a feature of the world, not an artifact of rules we've invented. We see the same pattern in fame, power, military victories, knowledge, and even benefit to humanity. In all of these, the rich get richer. [1]

You can't understand the world without understanding the concept of superlinear returns. And if you're ambitious you definitely should, because this will be the wave you surf on.

It may seem as if there are a lot of different situations with superlinear returns, but as far as I can tell they reduce to two fundamental causes: exponential growth and thresholds.

Upload .txt

Splitter: Character Splitter 

Chunk Size: 434 

Chunk Overlap: 0 

Total Characters: 2658

Number of chunks: 7

Average chunk size: 379.7

One of the most important things I didn't understand about the world when I was a child is the degree to which the returns for performance are superlinear.

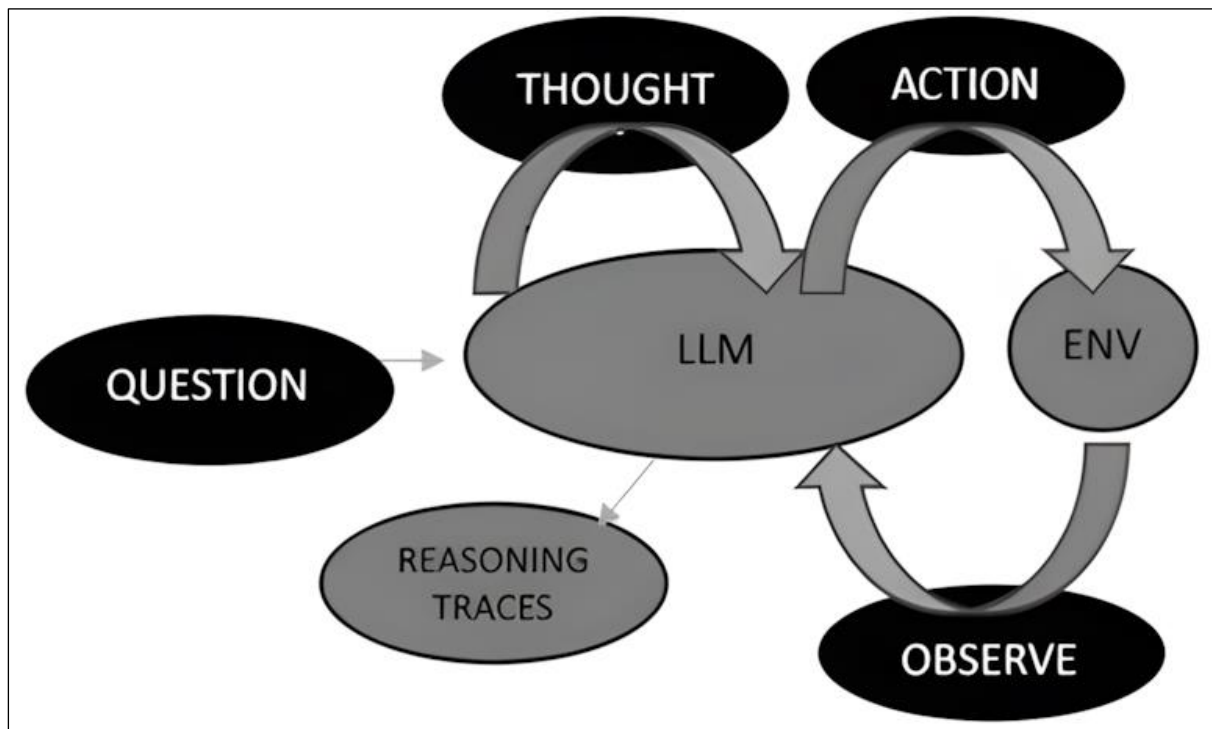
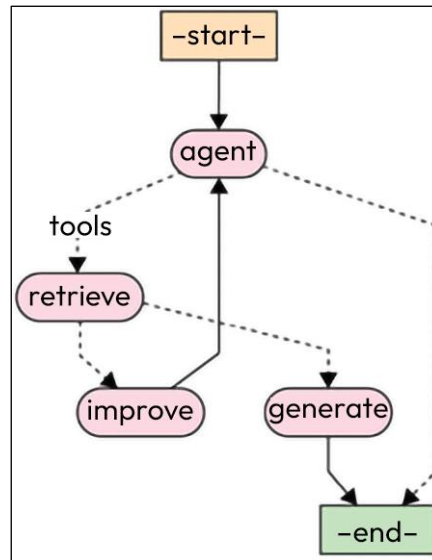
Teachers and coaches implicitly told us the returns were linear. "You get out," I heard a thousand times, "what you put in." They meant well, but this is rarely true. If your product is only half as good as your competitor's, you don't get half as many customers. You get no customers, and you go out of business.

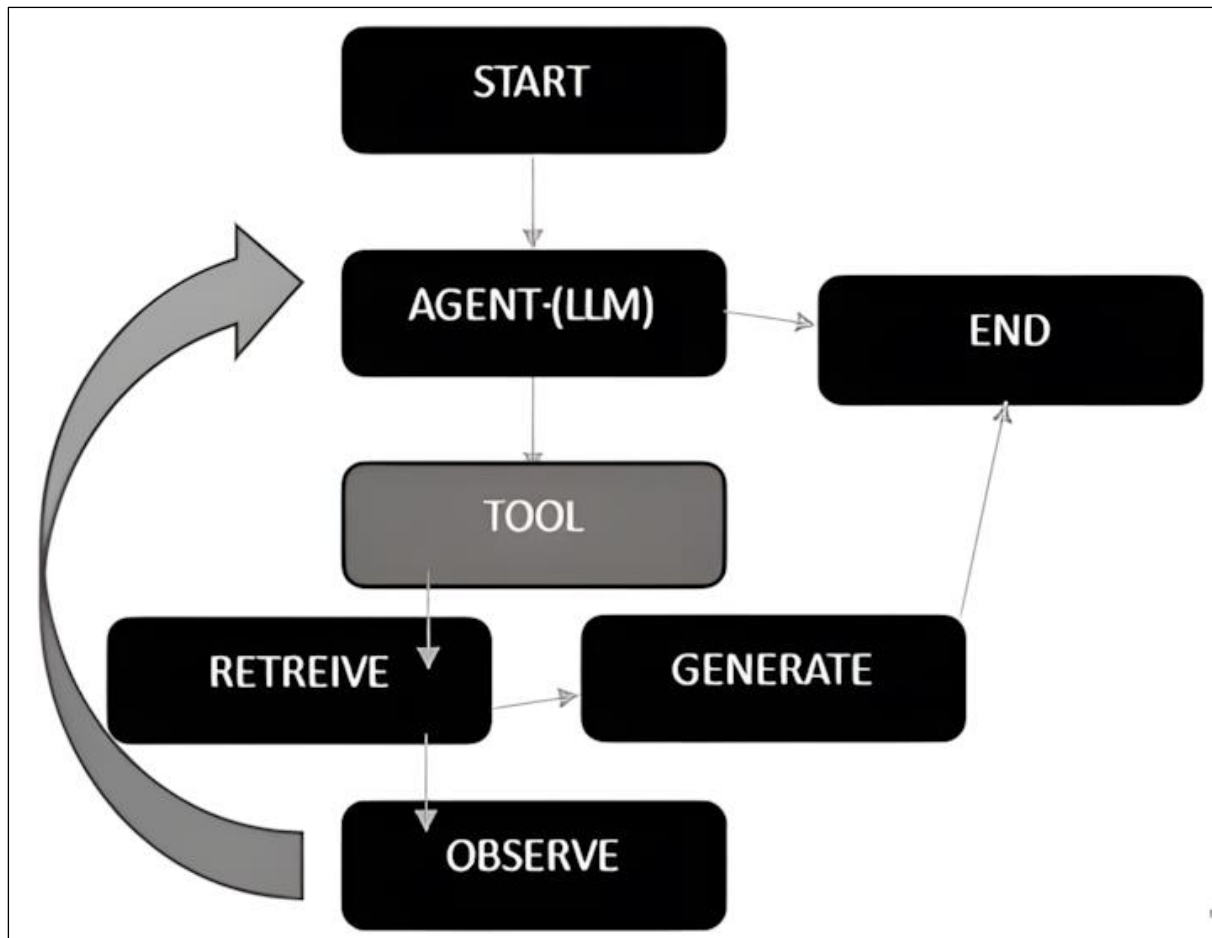
It's obviously true that the returns for performance are superlinear in business. Some think this is a flaw of capitalism, and that if we changed the rules it would stop being true. But superlinear returns for performance are a feature of the world, not an artifact of rules we've invented. We see the same pattern in fame, power, military victories, knowledge, and even benefit to humanity. In all of these, the rich get richer. [1]

You can't understand the world without understanding the concept of superlinear returns. And if you're ambitious you definitely should, because this will be the wave you surf on.

It may seem as if there are a lot of different situations with superlinear returns, but as far as I can tell they reduce to two fundamental causes: exponential growth and thresholds.

Chapter 12: Combining RAG with the Power of AI Agents and LangGraph





Chapter 13: Ontology-Based Knowledge Engineering for Graphs

The screenshot displays the Protégé ontology editor interface. At the top, there are tabs for 'Active ontology', 'Entities', 'Individuals by class', 'DL Query', and 'SPARQL Query'. Below these, a sidebar on the left contains tabs for 'Annotation properties', 'Datatypes', and 'Individuals'. The main area is divided into two panes. The left pane, titled 'Class hierarchy: owl:Thing', shows a tree view with 'owl:Thing' selected. The right pane, titled 'Annotations: owl:Thing', shows a list of annotations. The interface includes various icons for class manipulation and a search bar.

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Class hierarchy: owl:Thing ? [] [] [] [] []

Annotations Usage

Annotations: owl:Thing

Annotations +

Asserted []

owl:Thing

Annotations Usage

Annotations: issuedBy



Annotations

Characteristics:

Description: issuedBy

- ☐ Functional
- ☐ Inverse functional
- ☐ Transitive
- ☐ Symmetric
- ☐ Asymmetric
- ☐ Reflexive
- ☐ Irreflexive

Equivalent To

SubProperty Of

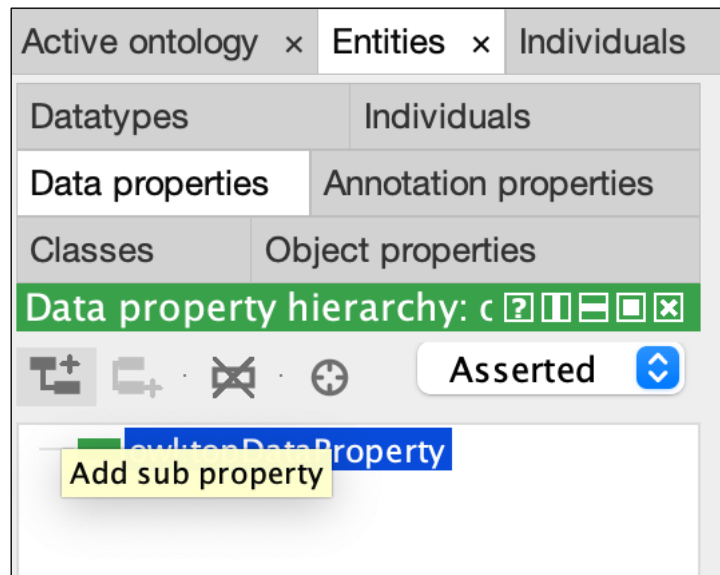
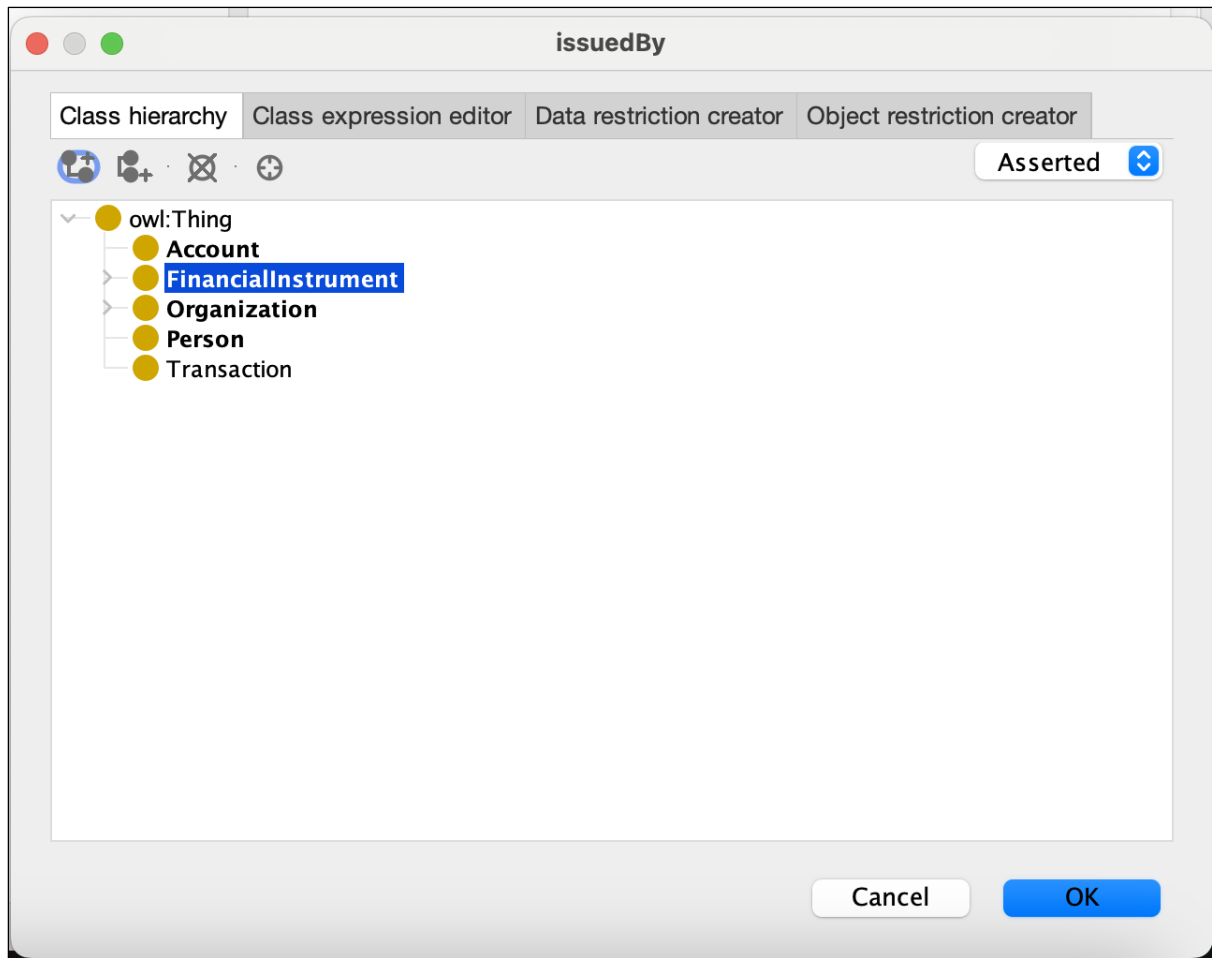
Inverse Of

Domains (intersection)

Ranges (intersection)

Disjoint With

SuperProperty Of (Chain)



Create a new Data property

Name

IRI

New entity options...

Cancel OK

Description: hasTicker

Equivalent To +

SubProperty Of +

Domains (intersection) +

FinancialInstrument ? @ x o

Ranges +

xsd:string ? @ x o

Disjoint With +

Active ontology x Entities x Individuals by class x

Class hierarchy: Orga ? || = □ x

Annotations

Annotations:

Annotations +

- owl:Thing
 - Account
 - FinancialInstrument
 - Organization**
 - Person
 - Transaction

Individuals Individuals (Inferred)

Direct instances: ? || = □ ×

◆+ ✕

For: ● Organization

Add individual

Individuals Individuals (Inferred)

Direct instances: US_T ? || = □ ×

◆+ ✕

For: ● Organization

- ◆ Apple_Inc
- ◆ Microsoft_Corp
- ◆ US_Treasury

Active ontology x Entities x Individuals by class x DL

Class hierarchy: Regu ? II = □ X

Annotations Usa

Annotations: SE

Annotations +

Description: SEC

Types +

Regulatory

Same Individual As

Different Individuals

Individuals Individuals (Inferred)

Direct instances: SEC ? II = □ X

+ X

For: RegulatoryAuthority

SEC

Active ontology x Entities x Individuals by class x

Class hierarchy: Stock ? ? ? ? ?

+

+

✗

+

Asserted

owl:Thing

- Account
- FinancialInstrument
 - Debt
 - Equity
 - Stock
- Organization
 - RegulatoryAuthority
- Person
- Transaction

Annotations

Annotations: Annotations +

Description: Types +

- Stock

Same Individual As

Different Individual

Individuals Individuals (Inferred)

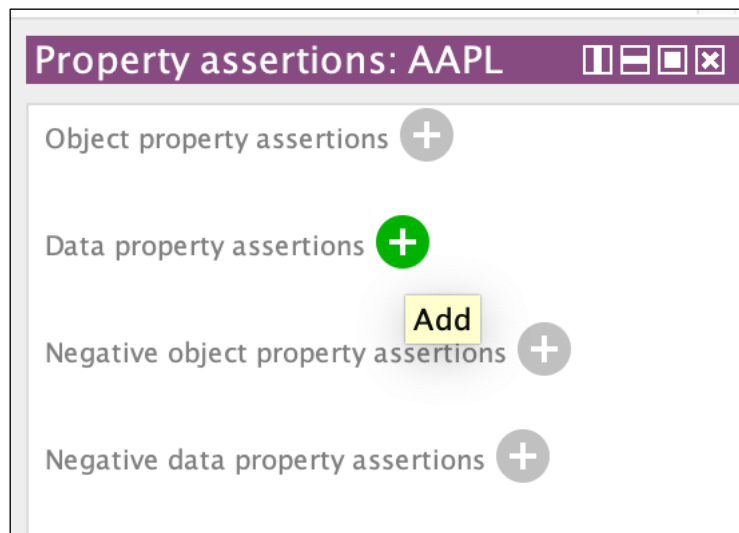
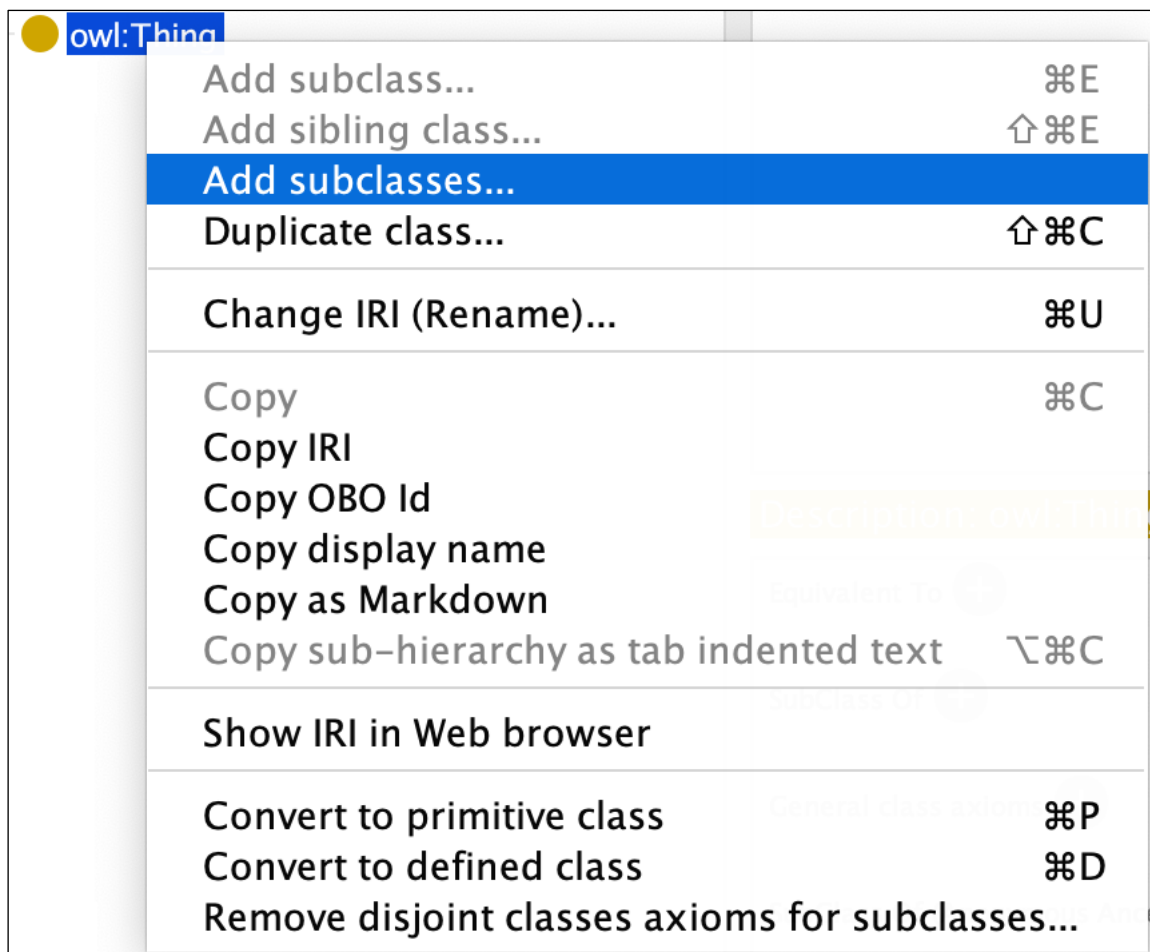
Direct instances: MSFT ? ? ? ? ?

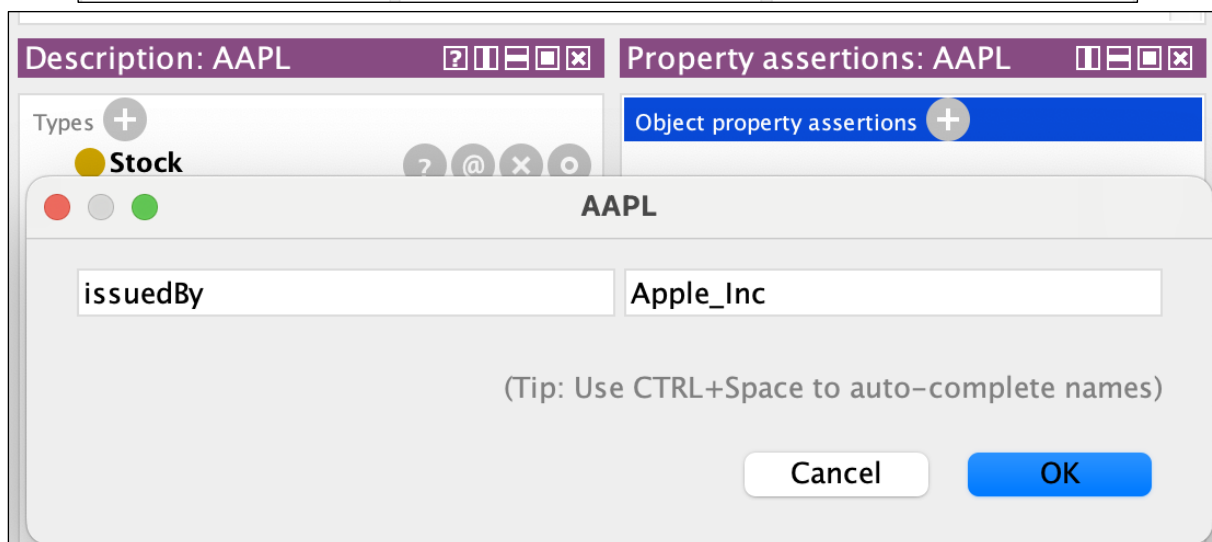
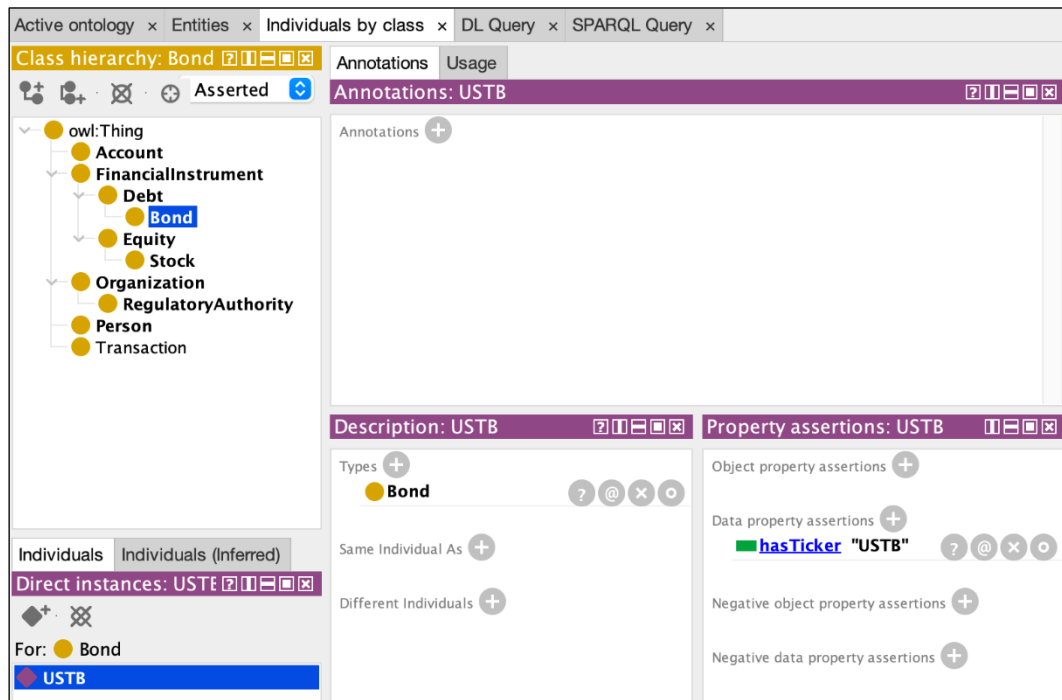
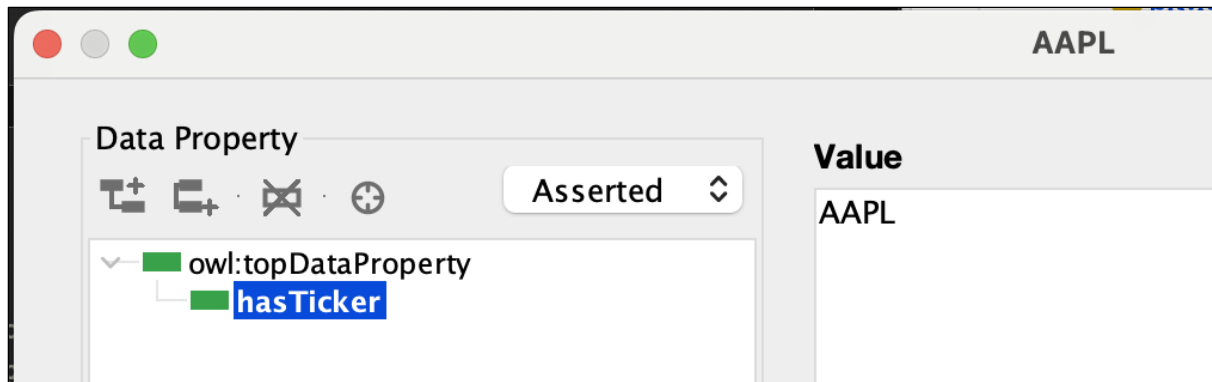
+

✗

For: Stock

- AAPL
- MSFT





Individuals

Individuals (Inferred)

Direct instances: USTB

For: Bond

USTB

Description: USTB

Types

Bond

Same Individual As

Different Individuals

Property assertions: USTB

Object property assertions

isRegulatedBy SEC

issuedBy US_Treasury

Data property assertions

hasTicker "USTB"

Negative object property assertions

Enter hierarchy

Please enter one name per line. You can use tabs to indent names to create a hierarchy.

FinancialInstrument
 Equity
 Stock
 Debt
 Bond
Organization
 RegulatoryAuthority
Person
Account

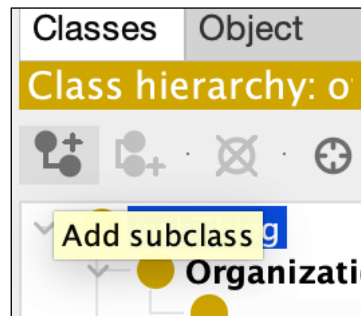
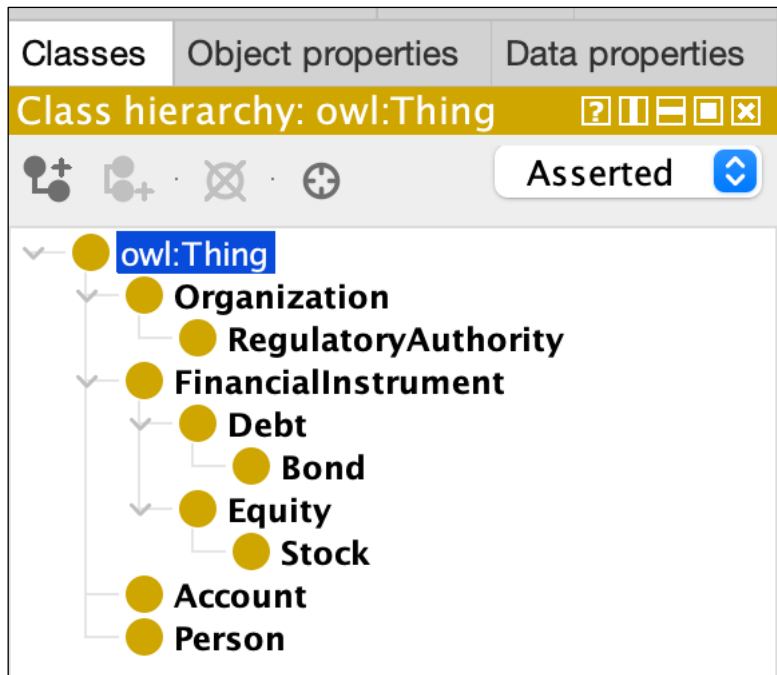
Prefix

Suffix

Go Back

Continue

Cancel



Create a new Class

Name Transaction

IRI web.org/keithbourne/ontologies/2025/8/FinancialOntology10#Transaction

New entity options...

Cancel OK

Active ontology x Entities x Individuals by class x DL Query x SPARQL Query x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Object property hierarchy: owl ? ? ? ? ?

Annotations Usage

Annotations: owl:topObjectProperty

Annotations +

Asserted

Add sub property

Create a new Object property

Name issuedBy

IRI ticweb.org/keithbourne/ontologies/2025/8/FinancialOntology10#issuedBy

New entity options...

Cancel OK

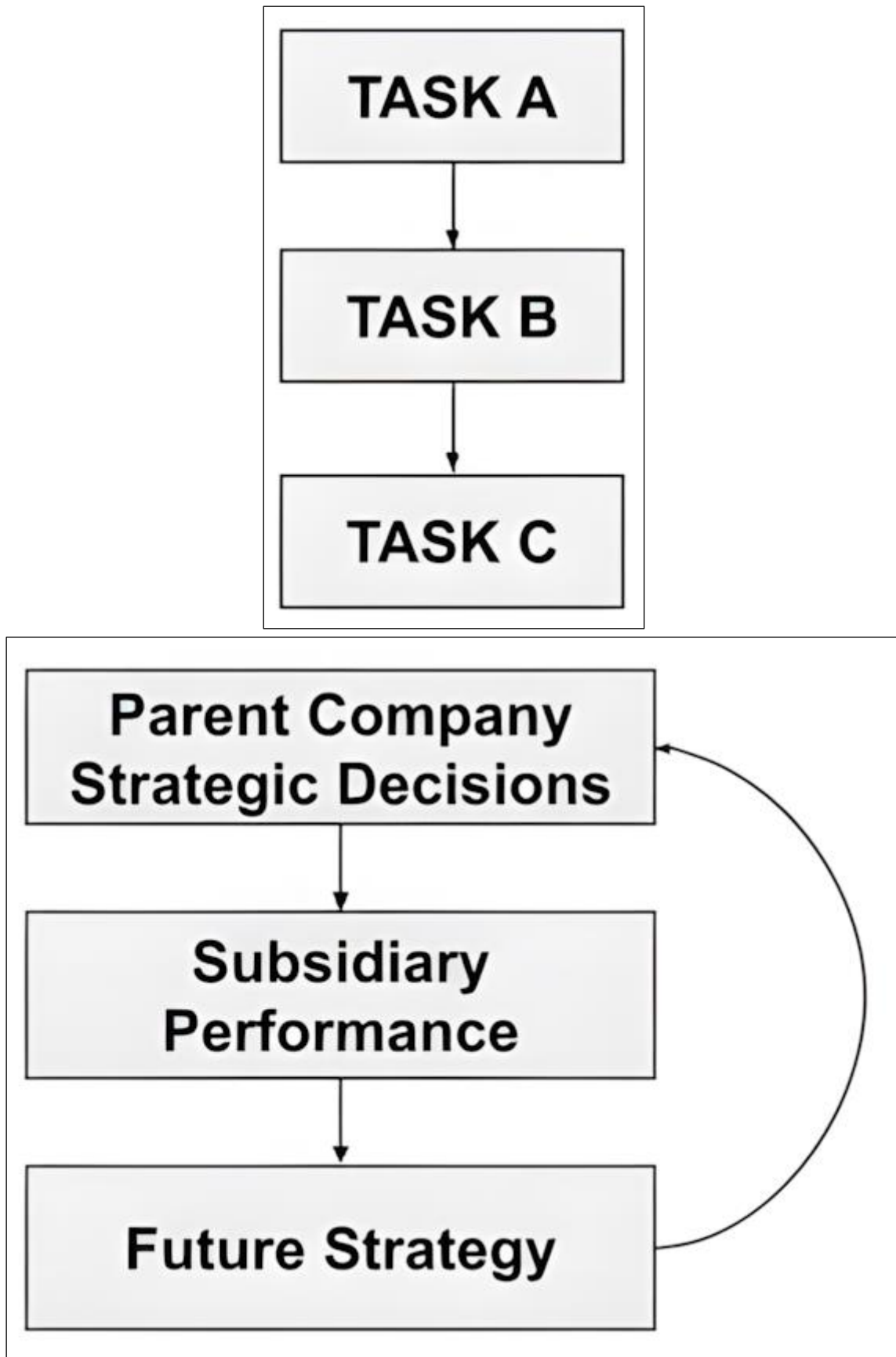
Classes Object properties Data properties

Object property hierarchy: issu ? ? ? ? ?

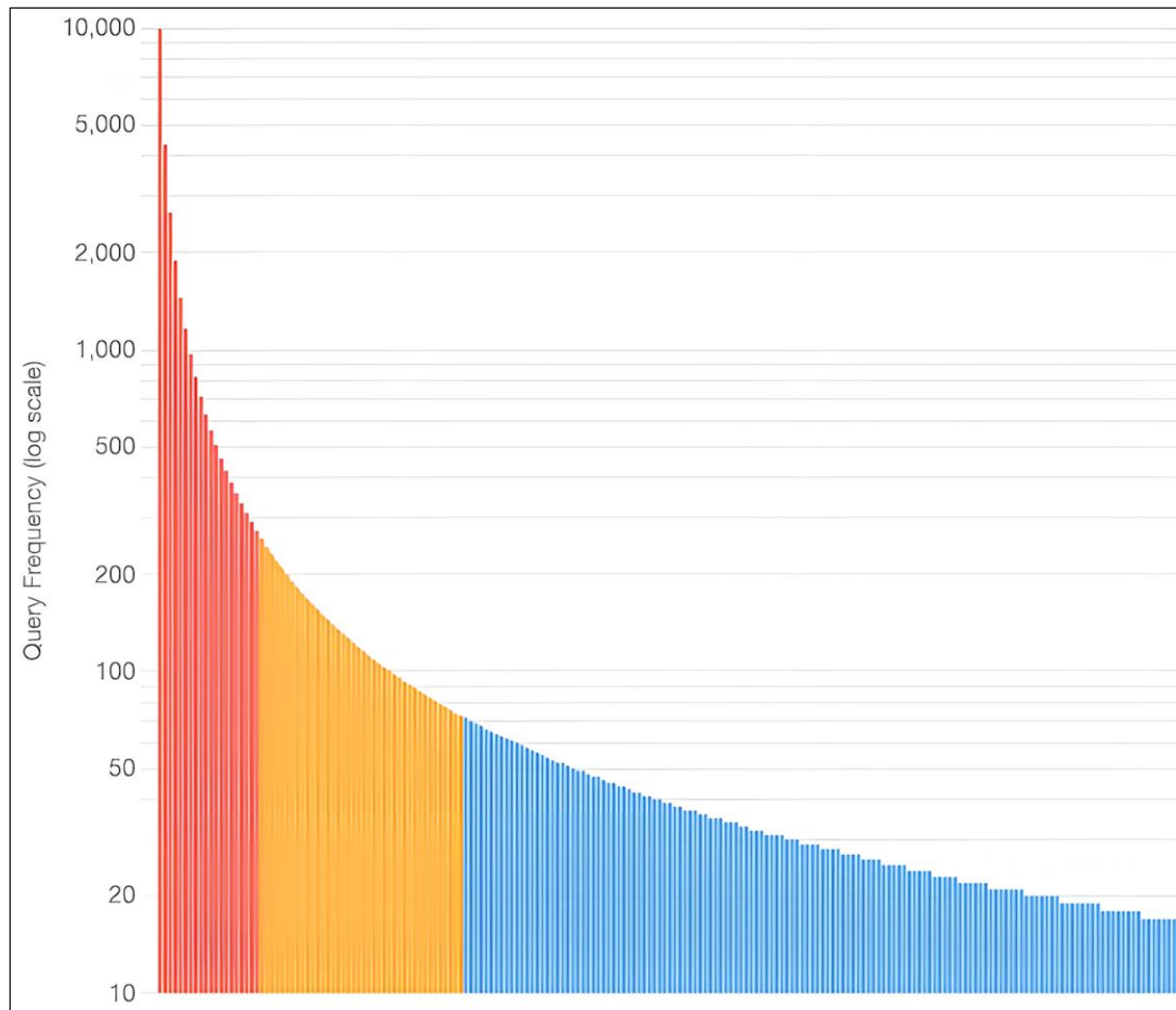
Asserted

- owl:topObjectProperty
 - issuedBy
 - ownedBy
 - isRegulatedBy

Chapter 14: Graph-Based RAG



Chapter 15: Semantic Caches



Chapter 16: Agentic Memory: Extending RAG with Stateful Intelligence

