

# Harnessing GenAI in DevOps



**John Willis**  
**@botchagalupe**

# Agenda

---

- Introduction
- Brief History
- DevOps'ing of GenAI
- GenAI Basics
- Demonstration
- Workshop Labs



# Introduction



# John Willis - @botchagalupe

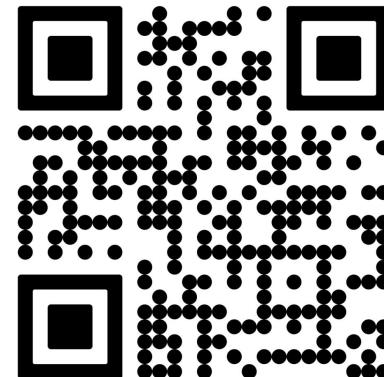
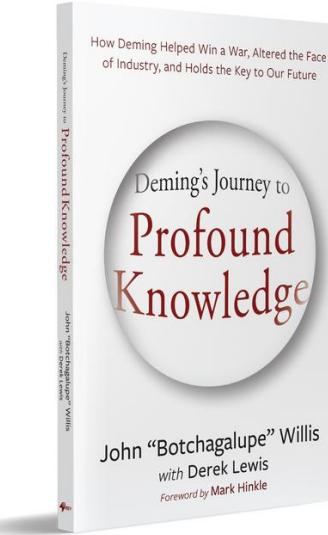
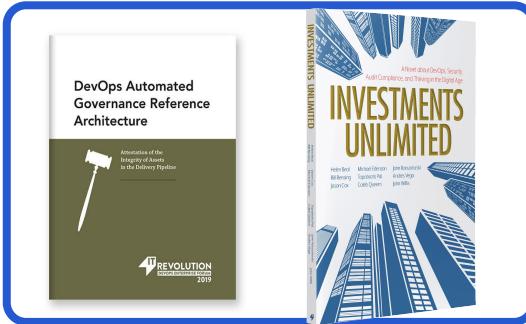
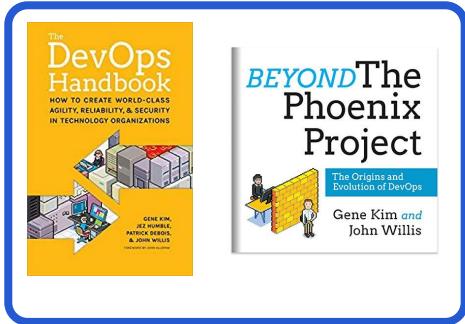


**John Willis**

As an accomplished author and innovative entrepreneur, I am deeply passionate abo...



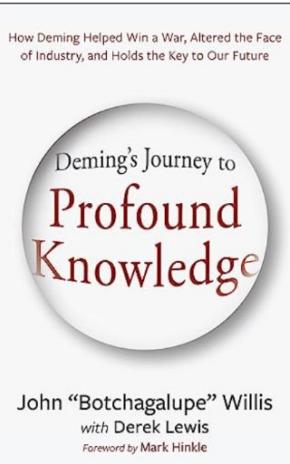
# @botchagalupe



John Willis @botchagalupe

You purchased this item on August 11, 2023.

[View this order](#) | You can find this title on your Amazon Kindle apps and devices, or the [Manage Your Content and Devices](#) page. Need additional help? [Click here](#)



# Deming's Journey to Profound Knowledge: How Deming Helped Win a War, Altered the Face of Industry, and Holds the Key to Our Future

by [John Willis](#) (Author) | Format: Kindle Edition

4.3 11 ratings

[See all formats and editions](#)

Kindle

\$10.99

You Earn: 33 pts

Audiobook

\$17.99

In your library

Paperback

\$28.00

1 New from \$28.00

Read with Our [Free App](#)

From the birth of modern industry, to winning WWII, to Japan's Economic Miracle, W. Edwards Deming helped shape some of the most profound moments in modern history. Deming, an American engineer and statistician, is widely recognized for his contributions to the field of quality management. But his teachings go beyond quality management; they influence not only how we work today, but also how we can continue to succeed into the future.

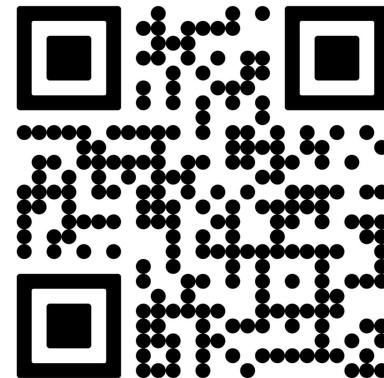
This Kindle book is part of the Kindle Single series. Kindle Singles are short books designed for Kindle devices. They are not available in print or audio formats. Kindle Singles are sold at discounted prices and are not eligible for Kindle Unlimited or Kindle Prime.

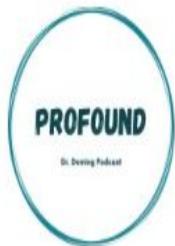
[Read more](#)

[Read sample](#)

[Audible sample](#)

Read Now  
  
You already own this item. Read anytime in your browser, or on your Kindle apps and devices.





NEWSLETTER

## Profound

Profound is a newsletter applying W. Edwards Deming's ideas in the digital transformation era.



By John Willis

As an accomplished author and innovative entrepre...

Published weekly

**1,352 subscribers**

...



NEWSLETTER

## Attention Is All You Need

A Botchagalupe spin on all thing Generative AI.



By John Willis

As an accomplished author and innovative entrepre...

Published weekly

**778 subscribers**

...

<https://github.com/OperationalizingAI/DevOpsGAIWorkshop>



botchagalupe@gmail.com

# Brief History of GenAI



# The History of AI

1940	McCulloch-Pitts (Mathematical Neuron) Vannevar Bush (As We May Think)	
1950	Alan Turing (The Turing Test) John McCarthy (Dartmouth Workshop) Douglas Engelbart (SRI) Frank Rosenblatt (Perceptrons) Arthur Samuel (Checkers)	
1970	First AI Winters Expert Systems Edmund Feigenbaum (DENDRAL) MARPA (Shakey) MYCIN	
1980	Second AI Winter DEC RI XCON Fifth Generation Computer Systems (FGCS) Geoffrey Hinton (Backpropagation)	
1990	IBM Deep Blue Gerald Tesauro (TDGammmon) Yann LeCun (MNIST)	
2010	IBM Watson (Jeopardy) Fei Fei Li (Imagenet) Geoffrey Hinton (AlexNet) Ian Goodfellow (GAN) DeepMind (AlphaGo) Andrew Ng (Google Brain) Google (Attention is All You Need)	
2020	OpenAI GPT-3 OpenAI ChatGPT Anthropic Meta (LaMA) Google (Gemini) Joy Buolamwini (Unmasking AI)	

The history of artificial intelligence (AI) traces back to the mid-20th century, evolving from ambitious beginnings at the 1956 Dartmouth Conference to a field marked by cycles of breakthroughs and setbacks. Despite challenges, advancements in algorithms, computing power, and data have driven significant progress, leading to technologies like machine learning and deep learning that have transformed industries and society. AI's journey reflects a blend of innovation and ethical considerations, underscoring its deep impact on the contemporary world.



NEWSLETTER

Attention Is All You Need

A Botchagalupe spin on all thing Generative AI.

# LLM Historical Highlights

## Attention Is All You Need

Ashish Vaswani\*  
Google Brain  
avaswani@google.com

Noam Shazeer\*  
Google Brain  
noam@google.com

Niki Parmar\*  
Google Research  
nikip@google.com

Jakob Uszkoreit\*  
Google Research  
usz@google.com

Llion Jones\*  
Google Research  
llion@google.com

Aidan N. Gomez\* †  
University of Toronto  
aidan@cs.toronto.edu

Łukasz Kaiser\*  
Google Brain  
lukasz.kaiser@google.com

Ilia Polosukhin\* ‡  
illia.polosukhin@gmail.com

### Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train.

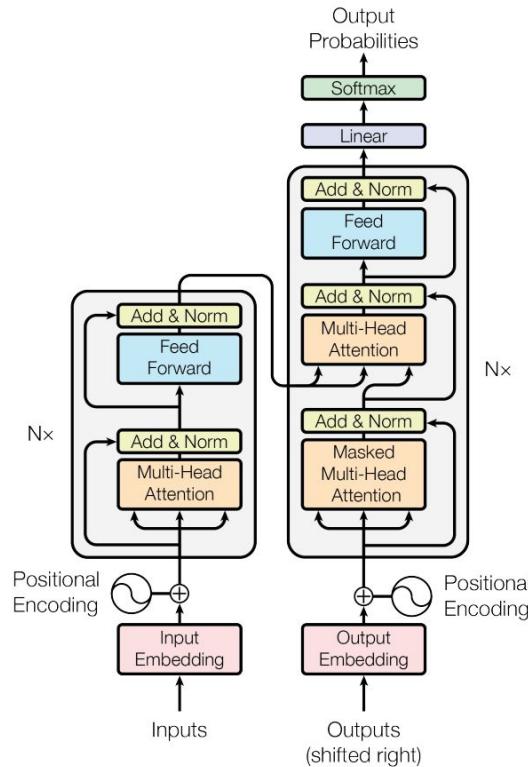


Figure 1: The Transformer - model architecture.

# DevOps'ing of GenAI



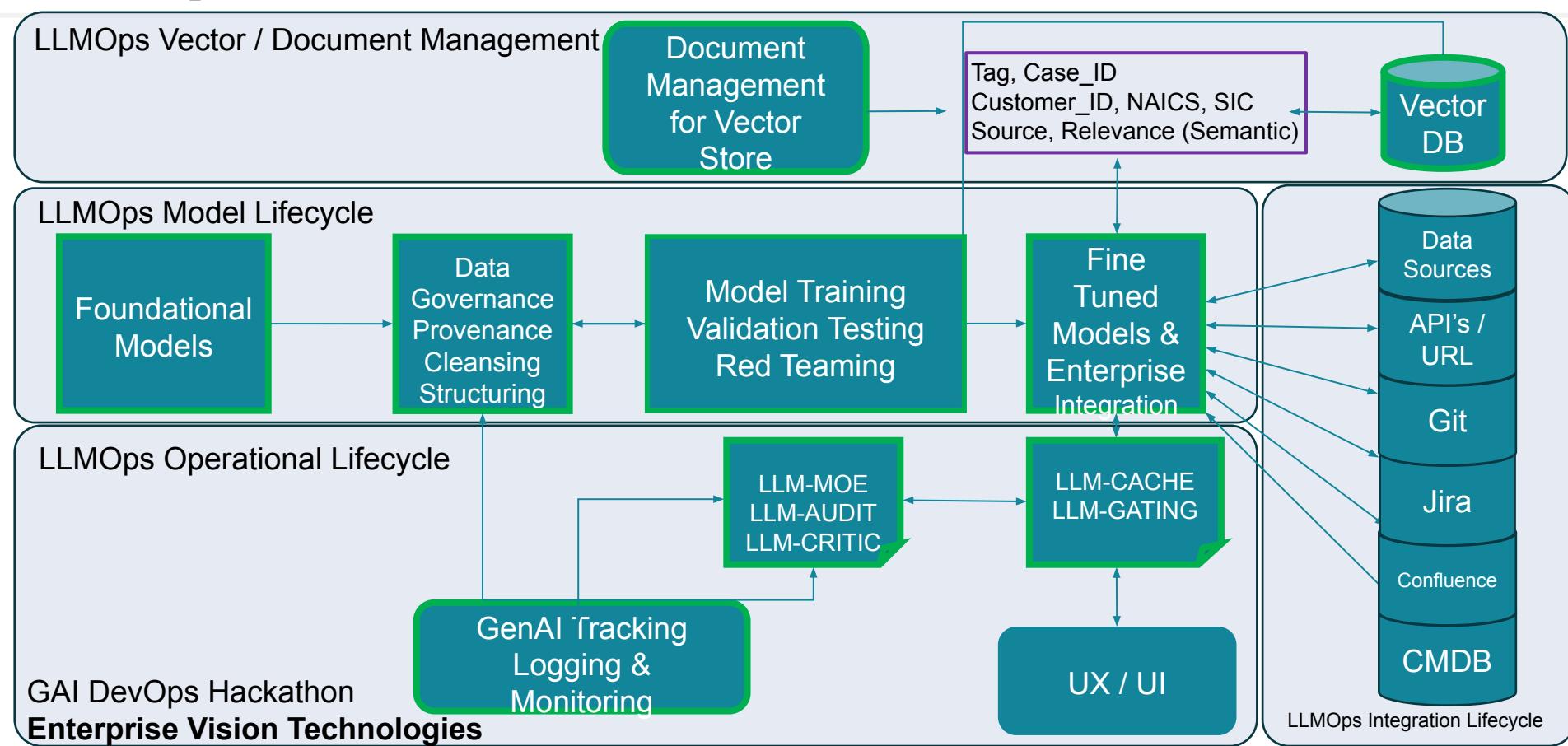
# GenAI DevOps and DevSecOps Use Cases

---

- Repo Analysis
- IT Audit Assist
- Incident Analysis
- Shell Copilots
- Security Review Analysis
- Slack/MatterMost Integration
- GPTScript
- Opendevin.ai

<https://github.com/orgs/OperationalizingAI/repositories>

# AI4Ops Architecture



# GenAI Basics



# GenAI Basics

---

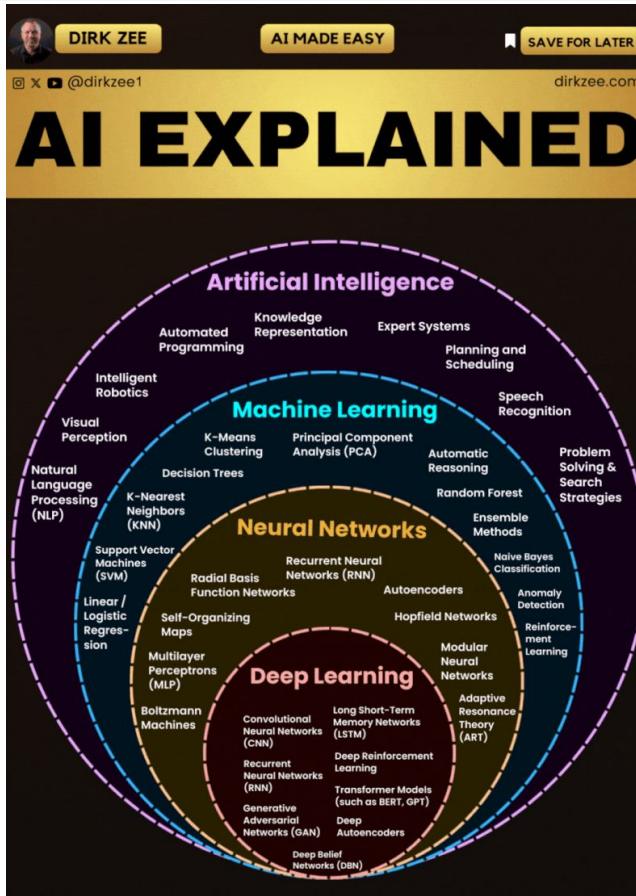
- Types of AI
- Machine Learning vs Deep Learning
- Neural Networks
- Large Language Models
- Vectors Search (embeddings)
- Retrieval Augmentation Generation (RAG)
- Query Chain Types
- Evaluations

# Types of AI

---

- **Narrow AI (ANI)**
  - Specialized AI for specific tasks, like Siri or Alexa.
- **General AI (AGI)**
  - Theoretical AI with human-like abilities across tasks; not yet achieved.
- **Super AI (ASI)**
  - Hypothetical AI surpassing human intelligence in all aspects.

# AI in a Nutshell



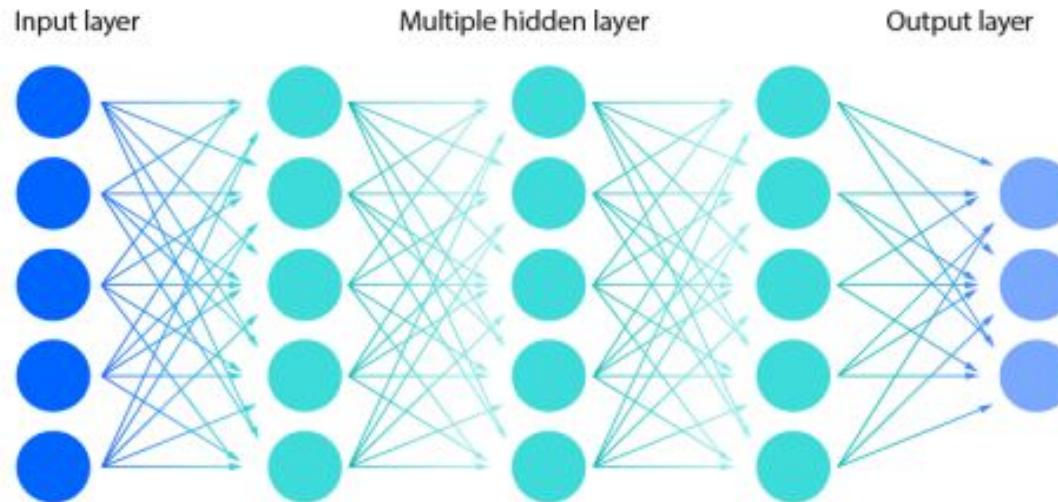
# ML versus DL

---

- **Machine Learning**
  - A subset of AI that involves training algorithms to learn from and make predictions or decisions based on data.
- **Deep Learning**
  - A subset of Machine Learning, uses neural networks with many layers (hence "deep") to analyze vast amounts of data, enabling more complex and accurate predictions.

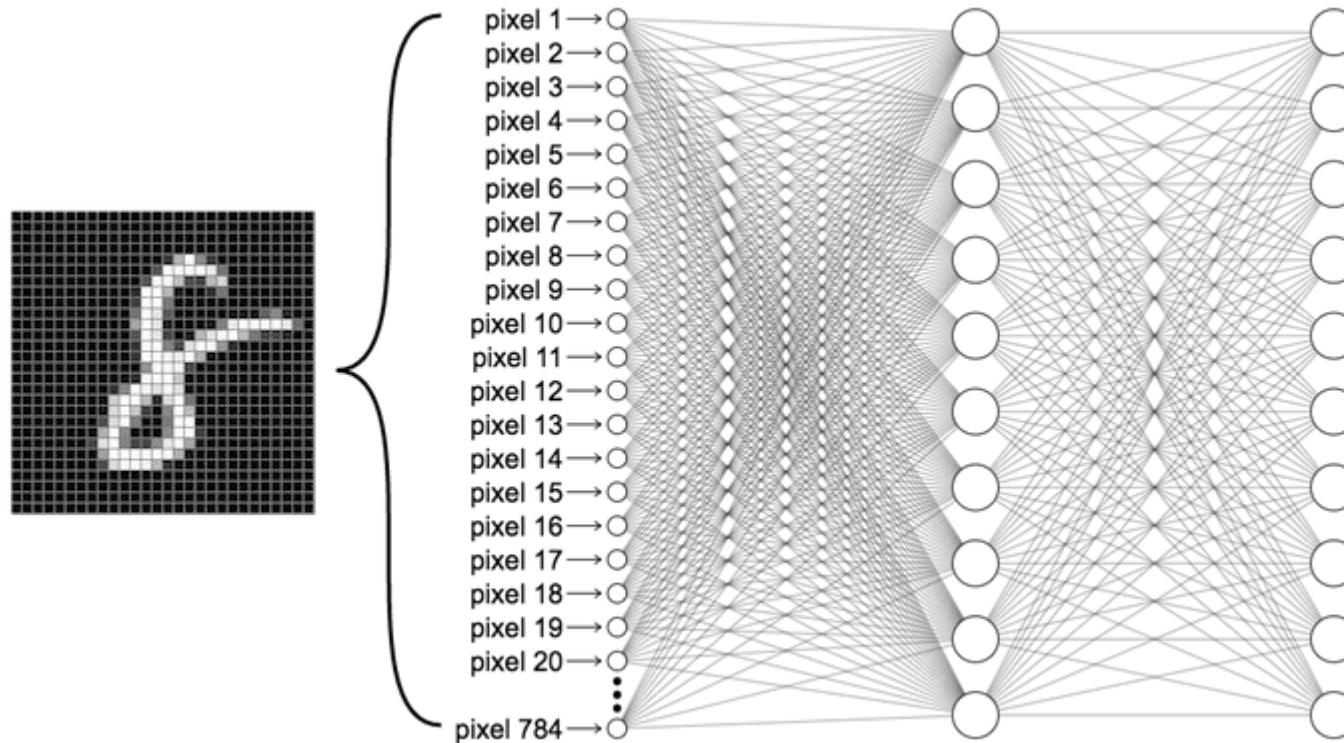
# Neural Networks

## Deep neural network



<https://www.ibm.com/topics/neural-networks>

# Neural Networks - MNIST Dataset



# Foundational (Frontier) LLMs

---

- Trained on the same corpus and given the same question, will give similar answers
- No knowledge of your data
- Knowledge not up-to-date
- Can hallucinate

# Large Language Models

---

- A Large Language Model (LLM)
  - An (AI) program capable of performing tasks such as recognizing and generating text. LLMs are named "large" due to the vast amount of data they are trained on. These models are built using a transformer model, which is a type of neural network.
- Frontier Models
  - Microsoft - OpenAI
  - Google - Gemini
  - Amazon - Bedrock/Claude
  - Meta - Llama
  - Mistral AI - Mixtral 8x7B

# What are embeddings?

OpenAI's text embeddings measure the relatedness of text strings. Embeddings are commonly used for:

- **Search** (where results are ranked by relevance to a query string)
- **Clustering** (where text strings are grouped by similarity)
- **Recommendations** (where items with related text strings are recommended)
- **Anomaly detection** (where outliers with little relatedness are identified)
- **Diversity measurement** (where similarity distributions are analyzed)
- **Classification** (where text strings are classified by their most similar label)

An embedding is a vector (list) of floating point numbers. The **distance** between two vectors measures their relatedness. Small distances suggest high relatedness and large distances suggest low relatedness.

# Embeddings

Spaces mteb/leaderboard like 2.16k Running

App

Files

Community 84

Massive Text Embedding Benchmark (MTEB) Leaderboard. To submit, refer to the [MTEB GitHub repository](#). Refer to the [MTEB paper](#) for details on metrics, tasks and models.

Overall Bitext Mining Classification Clustering Pair Classification Reranking Retrieval STS Summarization

English Chinese French Polish

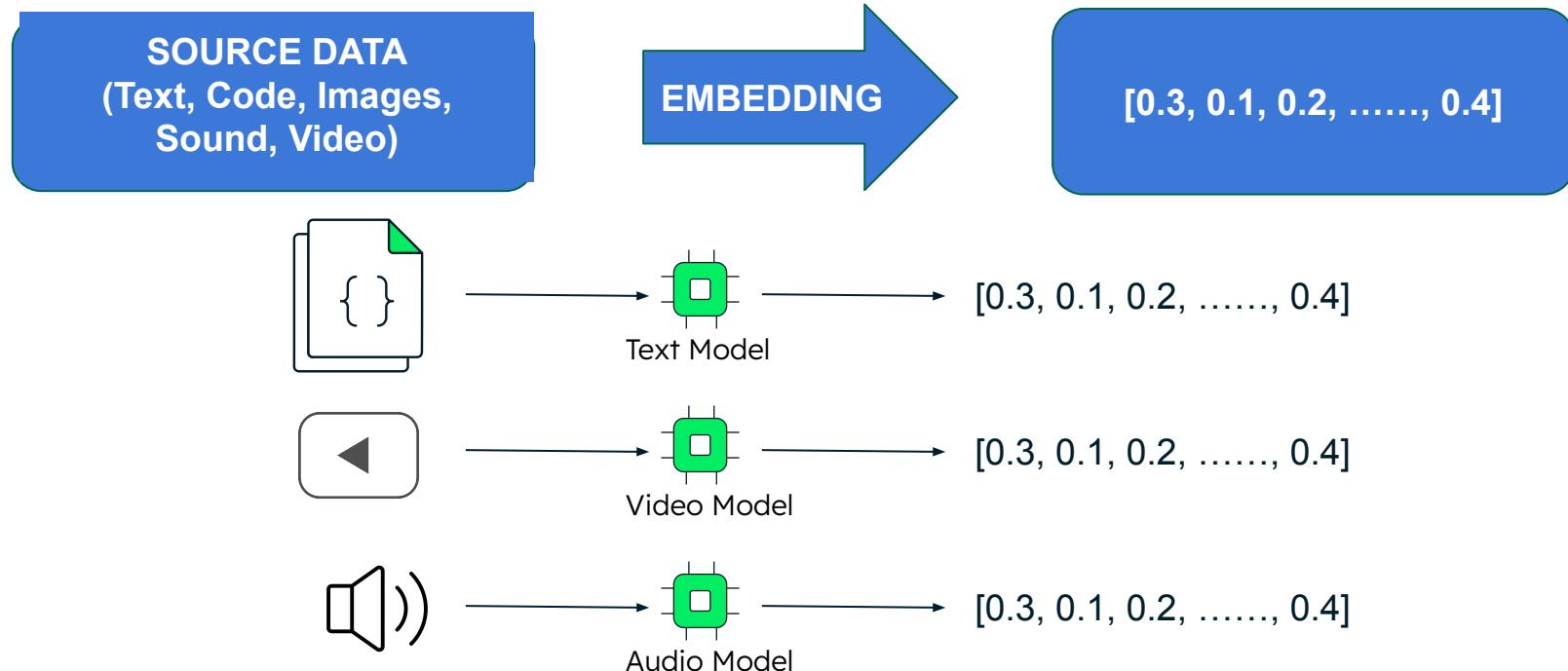
Overall MTEB English leaderboard 🌎

- Metric: Various, refer to task tabs
- Languages: English

Rank	Model	Model Size (GB)	Embedding Dimensions	Max Tokens	Average (56 datasets)	Classification Average (12 datasets)	Clustering Average (11 datasets)	Pair Classification Average (3 datasets)	Reranking Average (4 datasets)	Retrieval Average (15 datasets)	ST Av (1 da
1	<a href="#">SFR-Embedding-Mistral</a>	14.22	4096	32768	67.56	78.33	51.67	88.54	60.64	59	85
3	<a href="#">GritLM-7B</a>	14.48	4096	32768	66.76	79.46	50.61	87.16	60.49	57.41	83
4	<a href="#">e5-mistral-7b-instruct</a>	14.22	4096	32768	66.63	78.47	50.26	88.34	60.21	56.89	84
2	<a href="#">voyage-lite-02-instruct</a>		1024	4000	67.13	79.25	52.42	86.87	58.24	56.6	85
10	<a href="#">voyage-lite-01-instruct</a>		1024	4000	64.49	74.79	47.4	86.57	59.74	55.58	82
6	<a href="#">echo-mistral-7b-instruct-last</a>	14.22	4096	32768	64.68	77.43	46.32	87.34	58.14	55.52	82
9	<a href="#">text-embedding-3-large</a>		3072	8191	64.59	75.45	49.01	85.72	59.16	55.44	81
5	<a href="#">GritLM-8x7B</a>	93.41	4096	32768	65.66	78.53	50.14	84.97	59.8	55.09	83
11	<a href="#">Cohere-embed-english-v3.0</a>		1024	512	64.47	76.49	47.43	85.84	58.01	55	82
8	<a href="#">UAE-Large-V1</a>	1.34	1024	512	64.64	75.58	46.73	87.25	59.88	54.66	84
7	<a href="#">mxbai-embed-large-v1</a>	0.67	1024	512	64.68	75.64	46.71	87.2	60.11	54.39	85
14	<a href="#">bge-large-en-v1.5</a>	1.34	1024	512	64.23	75.97	46.08	87.12	60.03	54.29	83

# Vector Embeddings

**Vectors are numeric representation of data and related context.**



# OpenAI Embedding (Encoding and Decoding)

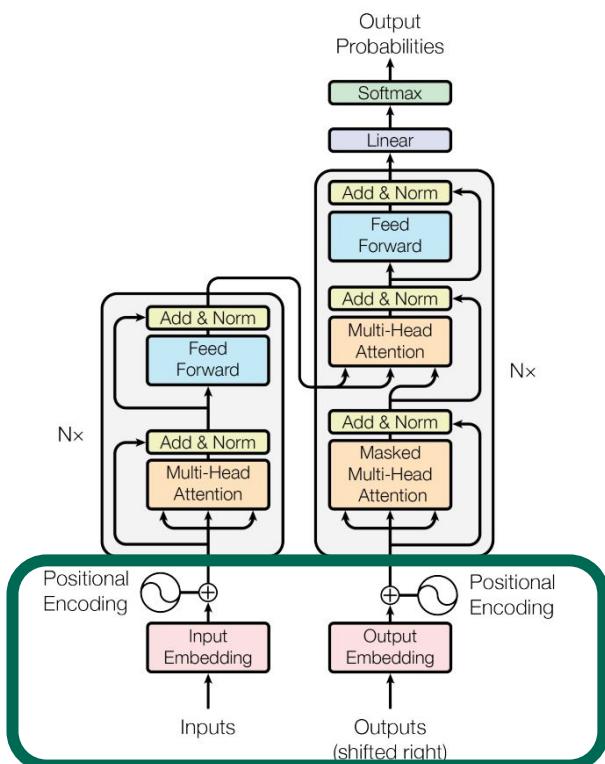


Figure 1: The Transformer - model architecture.

## Encodings

tiktoken is a fast BPE tokeniser for use with OpenAI's models.

tiktoken supports three encodings used by OpenAI models:

Encoding name	OpenAI models
cl100k_base	gpt-4, gpt-3.5-turbo, text-embedding-ada-002
p50k_base	Codex models, text-davinci-002, text-davinci-003
r50k_base (or gpt2 )	GPT-3 models like davinci

## Encoding and Decoding



<https://platform.openai.com/tokenizer>

# Embedding - Code Demo1

## Output

Overlap	Chunk 0	Chunk 1	Chunk 2	Chunk 3	Chunk 5	Chunk 6	Chunk 7	Chunk 9	Chunk 11	Chunk 13	Chunk 15	Chunk 16
Chunk 17	Chunk 19	Chunk 21	Chunk 23	Chunk 24	Chunk 25	Chunk 27	Chunk 29	Chunk 31	Chunk 33	Chunk 34	Chunk 35	
Chunk 36	Chunk 37	Chunk 38	Chunk 39	Chunk 41	Chunk 42	Chunk 43	Chunk 45	Chunk 46	Chunk 47	Chunk 49	Chunk 51	
Chunk 52	Chunk 53	Chunk 54	Chunk 56	Chunk 58	Chunk 59	Chunk 60	Chunk 62	Chunk 63	Chunk 64	Chunk 65	Chunk 67	
Chunk 69	Chunk 70	Chunk 71	Chunk 73	Chunk 75	Chunk 76	Chunk 77	Chunk 79	Chunk 81	Chunk 83	Chunk 85	Chunk 86	
Chunk 87	Chunk 89	Chunk 90	Chunk 91	Chunk 92	Chunk 94	Chunk 96	Chunk 98	Chunk 99	Chunk 100	Chunk 102	Chunk 104	
Chunk 105	Chunk 106	Chunk 107	Chunk 109	Chunk 110	Chunk 111	Chunk 113	Chunk 115	Chunk 116	Chunk 117	Chunk 119		
Chunk 121	Chunk 123	Chunk 124	Chunk 125	Chunk 127	Chunk 128	Chunk 129	Chunk 130	Chunk 131	Chunk 132	Chunk 134		
Chunk 135	Chunk 136	Chunk 137	Chunk 139	Chunk 140	Chunk 142	Chunk 143	Chunk 144	Chunk 145	Chunk 146	Chunk 148		
Chunk 150	Chunk 152	Chunk 153	Chunk 154	Chunk 155	Chunk 157	Chunk 158	Chunk 159	Chunk 160	Chunk 162	Chunk 163		
Chunk 164	Chunk 166	Chunk 167	Chunk 168	Chunk 169	Chunk 170	Chunk 172	Chunk 173	Chunk 174	Chunk 175	Chunk 176		
Chunk 177	Chunk 178	Chunk 179	Chunk 180	Chunk 182	Chunk 184	Chunk 185	Chunk 186	Chunk 188	Chunk 190	Chunk 191		
Chunk 192	Chunk 194	Chunk 196	Chunk 197	Chunk 198	Chunk 200	Chunk 201	Chunk 202	Chunk 204	Chunk 205	Chunk 206		
Chunk 208	Chunk 209	Chunk 210	Chunk 212	Chunk 213								

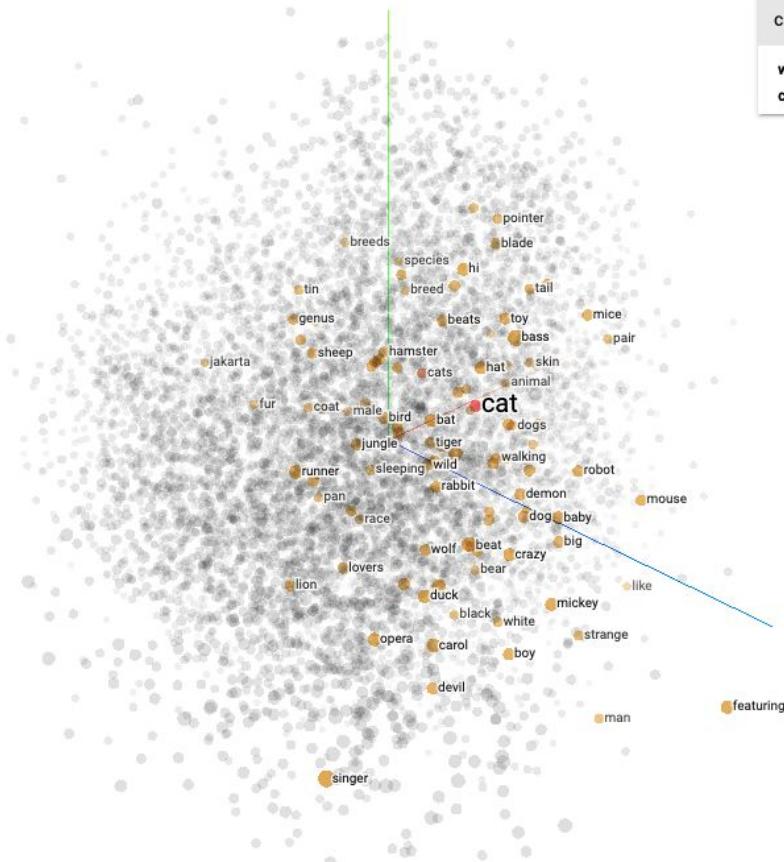
Deming's Journey to Profound Knowledge – How Deming Helped Win a War, Altered the Face of Industry, and Holds the Key to Our Future – Part 1 – Chapter 1: Humble Origins and Non-Determinism

### Deming's Humble Beginnings

Deming's one childhood claim to fame was when Buffalo Bill recognized him in the crowd during a performance of "Buffalo Bill's Wild West Show" outside Los Angeles where "Edwards," as his family called him, was visiting his cousins.

The notoriously flamboyant showman knew the boy from Cody, Wyoming, by sight if not by name. Buffalo Bill was arguably the most world-famous living American at the time, having extensively toured the US and then Europe, performing before Queen Elizabeth II and King George VI.

# Vector Embeddings (Word2Vec)



Search by word cat

neighbors 10C

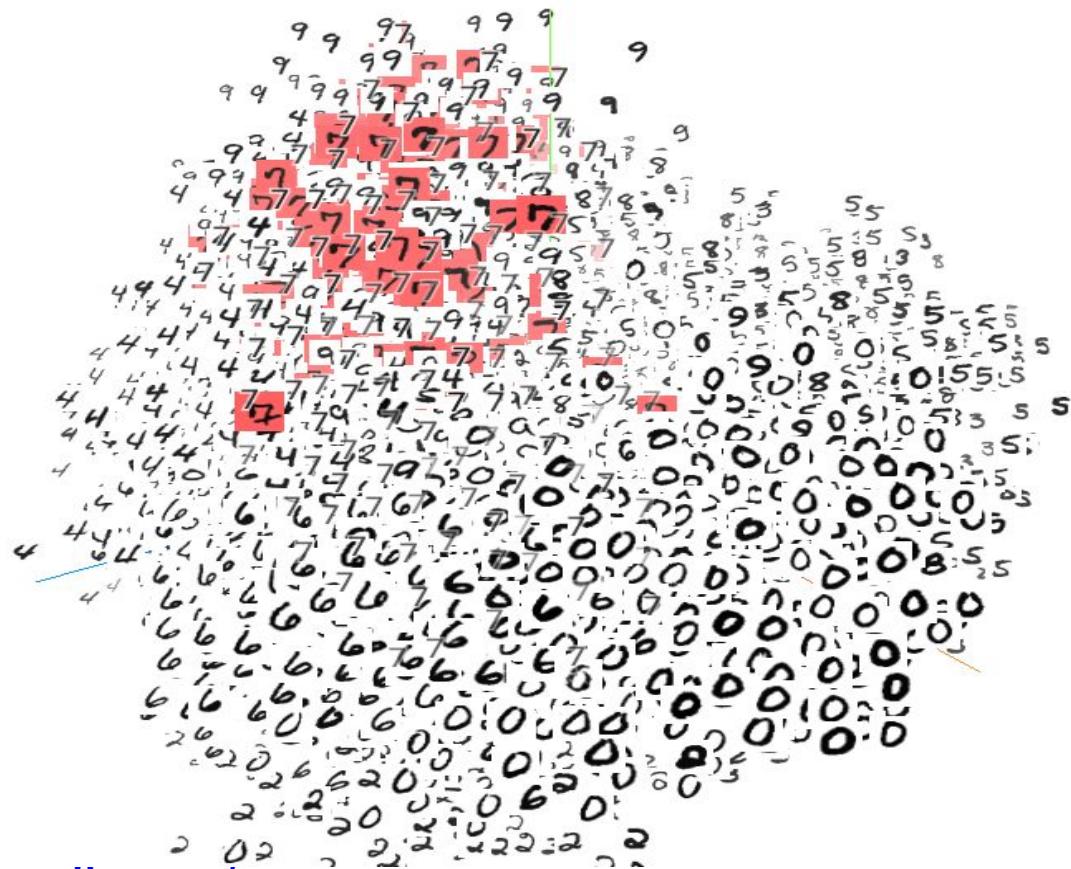
distance COSINE EUCLIDEAN

Nearest points in the original space:

cats	0.470
dog	0.575
dogs	0.604
pet	0.616
mouse	0.637
lovers	0.638
breeds	0.660
breed	0.683
creature	0.687
black	0.690
big	0.690
animal	0.693
hat	0.698
toy	0.698
walking	0.702
sleeping	0.718
sheep	0.724
cow	0.725

# Vector Embeddings (MNIST)

?

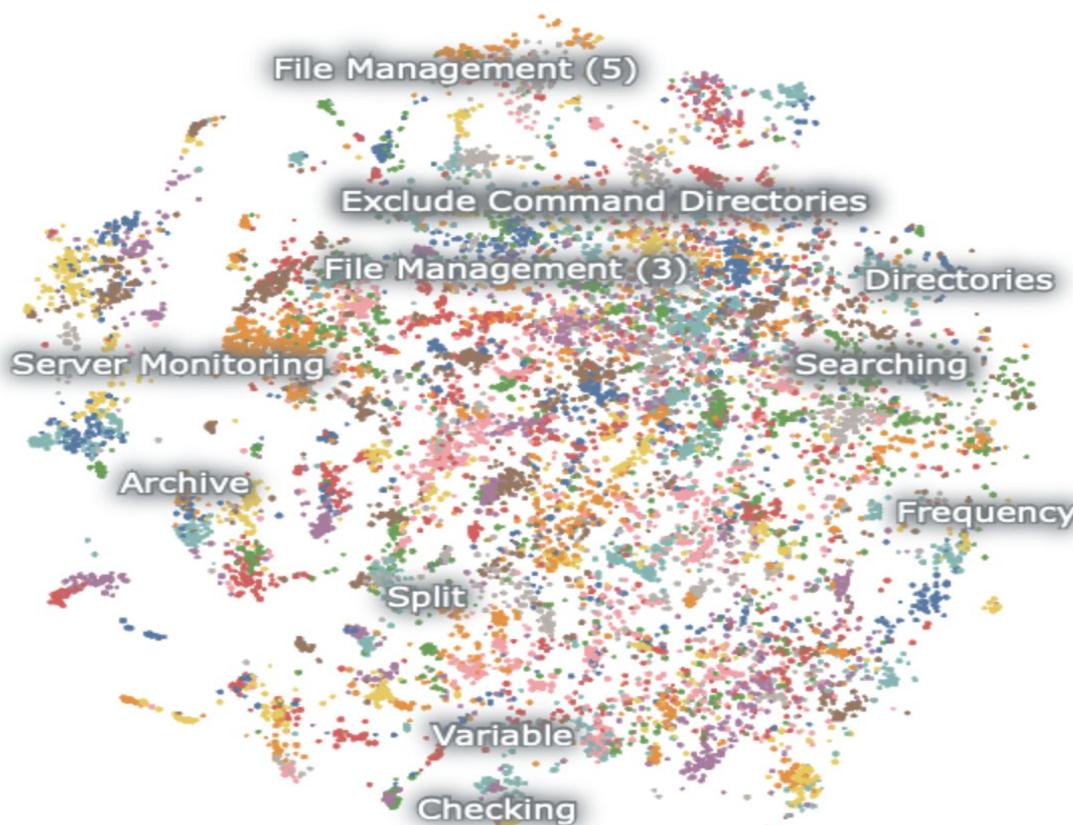


# Vector Embeddings (Arize)



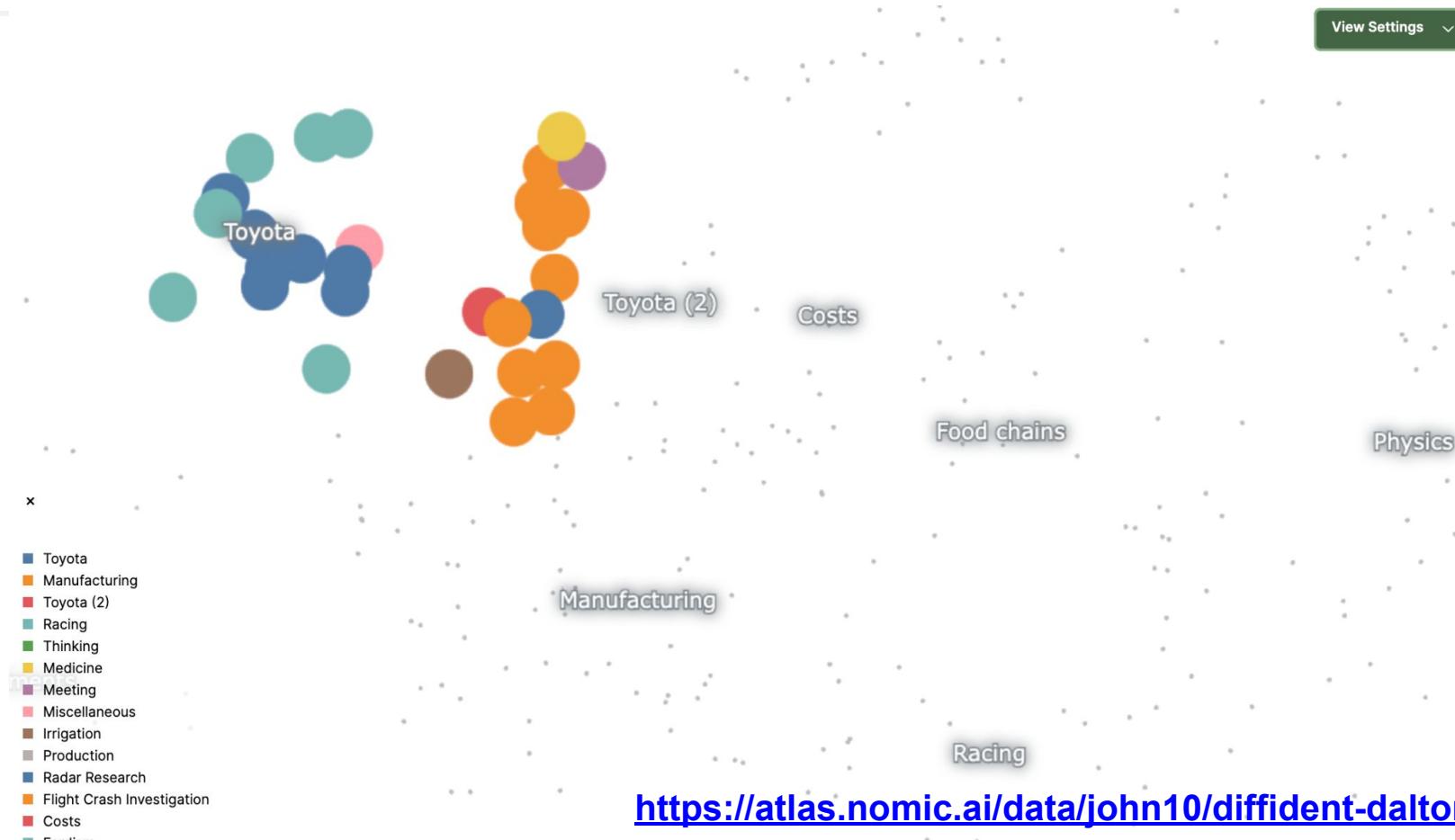
<https://phoenix-demo.arize.com/model>

# Understanding Unstructured Data



# Understanding Unstructured Data

[View Settings](#) ▾

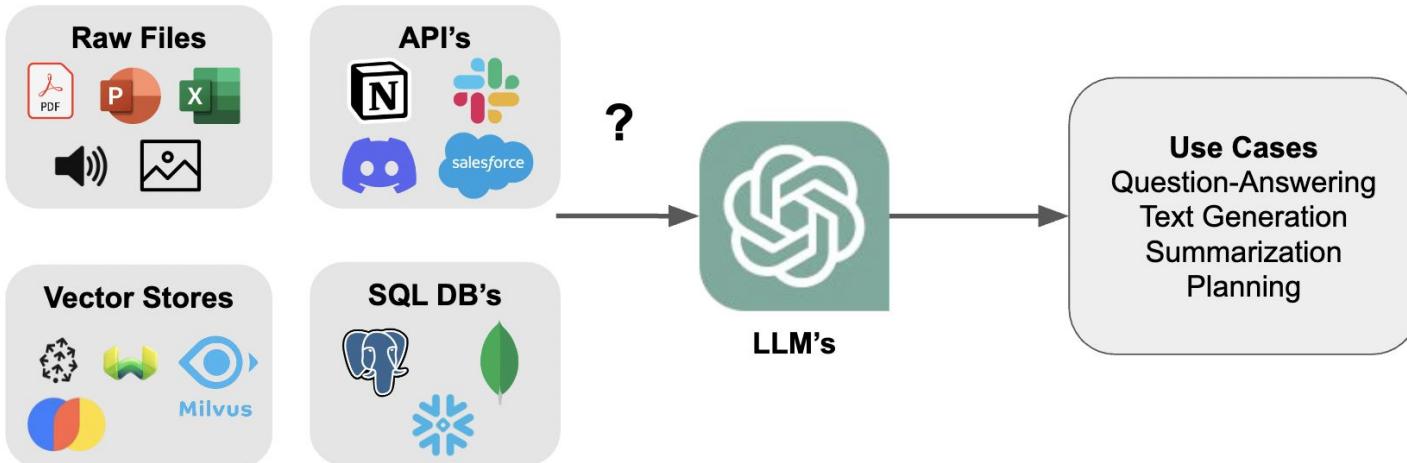


<https://atlas.nomic.ai/data/john10/diffident-dalton/map>

# Retrieval Augmentation Generation (RAG)

## Context

- How do we best augment LLMs with our own **private** data?



# Surprise: It's easier than expected to get going

## Perplexity.ai – Proprietary Service

critical analysis of Google "We Have No Moat, And Neither Does OpenAI"  
Leaked Internal Google Document Claims Open Source AI Will Outcompete Google and OpenAI

\* Copilot  
Q Searching web  
critical analysis Google "We Have No Moat, And Neither Does OpenAI" leaked internal document  
OpenAI outcompetes Google leaked internal document analysis  
Considering 15 sources  
Sources  
Google "We Have No Moat, And Neither Does OpenAI" - 1  
Leaked Google document reveals open source AI - 2  
Google "We have no moat, and neither does..." - 3  
Leaked internal Google document claims OpenAI - 4  
Leaked Google Memo: "We Have No Moat, and..." - 5  
Marketing Institute - 7  
View 8 more

Answer  
A leaked internal Google document claims that open-source AI models are advancing rapidly and may outpace both Google and OpenAI<sup>1</sup>. The document suggests that Google should embrace open-source AI, prioritize third-party integrations, and reconsider the focus on large models<sup>2</sup>. The anonymous Google researcher who leaked the document believes that neither Google nor OpenAI is positioned to win the AI "arms race" due to the rapid advancements made by open-source AI projects<sup>3</sup>.

One of the key factors contributing to the success of open-source AI projects is Low-Rank Adaptation (LoRA), a technique that allows model fine-tuning at a fraction of the cost and time by reducing the size of update matrices<sup>4</sup>. The document argues that Google and OpenAI do not have a "secret sauce" that makes their approaches better than what the wider community can come up with<sup>5</sup>. The researcher suggests that Google should establish itself as a leader in the open-source community, cooperate with the broader conversation, and potentially publish model weights for small ULMoL variants<sup>6</sup>.

The implications of this leaked document could significantly impact Google and OpenAI's future business strategies, as both companies may need to reevaluate their approach to AI research and development in light of the rapid advancements made by open-source AI projects<sup>7</sup>.

Learned how to re-create in 1.5 days  
(with python examples from Github, LangChain, GPT API)

## Critical Analysis of Leaked Google Document: "We Have No Moat, And Neither Does OpenAI"

The contemporary field of artificial intelligence (AI) has been stirred by the recent leak of an internal Google document penned by a Google researcher. The memo, while not an official statement from the company, suggests a drastic shift in the AI landscape, where the previously open-source nature of the document implies that Google and its AI industry competitor, OpenAI, have no "moat" in the AI technology domain, referring to the lack of a unique and constant competitive edge. This report provides a rigorous examination of the implications raised by the leaked document, the responses thus far, and the broader landscape of the AI industry.

### The Clarion Call: No Moat in AI

The leaked memo alludes to the rapid advancements of open-source AI projects, outpacing proprietary models developed by tech giants like Google and OpenAI. The document also insinuates that the "quality gap" between open and closed models is narrowing, with performance as well as innovation, privacy, and capability have been spotlighted as areas where open-source projects are excelling. Tangible exemplifications of this have been noted with models like Meta's LLaMA language model and Vicuna-13B, an open-source chatbot. Both models represent significant AI milestones and challenge more established players. However, the text provides no quantifiable data to support these assertions, leaving the actual degree of this impact open to interpretation.

### Response from Google and OpenAI

Contrary, Demis Hassabis, CEO of Google's DeepMind, authenticated the leaked memo but disagreed with its conclusions. Hassabis appeared optimistic about Google's stand in the AI industry, pointing to the company's innovative researchers and the potential for progress through collaborations between Google Brain and Google DeepMind teams. No specific counter-arguments, numerical data, or formal responses were provided to refute the contents of the memo. Also, there has been no formal response from OpenAI, thereby adding a level of ambiguity to the situation.

### Open-source AI Boon and Bane

Interestingly, open-source AI is represented as both an opportunity and a challenge. The mounting momentum of open-source AI projects, with breakthroughs like the Low Rank Adaptation technique, and reduced entry barriers have emphasized the democratizing potential of open-source AI. Yet, there are also concerns about potential EU regulations on open-source models, which could impede the release of open-source AI in Europe. Here, the text lacks specifics on what these regulations could entail and how they might shift the AI dynamic.

On the flip side, while the open-source AI revolution poses potential threats to AI stalwarts like Google and OpenAI, it simultaneously prompts the necessity for these organizations to reassess their competitive strategies. The documents

stress the need to embrace collaborative measures, learning from the innovations in the thriving open-source community, and staying open to external collaborations to retain their relevance in the AI market.

### Beyond Openness: A Matter of Dominance

The Wired.com article brings a broader perspective to this discourse by highlighting the challenges of true openness in AI. The researchers argue that ownership of software frameworks, the high cost of computing power, secrecy around training data, and the lack of available expertise to fine-tune models concentrate power among large corporations. In this context, the open AI model may consolidate authority rather than lead to democratization. Moreover, the proposed measures require strict adherence to technological controls set by large corporations, especially in sensitive domains like healthcare, finance, education, and the workplace. Here again, further context or specifics could provide a more comprehensive understanding of these challenges.

### Conclusion

The document leak may be seen as a hypothesis on the future trajectory of the AI race, highlighting the example of AI ethics and not just as infrastructural or economic. However, the underlying analysis remains somewhat speculative in the absence of concrete statistical data to support these assertions. Therefore, while it provokes thoughtful discourse on the future of AI, the suggestions of this document should be considered as one of the many perspectives in the dynamically evolving AI landscape.

### ### References

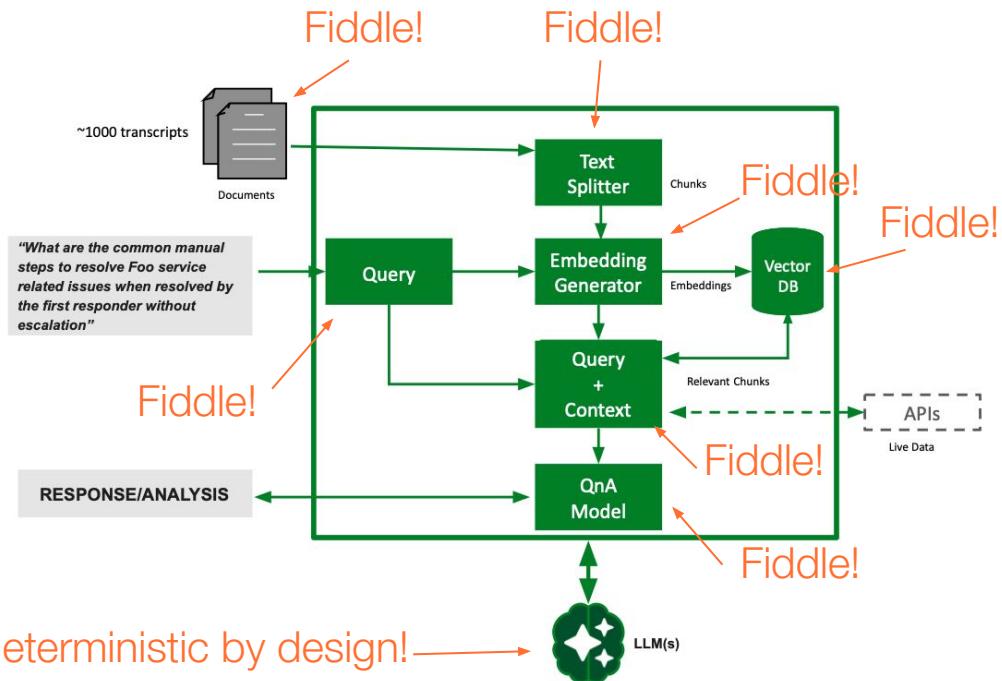
1. SemiAnalysis - Google "We Have No Moat, And Neither Does OpenAI"
2. Artisania - Google Google Memo Claiming "We Have No Moat, And Neither Does OpenAI" Still Circulates
3. LinkedIn - Google "We Have No Moat, And Neither Does Open AI"
4. The Guardian - Google Engineer Open-Source Technology AI OpenAI
5. UX Design Bootcamp - Seizing The Open Source AI Opportunity
6. CMSWire - The Open Source Revolution Challenging AI Giants
7. The Verge - Google Memo Moat AI Leak
8. Deepchecks - OpenAI Google AI Data Science
9. DeepChecks - OpenAI's ChatGPT vs. Google's Bard AI: A Comparative Analysis
10. Wired - The Myth of Open Source AI



Joseph Enochs  
EVT Technologies

# Challenge: Getting good and scaling is hard

Success!!  
("now nobody touch anything")



CACE (*Changing Anything Changes Everything*)

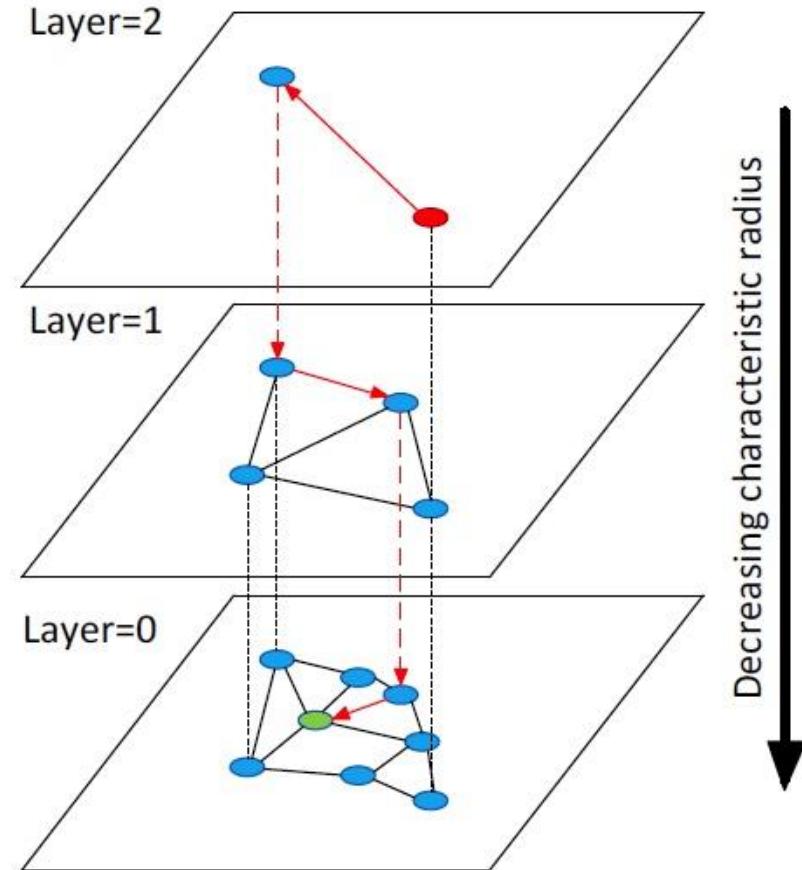
Supporting tools based on old ML technology

Uncontrolled data and config dependencies

Monitoring non-deterministic systems

# Vector Search (HNSW) Hierarchical Navigable Small World Graphs

- Most Vector Databases are powered by a graph-based algorithm called HNSW.
- The queries are called ANN (Approximate K nearest neighbors)
- They provide low-latency search and high-recall results.



# Vector Database Structure

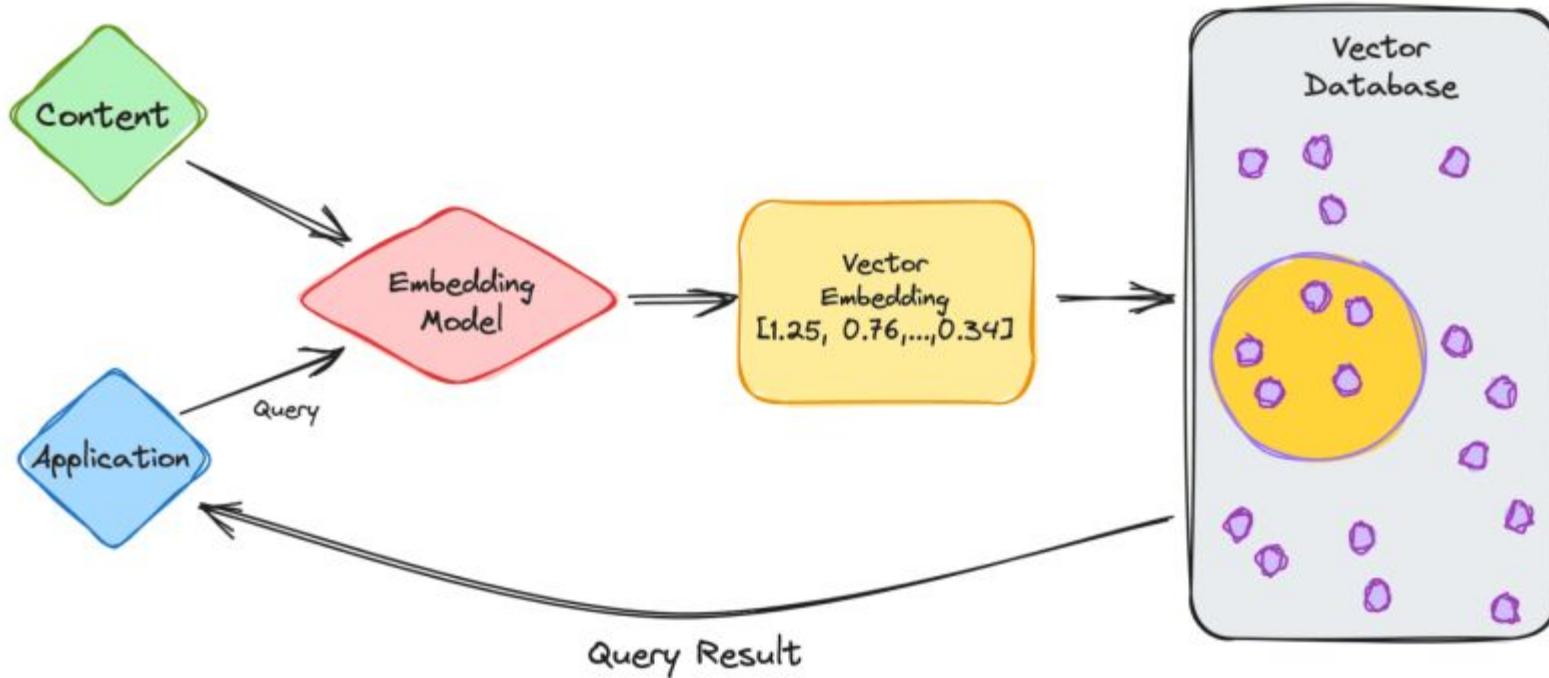
The screenshot shows the MongoDB Atlas interface for a project named "Project 0". The main navigation bar includes "Atlas", "John's Org - ...", "Access Manager", "Billing", "All Clusters", "Get Help", and "John". The left sidebar has sections for "Overview", "Deployment", "Database" (selected), "Data Lake", "Services", "Device Sync", "Triggers", "Data API", "Data Federation", "Atlas Search", "Stream Processing", "Migration", and "Security". The "Data Services" tab is selected. The main content area shows "JOHN'S ORG - 2024-01-17 > PROJECT 0 > DATABASES". A cluster named "Cluster0" is selected, with tabs for "Overview", "Real Time", "Metrics", "Collections" (selected), "Atlas Search", "Profiler", "Performance Advisor", "Online Archive", and "Cmd Line Tools". Below these tabs, it says "DATABASES: 2 COLLECTIONS: 2". There are buttons for "+ Create Database" and "Search Namespaces". Under "Cluster0", there is a section for "OpenContext0" which contains "sample\_mflix". The "Collections" tab shows the "Cluster0.OpenContext0" collection with metrics: STORAGE SIZE: 8.46MB, LOGICAL DATA SIZE: 534.6KB, TOTAL DOCUMENTS: 75, INDEXES TOTAL SIZE: 40KB. It has tabs for "Find", "Indexes", "Schema Anti-Patterns (0)", "Aggregation", and "Search Indexes". A "Filter" input field contains "Type a query: { field: 'value' }". Buttons for "Reset", "Apply", and "Options" are available. Below the filter, a document preview shows:

```
_id: ObjectId('65e0a652d5320559edac2c11')
text: "# Entity Reference
    Entities commonly have a need to reference other ..."
embedding: Array (512)
Header 1: "Entity Reference"
source: "entity-reference.md"
```

At the bottom, another document preview shows:

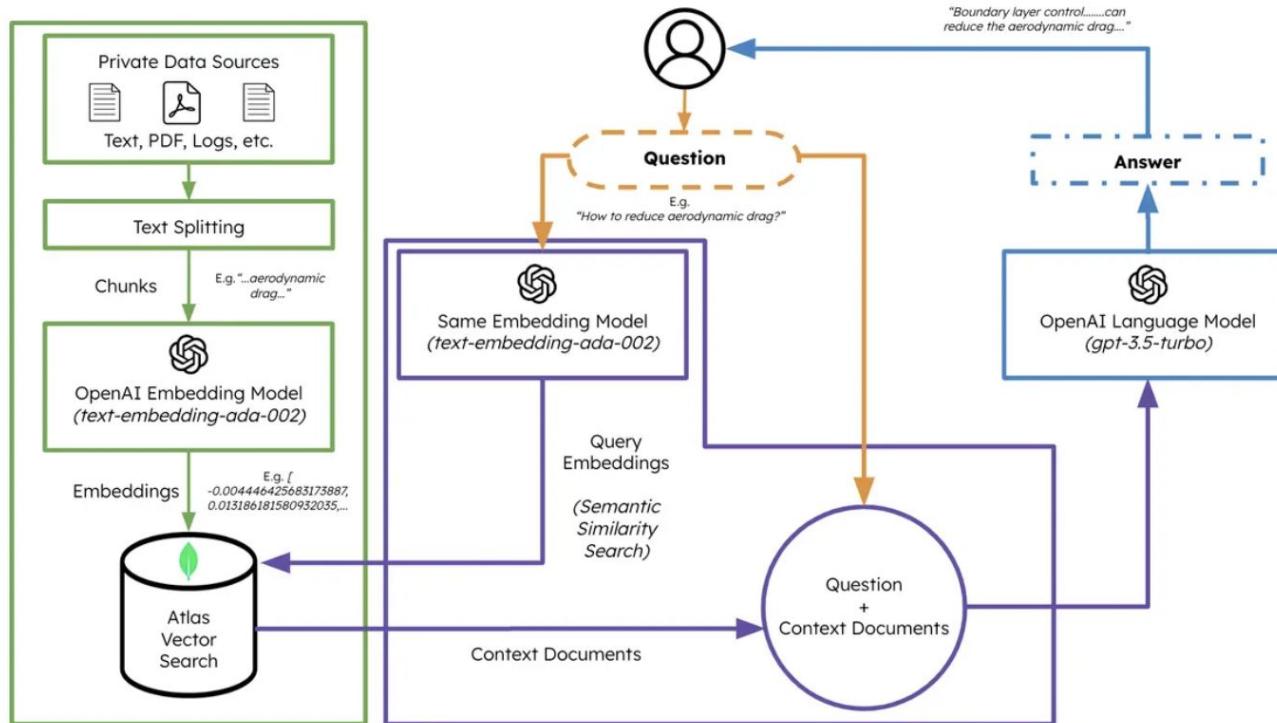
```
_id: ObjectId('65e0a652d5320559edac2c12')
text: "## **String References**"
```

# Retrieval Augmentation - Vectors



# RAG (Vector Databases)

Here's a high-level diagram of the chatbot's RAG architecture:



# Chunking Strategies - Fixed-Size Chunking

```
1  text = "..." # your text
2  from langchain.text_splitter import CharacterTextSplitter
3  text_splitter = CharacterTextSplitter(
4      separator = "\n\n",
5      chunk_size = 256,
6      chunk_overlap = 20
7  )
8  docs = text_splitter.create_documents([text])
```

# Chunking Strategies - Recursive Chunking

```
1 text = "..." # your text
2 from langchain.text_splitter import RecursiveCharacterTextSplitter
3 text_splitter = RecursiveCharacterTextSplitter(
4     # Set a really small chunk size, just to show.
5     chunk_size = 256,
6     chunk_overlap = 20
7 )
8
9 docs = text_splitter.create_documents([text])
```

# Chunking Strategies - Markdown

```
1 from langchain.text_splitter import MarkdownTextSplitter
2 markdown_text = "..."
3
4 markdown_splitter = MarkdownTextSplitter(chunk_size=100, chunk_overlap=0)
5 docs = markdown_splitter.create_documents([markdown_text])
```

# Retrieval Augmentation - Opportunities

---

- Hallucinations
- QA Correctness
- Relevance
- Drift
- Bias
- Toxicity

# Observability

---

- Arize
- Langsmith
- WhyLabs
- TruEra

# Observability - Evaluations

tracing

Total Traces 3 Total Tokens 6,414 Latency P50 ⏱ 1.75s Latency P99 ⏱ 1.75s Hallucination 0.67 QA Correctness 0.33 Relevance **ndcg 1.00** precision 0.25 hit rate 1.00 Stream **Stream**

Traces Spans

filter condition (e.g., span\_kind == 'LLM')

kind	name	input	output	evaluations	status
chain	RetrievalQA	Who is Rose Cihlar	{"result": "Rose Cihlar was a worker who worked as an assembly line inspector at a factory near Hawt..."}	Hallucination fact QA Correctness	✓
chain	RetrievalQA	Who is Honoré Blanc	{"result": "There is no information available about Honor\u00e9 Blanc in the given context.", "sourc..."}	Hallucination hallucinated QA Correctness incorrect	3/5/2024, 11:42 AM ⏱ 1.26s ⚡ 1826 ✓
chain	RetrievalQA	What is red violin story	{"result": "The Red Violin is a movie that tells the story of a unique violin and its journey through..."}	Hallucination hallucinated QA Correctness incorrect	3/5/2024, 11:42 AM ⏱ 2.69s ⚡ 2371 ✓

Normalized Discounted Cumulative Gain). It measures the quality of a ranked list of items by considering both the relevance and the position of each item in the list.

# Observability - Arize Phoenix

tracing

Total Traces 9 Total Tokens 9,866 Latency P50 ⏱ 4.49s Latency P99 ⏱ 12.97s Hallucination 0.33 QA Correctness 0.67 Relevance ndcg 0.94 precision 0.56 hit rate 1.00 Stream

Traces Spans

filter condition (e.g. span\_kind == 'LLM')

Columns

> kind	name	input	output	evaluations	start time	latency	total tokens
> chain	RetrievalQA	list the yaml example crates-erp	{"result": "Here is the YAML example for the 'crates-erp' component:\n\n```yaml\napiVersion: opencon..."}	Hallucination factual QA Correctness correct	3/1/2024, 11:35 AM	⌚ 12.98s	⌚ 2106
> chain	RetrievalQA	List yaml examples names	{"result": "The YAML examples provided are named as follows:\n\n1. Platform: bluesky\n2. Platform: ..."}	Hallucination factual QA Correctness correct	3/1/2024, 11:35 AM	⌚ 4.52s	⌚ 1175
> chain	RetrievalQA	List code examples	{"result": "Here are some code examples:\n\n1. Crates ERP:\n\n```yaml\napiVersion: opencontext.com/v1a..."}	Hallucination factual QA Correctness correct	3/1/2024, 11:35 AM	⌚ 17.08s	⌚ 2325
> chain	RetrievalQA	List all the entities	{"result": "I'm sorry, but I don't have access to the specific entities in your context. Can you pro..."}	Hallucination hallucinated QA Correctness incorrect	3/1/2024, 11:35 AM	⌚ 2.85s	⌚ 452
> chain	RetrievalQA	What is a Datacenter	{"result": "A datacenter is a physical or virtual facility that houses computer systems and associat..."}	Hallucination hallucinated QA NOT_PARSABLE Correctness	3/1/2024, 11:35 AM	⌚ 2.32s	⌚ 1230
> chain	RetrievalQA	What is a Location	{"result": "A Location is a marker that references other places to look for OpenContext data. It is ..."}	Hallucination factual QA Correctness correct	3/1/2024, 11:35 AM	⌚ 2.13s	⌚ 401
> chain	RetrievalQA	What is a Service	{"result": "A Service is a kind of entity that groups together Code Components, Platform Components, ..."}	Hallucination factual QA Correctness correct	3/1/2024, 11:35 AM	⌚ 5.12s	⌚ 700

# Observability - Langsmith Tracing



Personal > Projects > opencontext-1



opencontext-1

Opencontext data



Traces LLM Calls All Runs Monitor Setup



Filters Last 7 days



Name	Input	Start Time	Latency	Dataset	Annotation Queue	Tokens	Cost
RetrievalQA	list the yaml example ...	3/1/2024, 11:45:00 AM	12.97s	CSV	Queue	2,106	\$0.001351
RetrievalQA	List yaml examples na...	3/1/2024, 11:44:55 AM	4.34s	CSV	Queue	1,175	\$0.0006725
RetrievalQA	List code examples	3/1/2024, 11:44:37 AM	18.08s	CSV	Queue	2,325	\$0.0016605
RetrievalQA	List all the entities	3/1/2024, 11:44:35 AM	1.83s	CSV	Queue	452	\$0.00026
RetrievalQA	What is a Datacenter	3/1/2024, 11:44:31 AM	4.67s	CSV	Queue	1,230	\$0.000715
RetrievalQA	What is a Location	3/1/2024, 11:44:27 AM	3.92s	CSV	Queue	401	\$0.0002685
RetrievalQA	What is a Service	3/1/2024, 11:44:24 AM	2.38s	CSV	Queue	700	\$0.000445
RetrievalQA	What is a SaaS User	3/1/2024, 11:44:20 AM	4.76s	CSV	Queue	578	\$0.000377
RetrievalQA	What is a CodeComprn...	3/1/2024, 11:44:14 AM	5.51s	CSV	Queue	899	\$0.0005225

Load more ↓

## Details

RUN COUNT

9

TOTAL TOKENS

9,866 / \$0.006272 ⓘ

MEDIAN TOKENS

899

ERROR RATE

0%

% STREAMING

0%

LATENCY

P50: 4.67s P99: 17.67s

## Filter Shortcuts

Name

RetrievalQA

Run Type

Chain

Status

Success

Other

# Retrieval Augmentation - Advanced RAG



Post



Bindu Reddy

@bindureddy

...

Good paper by Netflix on cosine similarity.

It goes back to building good RAG systems, which is hard. Before deploying these systems, you have to make intelligent decisions about chunking, hierarchical chunking, embedding, and even the algorithm for similarity look-up.

Failure modes will be high and accuracy low if you don't use the appropriate techniques.

<https://arxiv.org/pdf/2403.05440.pdf>

# Retrieval Augmentation - Advanced Techniques

---

- Metadata, Metadata, Metadata
- Summaries in Chunks
- Embedding Optimization
- Evaluations
- LLM as a Judge
- Red Teaming
- Small-to-Big Retrieval - Additional Data in Chunks (Answers)
- Synthetic Data Generation
- Query Types
- Reranking
- E2E Evaluations

# Vector Database Structure

- **Chunks**
  - Chunks in a vector database index are partitions of the dataset, designed to improve the efficiency, scalability, and performance of operations within the database.
- **Tokens**
  - Tokens in an LLM represent the fundamental units into which text is divided for processing.
- **Dimensions**
  - A dimension in a vector context refers to one of its elements, and the total number of dimensions a vector has indicates the space within which it exists.
- **Metadata**
  - Metadata in a vector database refers to data that provides information about other data stored within the database, specifically regarding the vectors.
- **Indexes**
  - An index in a vector database is a structured representation of the data points' vectors, designed to optimize the retrieval of similar data points or objects.

# Retrieval Augmentation - Advanced RAG

## Red Teaming Language Models with Language Models

**WARNING:** This paper contains model outputs which are offensive in nature.

Ethan Perez<sup>1 2</sup> Saffron Huang<sup>1</sup> Francis Song<sup>1</sup> Trevor Cai<sup>1</sup> Roman Ring<sup>1</sup>

John Aslanides<sup>1</sup> Amelia Glaese<sup>1</sup> Nat McAleese<sup>1</sup> Geoffrey Irving<sup>1</sup>

<sup>1</sup>DeepMind, <sup>2</sup>New York University

perez@nyu.edu

### Abstract

Language Models (LMs) often cannot be deployed because of their potential to harm users in hard-to-predict ways. Prior work identifies harmful behaviors before deployment by using human annotators to hand-write test cases. However, human annotation is expensive, limiting the number and diversity of test cases. In this work, we automatically find cases where a target LM behaves in a harmful way, by generating test cases (“red teaming”) using another LM. We evaluate the target LM’s replies to generated test questions using a classifier trained to detect offensive content, uncovering tens of thousands of offensive replies in a 280B parameter LM chatbot. We explore several methods, from zero-shot generation to reinforcement learning, for generating test cases with varying levels of diversity and difficulty. Furthermore, we use prompt engineering to control LM-generated test cases to uncover a variety of other harms, automatically finding groups of people that the



Figure 1: Overview: We automatically generate test cases with a language model (LM), reply with the target LM, and find failing test cases using a classifier.

# Retrieval Augmentation - Cognitive Correctness

1	Example	NDCG	Percision	QA Correctness	Hallucination	Response Time	Query
2	Example 1 - PDF Full Book	100	25	67	33	572	Fair
3	Example 2 - PDF Full Book (chunking)	88	42	100	33	593	Good+
4	Example 3 - PDF Part1	100	42	67	33	868	Good
5	Example 4 - PDF Part1 (chunking)	67	42	100	33	615	Best
6	Example 5 - Semantic Chunking	33	17	100	0	420	Poor
7	Example 6 - Semantic Chunking Percent	33	17	100	0	469	Fair
8	Example 7 - Unstructured MD	33	25	100	33	567	Best-
9	Example 8 - Markdown Reader	33	8	67	33	483	Good
10							

# RAGS (Vector Databases)

---

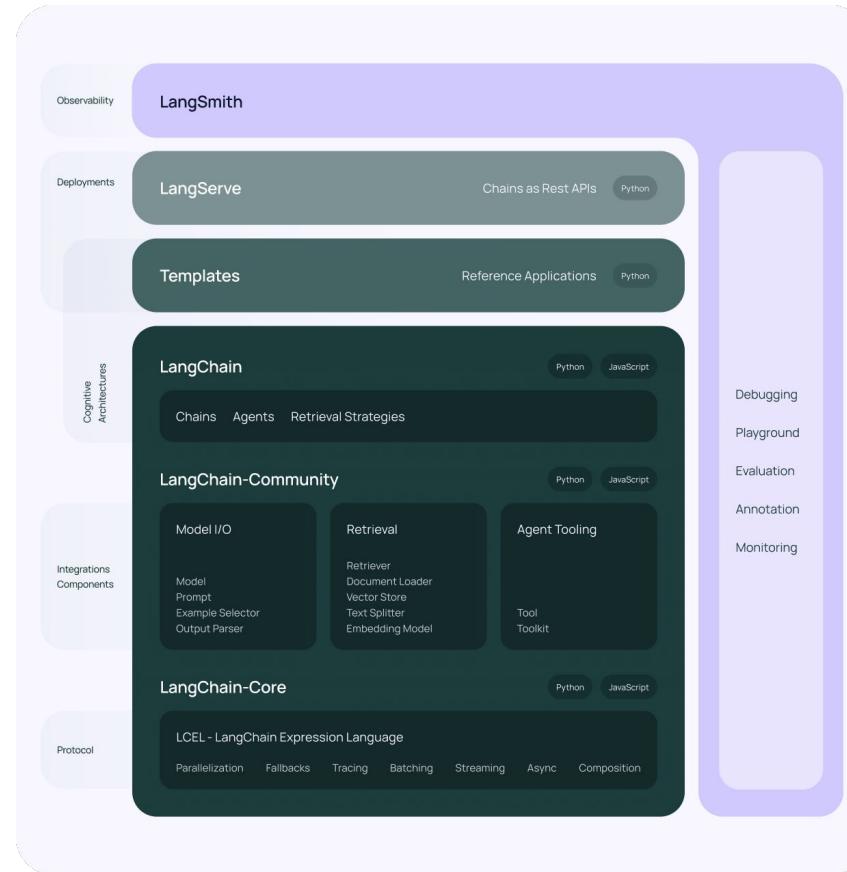
- MongoDB
- ChomaDB
- Milvus
- Pinecone
- Weaviate
- Qdrant
- Vespa
- ElasticSearch

# Orchestration

---

- Langchain
- Llamaindex

# Orchestration - Langchain



# Query Chain Types

---

- Stuff
- Refine
- Map\_Reduce
- Map\_Rerank

```
8 ## STUFF
9
10 The book is about W. Edwards Deming, a relatively quiet voice in one of this century's
    biggest advancements, and his System of Profound Knowledge. It connects the unique
    moments in Deming's life that culminated in his grand unifying theory of management
    that is the predictor of success or failure in every organization today. It was
    written by John Willis during the COVID-19 pandemic when he had extra time due to
    travel restrictions.
```

```
11
12 ## Refine
13 The Phoenix Project is a modern retelling of the bestselling business book The Goal,
    written by Michael Lewis and published when W. Edwards Deming was ninety-three years
    old, just before his death. It focuses on software development and delivery, and
    includes DevOps in the subtitle. The DevOps Handbook, published in 2016 by Gene Kim,
    Patrick Debois, Jez Humble, and the speaker, provides a prescriptive solution to the
    topics discussed in The Phoenix Project. The two books have sold almost a million
    copies combined and contain important ideas about the right quality and uniformity
    being the foundation of commerce, prosperity, and peace. The book also provides a
    portrait of the hard-working, humble man from Wyoming who helped revolutionize
    manufacturing in Japan and around the world, and how his ideas were developed through
    his life experiences which culminated in the System of Profound Knowledge - a grand
    unifying theory of management. During the COVID-19 pandemic, the author was able to
    utilize the extra time to write and connect the unique moments in Deming's life to his
    Profound Knowledge.
```

```
14
15 ## Map_reduce
16 This book is a portrait of W. Edwards Deming and John Willis, two soft-spoken experts
    in this century's biggest advancements. It captures the essence of how to improve most
    any business, with a focus on quality and understanding of the people responsible for
    that quality. It also gives a portrait of the hard-working, humble man from Wyoming
    who helped revolutionize manufacturing in Japan and around the world.
```

```
17
18 ## Chunks (chunk_size=1000, chunk_overlap=200)
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

Untitled

Automatically reload preview

Preview HTML



## STUFF

The book is about W. Edwards Deming, a relatively quiet voice in one of this century's biggest advancements, and his System of Profound Knowledge. It connects the unique moments in Deming's life that culminated in his grand unifying theory of management that is the predictor of success or failure in every organization today. It was written by John Willis during the COVID-19 pandemic when he had extra time due to travel restrictions.

## Refine

The Phoenix Project is a modern retelling of the bestselling business book The Goal, written by Michael Lewis and published when W. Edwards Deming was ninety-three years old, just before his death. It focuses on software development and delivery, and includes DevOps in the subtitle. The DevOps Handbook, published in 2016 by Gene Kim, Patrick Debois, Jez Humble, and the speaker, provides a prescriptive solution to the topics discussed in The Phoenix Project. The two books have sold almost a million copies combined and contain important ideas about the right quality and uniformity being the foundation of commerce, prosperity, and peace. The book also provides a portrait of the hard-working, humble man from Wyoming who helped revolutionize manufacturing in Japan and around the world, and how his ideas were developed through his life experiences which culminated in the System of Profound Knowledge - a grand unifying theory of management. During the COVID-19 pandemic, the author was able to utilize the extra time to write and connect the unique moments in Deming's life to his Profound Knowledge.

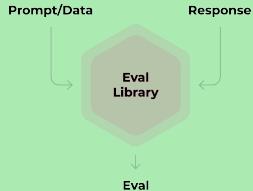
## Map\_reduce

This book is a portrait of W. Edwards Deming and John Willis, two soft-spoken experts in this century's biggest advancements. It captures the essence of how to improve most any business, with a focus on quality and understanding of the people responsible for that quality. It also gives a portrait of the hard-working, humble man from Wyoming who helped revolutionize manufacturing in Japan and around the world.

# 5 Pillars of LLM Observability

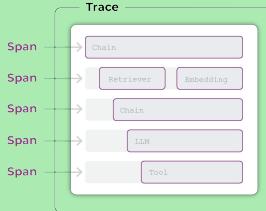
## Evaluation

Evaluations of LLM outputs by using a separate evaluation LLM



## LLM Traces & Spans

Visibility into where workflow broke in LangChain & LlamaIndex



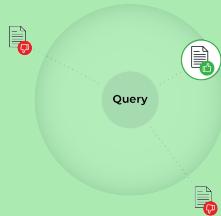
## Prompt Analysis & Troubleshooting

Iterating on prompt templates with production data



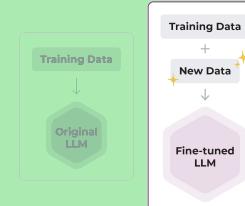
## Search & Retrieval (RAG)

Locate and improve retrieved context



## Fine-tuning

Re-train LLM on use case / company data



# Demo



<https://github.com/OperationalizingAI/DevOpsGAIWorkshop>



botchagalupe@gmail.com

# Setup Gotcha

---

- IP White List
- Password in the MongoDB URI
- Setting up an Index

# Demo

---

- Hugging Face
  - Netflix Movies
  - What's Cooking
- Arize
  - Tracing
  - Evaluations

# Retrieval Augmentation - Hugging Face

The screenshot shows the Hugging Face dataset viewer for the 'ProfoundDeming' dataset. At the top, there's a navigation bar with icons for Hugging Face, search, models, datasets, spaces, posts, docs, solutions, and pricing. Below the header, the dataset title 'Datasets: botchagalupe/ProfoundDeming' is displayed, along with a 'like' button and a count of 0. A license notice for 'cc-by-nd-4.0' is also present.

The main interface features a 'Dataset card' tab, 'Files and versions', 'Community', and 'Settings'. The 'Dataset Viewer' section shows a split view: the left side displays the raw text data, and the right side shows the auto-converted Parquet file. The dataset contains one split, 'train', with 669 rows. The first few rows of the 'text' column are:

- Deming's Journey to Profound Knowledge - How Deming Helped Win a War, Altered the Face of Industry, and Holds the Key to Our Future - Preface
- I pulled on a thread and found a fascinating tapestry.
- My professional career started in 1980, just as New York was coming out of one of the worst financial times since the Great Depression. The joke was you couldn't get a job with IBM, J.P. Morgan, or Grumman without inheriting it. So, at just nineteen years old, I headed...
- Deming's teachings—deeply resonated with me as I continued my career path. While I had unknowingly learned much of his teachings, my knowledge of Dr. W. Edwards Deming didn't begin until the 2000s. I had started working with best-selling author and award-winning CTO...
- At a DevOps Days conference in 2011, my friend and mentor Ben Rock-
- wood, a pioneer in internet engineering, was running an open discussion on Goldratt. During the discussion, Ben intimated that Goldratt was heavily
- influenced by someone called William Edwards Deming. I didn't know who the

Below the dataset viewer, there are navigation links for 'Previous' and 'Next', and a page number indicator '1 | 2 | 3 | ... | 7 | Next >'. To the right of the dataset viewer, there are sections for 'Downloads last month' (0), 'Use in Datasets library', 'Edit dataset card', and statistics for file size (74.4 kB) and number of rows (669).

<https://huggingface.co/datasets/botchagalupe/ProfoundDeming>

# Netflix Movies Dataset

Hugging Face Search models, datasets, users...

Datasets: MongoDB/embedded\_movies like 4

Languages: English Tags: movies Croissant License: apache-2.0

Dataset card Viewer Files and versions Community

Dataset Viewer Auto-converted to Parquet API View in Dataset Viewer

Split (1)  
train · 1.5k rows

Search this dataset

type string · classes	metacritic null	writers sequence	countries sequence	plot_embedding sequence	runtime int64
2 values	9 97 0			[ 0.00072939653, -0.026834568, 0.013515796, -0.032357525, -0.001295428, 0.022092875, ... ]	6 1.
movie	null	[ "Charles W. Goddard..." ]	[ "USA" ]	[ 0.00072939653, -0.026834568, 0.013515796, -0.032357525, -0.001295428, 0.022092875, ... ]	
movie	null	[ "H.M. Walker (titles)" ]	[ "USA" ]	[ -0.022837115, -0.022941574, 0.014937485, -0.024743473, -0.008167305, 0.019429177, ... ]	
movie	null	[ "Herbert Brenon..." ]	[ "USA" ]	[ 0.00023330493, -0.028511643, 0.014653289, -0.03847482, -0.016243158, 0.049179934, ... ]	
movie	null	[ "Douglas Fairbanks..." ]	[ "USA" ]	[ -0.005927917, -0.033394486, 0.0015323418, -0.037410278, -0.020594146, 0.027185857, ... ]	

< Previous 1 2 3 ... 15 Next >

sample\_mflix.embedded\_movies [https://huggingface.co/datasets/MongoDB/embedded\\_movies](https://huggingface.co/datasets/MongoDB/embedded_movies)

# Netflix Movies Dataset

The screenshot shows the Apache Atlas web interface for managing a dataset named "sample\_mflix".

**Atlas Navigation Bar:** Includes links for "Atlas", "John's Org - ...", "Access Manager", and "Billing".

**Project Selection:** "Project 0" is selected.

**Top Navigation:** "Data Services", "App Services", "Charts", "Collections" (highlighted), "Atlas Search", "Profiler", "Performance Advisor", and "Online A".

**Left Sidebar (DEPLOYMENT):** Includes "Database", "Data Lake", "Cluster0", and "sample\_mflix" (selected).

**sample\_mflix Namespace:** Contains "embedded\_items".

**Collection Overview:** Shows 2 databases and 2 collections. Includes a "+ Create Database" button and a search bar for namespaces.

**Collection Details:** For "sample\_mflix.embedded\_items":

- Storage Size: 29.03MB
- Logical Data Size: 30.74MB
- Total Documents: 1800
- Indexes Total Size: 72KB

**Actions:** Find, Indexes, Schema Anti-Patterns (0), Aggregation, Search Indexes.

**Filter:** A code editor containing the query: `{'title':'Scarface'}`.

**Query Results:** 1-1 OF 1

**Document Structure:**

```
_id: ObjectId('65dd0a5d4c8efbdd1226a226')
  languages: Array (1)
    metacritic: null
  cast: Array (4)
    poster: "https://m.media-amazon.com/images/M/MV5BYmMxZTU2ZDUtM2Y1MS00ZWVmLWJlN2..."
    directors: Array (2)
    imbd: Object
    genres: Array (3)
      plot: "An ambitious and near insanely violent gangster climbs the ladder of s..."
      num_mflix_comments: 1
      fullplot: "Johnny Lovo rises to the head of the bootlegging crime syndicate on th..."
    awards: Object
      runtime: 93
      type: "movie"
      rated: "PASSED"
    writers: Array (8)
    countries: Array (1)
    plot_embedding: Array (1536)
    title: "Scarface"
```

# Restaurant Planner

Spaces MongoDB whatscoking-advisor like 0 Running App Files Comm

## MongoDB's Vector Restaurant planner

Start typing below to see the results. You can search a specific cuisine for you and choose 3 predefined locations.

The radius specify the distance from the start search location. This space uses the dataset called [whatscoking.restaurants](#)

search  
indian

Location  
What location you need?  
 Timesquare Manhattan  Westside Manhattan  Downtown Manhattan

Radius in meters  
500

[Clear](#) [Submit](#)

MongoDB Vector Recommendations

Trade-offs:  
- Lower price range but fewer reviews and lower rating compared to other options.

Reason to Consider:  
- Haru is a budget-friendly option with a variety of Japanese dishes and a casual dining atmosphere, perfect for a quick bite or casual meal.

If you're looking for an alternative option, you may consider:  
\*\*One Thai Chef\*\*  
- Cuisine: Thai  
- Price Range: \$\$  
- Location: 339 West 44 Street, Manhattan, NY 10036  
- Distance: 442.78 meters  
- Features: Dogs Allowed, Outdoor Seating, Takeout, Full Bar, Reservations, Table Service  
- Menu Highlights: Mango Sticky Sweet Rice Dessert, Basil Chicken Lettuce Wraps, Mussels Simmered in a Basil-Coconut Sauce, Mango Sorbet  
- Review Count: 28  
- Rating: 2 Stars

Overall, your decision may be based on your budget, the type of dining experience you prefer, and the specific dishes you are craving.

Map

stars  
2  
2.5  
3.5

<https://huggingface.co/spaces/MongoDB/whatscoking-advisor>

# Evaluations

tracing

Trace Details

Total Traces 9

Trace Status OK Latency 2.85s Evaluations Hallucination hallucinated QA Correctness incorrect

Traces Spans

filter co > kind > chain > chain

The interface displays a trace tree on the left and detailed evaluations on the right. The trace tree shows components like RetrievalQA, Retriever, StuffDocumentsChain, LLMChain, and ChatOpenAI. The detailed evaluations section shows two evaluations for 'Hallucination' and one for 'QA Correctness'. Each evaluation includes a table with columns for name, label, score, and explanation.

name	label	score	explanation
Hallucination	hallucinated	1	The query asks to list all the entities, which implies a request for specific names or types of entities mentioned in the reference text. The reference text discusses the concept of entities within a system called OpenContext, mentioning types such as CodeComponent, Team, and Person entities, and explains how these entities can reference each other. It also provides details on how entities are identified and how to figure out an entity reference based on its catalog URL. However, the answer provided does not list any entities or provide specific information about entities from the reference text. Instead, it states an inability to access specific entities and asks for more information, which does not align with the information available in the reference text. Therefore, the answer does not directly respond to the query based on the information provided in the reference text and introduces an unrelated context of needing more information, which is not suggested by the reference text.
QA Correctness	incorrect	0	The question asks to list all the entities based on the reference text provided. The reference text mentions several types of entities, including CodeComponent, Team, Person, and Platform. It also explains how entities in OpenContext are identified and how to reference them. However, the given answer does not list these entities. Instead, it provides a response that suggests an inability to access specific entities and asks for more information. This indicates that the answer does not address the question's request to list all the entities mentioned in the reference text. Therefore, the answer is not correct in the context of the question.

# Evaluations

tracing

Trace Details

Total Traces 9 | Trace Status OK | Latency 2.85s | Evaluations Hallucination hallucinated, QA Correctness incorrect

Traces Spans

filter co > kind > chain > chain

Retriever

Info Evaluations 0 Attributes Events 0

document

# Entity Reference Entities commonly have a need to reference other entities. For example, a CodeComponent entity may want to declare who its owner is by mentioning a Team or Person entity, and a Person entity may want to declare what Team entities it is a member of. This describes how to write those references in your yaml entity declaration files. Each entity in OpenContext is uniquely identified by the triplet of its [kind, namespace, and name](common). But that's a lot to type out manually, and in a lot of circumstances, both the kind and the namespace are fixed, or possible to deduce, or could have sane default values. Each reference can be expressed in one of two ways: as a compact string, or as a compound reference structure.

```
1 v {  
2   "_id": "65e22c13964628b024b2dde1",  
3   "Header 1": "Entity Reference",  
4   "source": "entity-reference.md"  
5 }
```

Evaluations

Relevance unrelated score 0.00

The question asks to list all the entities, which implies a need for a comprehensive enumeration or description of various entities. The reference text, however, does not provide a list or enumeration of entities. Instead, it discusses how entities can reference other entities and the structure of these references in yaml entity declaration files. While it mentions examples of entities such as CodeComponent, Team, and Person, and discusses the unique identification of entities, it does not provide a list of all entities or a broad categorization that would directly answer the question. Therefore, the reference text is not directly relevant to the question asked.

# Observability - Book Analysis

tracing

Trace Details

Total Traces 3

Trace Status OK

Latency 3.53s

Evaluations Hallucination hallucinated, QA Correctness incorrect

Traces Spans

filter co > kind > chain > chain > chain > chain

chain RetrievalQA 3.53s

chain Retriever

chain StuffDocumentsChain

chain LLMChain

chain ChatOpenAI

chain RetrievalQA 3.53s

Info Evaluations 2 Attributes Events 0

name	label	score	explanation
Hallucination	hallucinated	1	The answer provided describes "The Red Violin" as a movie that tells the story of a unique and valuable violin, known as the Red Violin, and its journey through different time periods and locations, from its creation in the workshop of Antonio Stradivari in the 17th century to its modern-day auction. However, the reference text does not mention any movie named "The Red Violin" or its plot. Instead, the reference text primarily discusses the contributions of Deming to quality control, standardization, and profound knowledge, along with historical context related to industrial America and the Hawthorne Works. Therefore, the answer introduces information not found in the reference text, making it a hallucination.
QA Correctness	incorrect	0	The question asks for the story of the red violin. The provided answer describes the plot of the movie "The Red Violin," which follows the journey of a unique and valuable violin through different time periods and locations, from its creation by Antonio Stradivari to its modern-day auction. The answer correctly captures the essence of the movie's story, including the themes of passion, love, and the enduring power of music. However, the reference text provided does not contain any information about the movie "The Red Violin" or its plot. Instead, the reference text discusses various topics related to quality control, standardization, and the history of manufacturing, with no mention of the red violin story or the movie. Therefore, the given answer does not correctly answer the question based on the reference text provided.

# Evaluations

tracing

Trace Details

Total Traces 9 | Trace Status OK | Latency 2.85s | Evaluations Hallucination hallucinated, QA Correctness incorrect

Traces Spans

filter co > kind > chain > chain

Retriever

Info Evaluations 0 Attributes Events 0

document

# Entity Reference Entities commonly have a need to reference other entities. For example, a CodeComponent entity may want to declare who its owner is by mentioning a Team or Person entity, and a Person entity may want to declare what Team entities it is a member of. This describes how to write those references in your yaml entity declaration files. Each entity in OpenContext is uniquely identified by the triplet of its [kind, namespace, and name](common). But that's a lot to type out manually, and in a lot of circumstances, both the kind and the namespace are fixed, or possible to deduce, or could have sane default values. Each reference can be expressed in one of two ways: as a compact string, or as a compound reference structure.

```
1 v {  
2   "_id": "65e22c13964628b024b2dde1",  
3   "Header 1": "Entity Reference",  
4   "source": "entity-reference.md"  
5 }
```

Evaluations

Relevance unrelated score 0.00

The question asks to list all the entities, which implies a need for a comprehensive enumeration or description of various entities. The reference text, however, does not provide a list or enumeration of entities. Instead, it discusses how entities can reference other entities and the structure of these references in yaml entity declaration files. While it mentions examples of entities such as CodeComponent, Team, and Person, and discusses the unique identification of entities, it does not provide a list of all entities or a broad categorization that would directly answer the question. Therefore, the reference text is not directly relevant to the question asked.

# Resources

- <https://platform.openai.com/docs/guides/embeddings/limitations-risks>
- <https://python.langchain.com/docs/modules/chains/>
- <https://smith.langchain.com/>
- <https://www.mongodb.com/products/platform/atlas-vector-search>
- [https://python.langchain.com/docs/integrations/vectorstores/mongodb\\_atlas](https://python.langchain.com/docs/integrations/vectorstores/mongodb_atlas)
- <https://techstrong.ai/author/john-willis/>

# Labs



# Bonus Material



# Retrieval Augmentation - Chunking

- Langchain
  - Loaders
  - Text Splitting
  - Chunking Strategies



```
text_splitter = RecursiveCharacterTextSplitter(  
    # Set a really small chunk size, just to show.  
    chunk_size=100,  
    chunk_overlap=20,  
    length_function=len,  
    is_separator_regex=False,  
)
```

# Retrieval Augmentation - Chunking

- **Llamaindex**

- Loaders
- Text Splitting
- Chunking Strategies



```
splitter = SentenceSplitter(  
    chunk_size=1024,  
    chunk_overlap=20,  
)  
nodes = splitter.get_nodes_from_documents([documents])
```

```
# Initialize the sentence transformer model for embeddings

query = "SaaS users"
#query = "What is a CodeComponet"

#model = SentenceTransformer('all-MiniLM-L6-v2')
#model = SentenceTransformer('BAAI/bge-small-en-v1.5')
#model = SentenceTransformer('sentence-transformers/all-mpnet-base-v2')
model = SentenceTransformer('nomic-ai/nomic-embed-text-v1',trust_remote_code=True)

#method = "RecursiveCharacterTextSplitter"
method = "MarkdownHeaderTextSplitter"

chunk_size = 1000
chunk_overlap = 25
num_chunks = 50
text = str(md)

df_with_embeddings = update_output(method, text, chunk_size, chunk_overlap, num_chunks, query)
```

# all-mpnet-base-v2

Embedding calculation returned in 56478.096300998004 ms

Similarity search returned in 76.28776899946388 ms

Chunk #	Similarity	Text	Chunk	Character Count	Token Count	Embeddings	Actions
36	36	0.558414	:::caution SaaS users should only use this to ...	608	92	[0.013529987074434757, -0.007926667109131813, ...	  
38	38	0.259418	- **primaryEmail** (string) [required]\nThe em...	994	150	[0.07691776752471924, -0.05859971046447754, -0...	
40	40	0.255744	This kind of entity describes the infrastructu...	289	42	[0.0025577698834240437, -0.07189366221427917, ...	
34	34	0.243376	- **owner** (string array) [required]\nAn arra...	472	63	[0.02094290778040886, -0.03774764761328697, 0....	
46	46	0.242870	- **profile** [optional]\nOptional profile inf...	799	128	[0.067266084253788, -0.04007098823785782, 0.00...	

# BAAI/bge-small-en-v1.

Embedding calculation returned in 33761.18713300093 ms

Similarity search returned in 34.74714100229903 ms

Chunk #	Similarity	Text	Chunk	Character Count	Token Count	Embeddings
36	36	0.763518	:::caution SaaS users should only use this to ...	608	92	[0.05116117745637894, -0.04744413495063782, -0...
44	44	0.636932	A Person describes an individual such as an em...	190	31	[0.021428735926747322, -0.0049504549242556095,...
46	46	0.620110	- **profile** [optional]\nOptional profile inf...	799	128	[-0.010465038008987904, 0.014629191718995571, ...
34	34	0.607270	- **owner** (string array) [required]\nAn arra...	472	63	[-0.030688628554344177, -0.029437243938446045, ...
38	38	0.603923	- **primaryEmail** (string) [required]\nThe em...	994	150	[0.0024873148649930954, 0.00909623596817255, -...

# all-MiniLM-L6-v

Embedding calculation returned in 5621.123635999538 ms

Similarity search returned in 19.96587699977681 ms

Chunk #	Similarity	Text Chunk	Character Count	Token Count	Embeddings
36	36	0.505570 ::caution SaaS users should only use this to ...	608	92	[-0.07595530152320862, -0.047116488218307495, ...]
44	44	0.272336 A Person describes an individual such as an em...	190	31	[-0.020194286480545998, -0.01999472640454769, ...]
40	40	0.231006 This kind of entity describes the infrastructu...	289	42	[0.05279329791665077, -0.08227327466011047, -0...
34	34	0.225216 - **owner** (string array) [required]\nAn arra...	472	63	[0.0203170794993639, -0.028190961107611656, -0...
46	46	0.220922 - **profile** [optional]\nOptional profile inf...	799	128	[-0.06263989955186844, 0.10522248595952988, -0...
38	38	0.215048 - **primaryEmail** (string) [required]\nThe em...	994	150	[-0.06164391338825226, 0.07935108989477158, -0...
42	42	0.204251 - **type** (string) [required]\nThe type of pl...	1533	190	[-0.04651326686143875, -0.029575956985354424, ...]

# nomic-ai/nomic-embed-text-v

Embedding calculation returned in 178716.813731 ms  
Similarity search returned in 108.91081600129837 ms

Chunk #	Similarity	Text	Text Chunk	Character Count	Token Count	Embeddings	Actions
36	36	0.628071	:::caution SaaS users should only use this to ...	608	92	[-0.028383471071720123, 0.042075514793395996, ...]	
46	46	0.453991	- **profile** [optional]\nOptional profile inf...	799	128	[-0.02650478109717369, 0.04022234305739403, -0...]	
15	15	0.450684	```yaml\n# Example\nmetadata:\nannotations:\n...`	78	8	[0.006941637489944696, 0.038732241839170456, -...]	
38	38	0.449998	- **primaryEmail** (string) [required]\nThe em...	994	150	[-0.026276206597685814, 0.04910372570157051, -...]	
34	34	0.445904	- **owner** (string array) [required]\nAn arra...	472	63	[-0.03077189065515995, 0.027758771553635597, 0...	
42	42	0.431073	- **type** (string) [required]\nThe type of pl...	1533	190	[-0.017513280734419823, 0.04951305687427521, 0...	
16	16	0.422538	```yaml\n# Example\nmetadata:\nannotations:\n...`	86	8	[0.027934378013014793, 0.03041982091963291, -...]	
14	14	0.421641	```yaml\n# Example\nmetadata:\nannotations:\n...`	93	8	[0.013335241004824638, 0.022914163768291473, 0...	

# Generative AI

A type of machine learning known as generative AI involves creating new data samples based on trained models. This subset of artificial intelligence can produce various kinds of content, such as text, images, audio and synthetic data. A generative AI model is a subset of a deep learning model that creates new content in response to input. Machine learning algorithms focus on understanding data and making accurate predictions, while generative AI seeks to create new data samples similar to existing ones. The Generative Pre-Trained Transformer (GPT) by OpenAI is widely used in generative AI. LLMs are generally speaking statistical models of textual patterns and transformers for predicting the next word in the text. When GPT sees the word "New," it processes the probability of the next word as follows:

1. "New" car – 20%
2. "New" York – 15%
3. "New" Zealand – 10%
4. "New" spaper – 8%
5. "New" ly – 7%

Here are the probabilities for the next word if it is "York":

1. "New York" City – 50%
2. "New York" Times – 20%
3. "New York" Yankees – 10%
4. "New York" State – 8%
5. "New York" Knicks – 5%

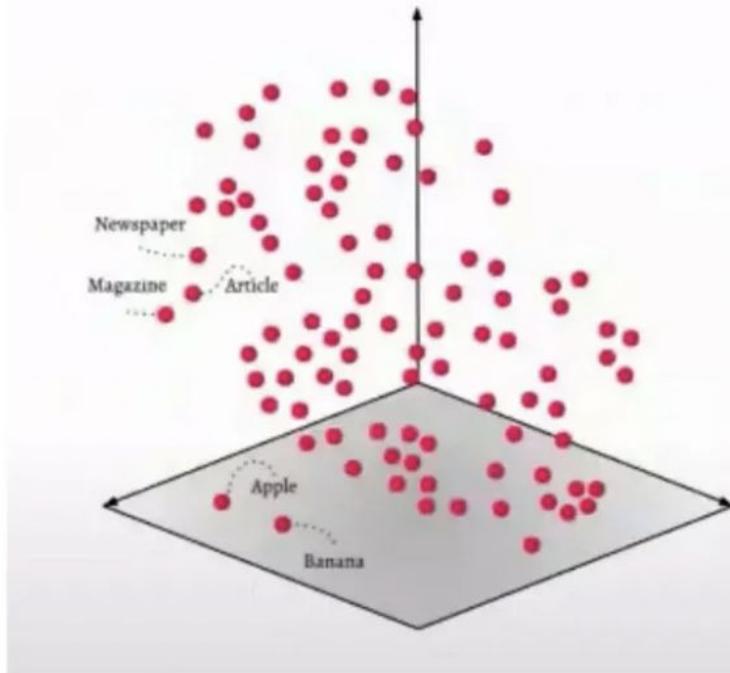
There are considerations when deciding how to configure your index:

- Number of vectors
- Dimensionality of your vectors
- Size of metadata on each vector
- QPS throughput
- Cardinality of indexed metadata

# Vector search in a nutshell

**Vector search** is a way to represent and search your objects (documents, songs, images..) in a geometric space (usually of high-dimension) in the form of an embedding (a vector of numbers: [0.9, -0.1, 0.15, ...])

- At small scale you can apply exact KNN search
- At larger scale you need to use ANN search: trade some precision for speed



Credit: Weaviate V1.0 release - virtual meetup

# Chunking Strategies - Fixed-Size Chunking

```
1  text = "..." # your text
2  from langchain.text_splitter import CharacterTextSplitter
3  text_splitter = CharacterTextSplitter(
4      separator = "\n\n",
5      chunk_size = 256,
6      chunk_overlap = 20
7  )
8  docs = text_splitter.create_documents([text])
```

# Chunking Strategies - Recursive Chunking

```
1 text = "..." # your text
2 from langchain.text_splitter import RecursiveCharacterTextSplitter
3 text_splitter = RecursiveCharacterTextSplitter(
4     # Set a really small chunk size, just to show.
5     chunk_size = 256,
6     chunk_overlap = 20
7 )
8
9 docs = text_splitter.create_documents([text])
```

# Chunking Strategies - Markdown

```
1 from langchain.text_splitter import MarkdownTextSplitter
2 markdown_text = "..."
3
4 markdown_splitter = MarkdownTextSplitter(chunk_size=100, chunk_overlap=0)
5 docs = markdown_splitter.create_documents([markdown_text])
```

```
show_chunks(docs, chunk_size=1000, chunk_overlap=200)
```

## Chunk 1:

page\_content='math and statistics enabled management to see defects as results of the process instead of the workers. Before this, most managers viewed their employees like Stradivari did his apprentices. If the product was bad, it must be the workers' fault. Stradivari never stopped to consider whether the woodcutter had sold him wood from a diseased tree. Never stopped to see if the apprentices' tools were sufficient. Never once considered that he himself might be a poor teacher. His reaction was to blame the worker.' metadata={'source': '/content/qdrive/MyDrive/GAI/transcripts/DEM\_Interior.6.24.23.pdf'}]

## Chunk 2:

page\_content='By 1929, Shewhart had formalized this new method of tracking and tracing variation. Basically, he applied the scientific method to manufacturing. Before this, manufacturing was a linear process. You figured out what you wanted and how many. You made them. Then you inspected and threw away the defects.  
Shewhart turned this into a cycle, what Ed would later call the Shewhart Cycle: Figure out what you want, make it, inspect it, figure out what caused the defects, go fix it, and then go through the whole cycle again, using feedback from your mistakes to continuously improve production quality.  
Even after Deming tweaked the Shewhart cycle, he still referred to it throughout his life as the Shewhart wheel. Despite this, his students in Japan called it the Deming cycle. Today, you might recognize it as the "plan, do, check, act" method, or simply the PDCA cycle.' metadata={'source': '/content/gdrive/MyDrive/GAI/transcripts/DEM\_Interior.6.24.23.pdf'}

### **Chunk 3:**

## **show\_chunks(docs, chunk\_size=500, chunk\_overlap=200)**

### **Chunk 1:**

```
page_content='Thus, Buffalo Bill recognized Edwards and his little brother at the LA\n\nperformance.' metadata={'source': '/content/gdrive/MyDrive/GAI/transcripts/DEM_Interior.6.24.23.pdf'}
```

### **Chunk 2:**

```
page_content='Thus, Buffalo Bill recognized Edwards and his little brother at the LA\n\nperformance.' metadata={'source': '/content/gdrive/MyDrive/GAI/transcripts/DEM_Interior.6.24.23.pdf'}
```

### **Chunk 3:**

```
page_content='Unfortunately for Buffalo Bill, the Cody-based irrigation empire failed. But in 1905, the federal government began a massive public works project via the US Reclamation Service aimed at irrigating ninety thousand acres to turn the semi-arid Bighorn Basin into fertile farms. This necessitated the construction of the Shoshone River Dam twenty-five miles northeast of Cody, around the settlement of Powell. Once completed, the concrete-arch gravity dam—itself a predecessor to the Hoover Dam—was the' metadata={'source': '/content/gdrive/MyDrive/GAI/transcripts/DEM_Interior.6.24.23.pdf'}
```

### **Chunk 4:**

```
page_content='of the Shoshone River Dam twenty-five miles northeast of Cody, around the settlement of Powell. Once completed, the concrete-arch gravity dam—itself a predecessor to the Hoover Dam—was the tallest dam in the world.' metadata={'source': '/content/gdrive/MyDrive/GAI/transcripts/DEM_Interior.6.24.23.pdf'}
```

### **Chunk 5:**

```
page_content='The area around Powell was opened to homesteaders, and, in 1906, Mr. William Deming applied for and received forty acres of farmland on the edge of town. Or at least what everyone hoped would be farmland one day. In the meantime, the Demings eked out what living they could out where the Great Plains meet the Rocky Mountains.' metadata={'source': '/content/gdrive/MyDrive/GAI/transcripts/DEM_Interior.6.24.23.pdf'}
```

# Generative AI Data Strategies (Vector Embeddings Strategies)

## Initialize Embedding Model

We use `text-embedding-ada-002` as the embedding model. We can embed text like so:

```
[]:  
import openai  
  
# initialize openai API key  
openai.api_key = "sk-..." #platform.openai.com  
  
embed_model = "text-embedding-ada-002"  
  
res = openai.Embedding.create(  
    input=[  
        "Sample document text goes here",  
        "there will be several phrases in each batch"  
    ], engine=embed_model  
)
```

# What are embeddings?

OpenAI's text embeddings measure the relatedness of text strings. Embeddings are commonly used for:

- **Search** (where results are ranked by relevance to a query string)
- **Clustering** (where text strings are grouped by similarity)
- **Recommendations** (where items with related text strings are recommended)
- **Anomaly detection** (where outliers with little relatedness are identified)
- **Diversity measurement** (where similarity distributions are analyzed)
- **Classification** (where text strings are classified by their most similar label)

An embedding is a vector (list) of floating point numbers. The **distance** between two vectors measures their relatedness. Small distances suggest high relatedness and large distances suggest low relatedness.



+ Code



```
[ ] prompt = """In a quiet town nestled near the coast, there lived three extraordinary individuals: Jake, a passionate baseball player with a swing that could send a b
```



One clear summer night, while they were honing their skills under a starlit sky, a brilliant streak of light streaked across the heavens, catching their attention.



Curiosity outweighed their fear as they approached the spacecraft. The hatch opened, revealing a small, friendly alien named Zara. With shimmering eyes and a warm s



1. Jake [Baseball Player]

2. Amilea [Artist]

3. Ryan [Golfer]

The football player, known for their agility and strategic thinking, was named Ethan. With lightning-fast reflexes and an innate ability to read the game, Ethan was

The doctor, driven by a passion for healing and dedicated to the well-being of others, was called Dr. Olivia Reed. With her empathetic nature and extensive medical

The carpenter, a master of craftsmanship with an eye for detail, went by the name of Lucas Bennett. Lucas possessed a rare blend of creativity and technical skill,

....

```
[ ] print(openai(prompt))
```

1. Ethan [Football Player]

2. Dr. Olivia Reed [Doctor]

3. Lucas Bennett [Carpenter]



Botchagal...



Access Manager

Billing

All Clusters

Get Help

John

Project 0

Data Services

App Services

Charts



## DEPLOYMENT

### Database

Data Lake

### SERVICES

Device Sync

Triggers

Data API

Data Federation

Search

Stream Processing

### SECURITY

Quickstart

Backup

Database Access

Network Access

Advanced

Goto

DATABASES: 3 COLLECTIONS: 8

VISUALIZE YOUR DATA

REFRESH

+ Create Database

Search Namespaces

db\_docstore

docstore/data

docstore/metadata

docstore/ref\_doc\_info

ac-sv4u64b-shard-00-00.xqonqh.mongodb.net:27017

RESULTS: 1-5 OF 5

```
_id: "3fa66937-89dc-48f5-a313-f81b43c3687f"
  __data__: Object
    id_: "3fa66937-89dc-48f5-a313-f81b43c3687f"
    embedding: null
    metadata: Object
    excluded_embed_metadata_keys: Array
    excluded_llm_metadata_keys: Array
    relationships: Object
      hash: "4c3f2de1ab1735ce87f6f26bf7ac62405bdb1c8cef48778a18aab1ce543ec76f"
      text: "GPT-4 Technical Report
OpenAI
Abstract
We report the development of G..."
      start_char_idx: 0
      end_char_idx: 3433
      text_template: "{metadata_str}"
```

INSERT DOCUMENT

Reset

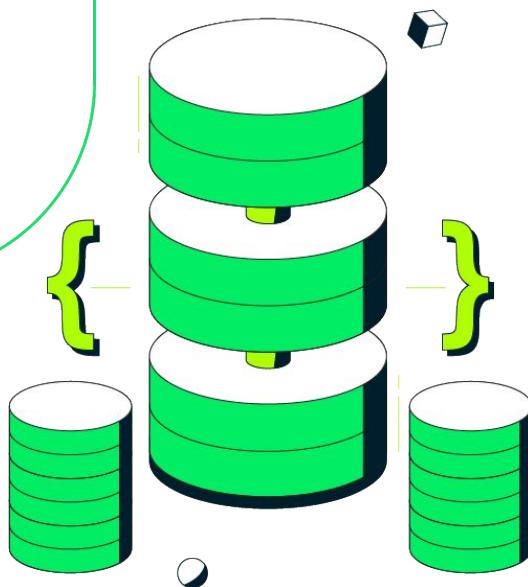
Apply

More Options ▾



# Generative AI Data Strategies (Metadata)

# MongoDB Atlas Vector Search



- Build semantic search directly on top of your operational data in the cloud with **Atlas Vector Search**
- A couple of clicks in the UI or API calls instantly creates a vector search index to power knn queries
- Combines the power of an approximate nearest neighbor query with the productivity, scale, and resilience of MongoDB Atlas in an integrated and fully managed platform

# MongoDB Atlas Vector Database (Load)

atlas-vector-search-rag / load\_data.py 



harshadpd Add files via upload

3fe5c2a · 2 months ago

 Hist

Code

Blame 25 lines (20 loc) · 1.04 KB

```
1  from pymongo import MongoClient
2  from langchain.embeddings.openai import OpenAIEMBEDDINGS
3  from langchain.vectorstores import MongoDBAtlasVectorSearch
4  from langchain.document_loaders import DirectoryLoader
5  from langchain.llms import OpenAI
6  import key_param
7
8  # Set the MongoDB URI, DB, Collection Names
9
10 client = MongoClient(key_param.MONGO_URI)
11 dbName = "langchain_demo"
12 collectionName = "collection_of_text_blobs"
13 collection = client[dbName][collectionName]
14
15 # Initialize the DirectoryLoader
16 loader = DirectoryLoader('./sample_files', glob=".*.txt", show_progress=True)
17 data = loader.load()
18
19 # Define the OpenAI Embedding Model we want to use for the source data
20 # The embedding model is different from the language generation model
21 embeddings = OpenAIEMBEDDINGS(openai_api_key=key_param.openai_api_key)
22
23 # Initialize the VectorStore, and
24 # vectorise the text from the documents using the specified embedding model, and insert them into the specified MongoDB collection
25 vectorStore = MongoDBAtlasVectorSearch.from_documents( data, embeddings, collection=collection )
```

# MongoDB Atlas Vector Database (Extract)

atlas-vector-search-rag / extract\_information.py

[Code](#) [Blame](#) 75 lines (53 loc) · 2.88 KB [Raw](#)

```
23
24     def query_data(query):
25         # Convert question to vector using OpenAI embeddings
26         # Perform Atlas Vector Search using Langchain's vectorStore
27         # similarity_search returns MongoDB documents most similar to the query
28
29         docs = vectorStore.similarity_search(query, K=1)
30         as_output = docs[0].page_content
31
32         # Leveraging Atlas Vector Search paired with Langchain's QARetriever
33
34         # Define the LLM that we want to use -- note that this is the Language Generation Model and NOT an Embedding Model
35         # If it's not specified (for example like in the code below),
36         # then the default OpenAI model used in LangChain is OpenAI GPT-3.5-turbo, as of August 30, 2023
37
38         llm = OpenAI(openai_api_key=key_param.openai_api_key, temperature=0)
39
40
41         # Get VectorStoreRetriever: Specifically, Retriever for MongoDB VectorStore.
42         # Implements _get_relevant_documents which retrieves documents relevant to a query.
43         retriever = vectorStore.as_retriever()
44
45         # Load "stuff" documents chain. Stuff documents chain takes a list of documents,
46         # inserts them all into a prompt and passes that prompt to an LLM.
47
48         qa = RetrievalQA.from_chain_type(llm, chain_type="stuff", retriever=retriever)
49
50         # Execute the chain
51
52         retriever_output = qa.run(query)
53
54         # Return Atlas Vector Search output, and output generated using RAG Architecture
55         return as_output, retriever_output
```

# Generative AI Data Strategies (Query Chain Types)

# Query Chain Types

---

- Stuff
- Refine
- Map\_Reduce
- Map\_Rerank



+ Code + Text

## ▼ 9. Example Usage of the Question-Answering Function

Below is an example of how to use the qa() function to answer a question about HRMC (Deming's Journey to Profound Knowledge) using the different chain types.

```
result = qa("Summarize the book", "stuff", 5)
--NORMAL--
sk-imRgCQmkJjSXkM6eCdR5T3B1bkFJdUtkaq5s0u4jXmqfYDbW
The book is about W. Edwards Deming, a relatively quiet voice in one of this century's biggest advancements, and his System of Pro:
```

```
[ ] result = qa("Summarize the book", "refine", 5)
```

The Phoenix Project is a modern retelling of the bestselling business book The Goal, written by Michael Lewis and published when W.

```
[ ] result = qa("Summarize the book", "map_reduce", 5)
```

This book is a portrait of W. Edwards Deming and John Willis, two soft-spoken experts in this century's biggest advancements. It ca

```
8  
9 ## STUFF  
10 The book is about W. Edwards Deming, a relatively quiet voice in one of this century's  
11 biggest advancements, and his System of Profound Knowledge. It connects the unique  
12 moments in Deming's life that culminated in his grand unifying theory of management  
13 that is the predictor of success or failure in every organization today. It was  
14 written by John Willis during the COVID-19 pandemic when he had extra time due to  
15 travel restrictions.
```

```
16 ## Refine  
17 The Phoenix Project is a modern retelling of the bestselling business book The Goal,  
18 written by Michael Lewis and published when W. Edwards Deming was ninety-three years  
19 old, just before his death. It focuses on software development and delivery, and  
20 includes DevOps in the subtitle. The DevOps Handbook, published in 2016 by Gene Kim,  
21 Patrick Debois, Jez Humble, and the speaker, provides a prescriptive solution to the topics  
22 discussed in The Phoenix Project. The two books have sold almost a million  
23 copies combined and contain important ideas about the right quality and uniformity  
24 being the foundation of commerce, prosperity, and peace. The book also provides a  
25 portrait of the hard-working, humble man from Wyoming who helped revolutionize  
26 manufacturing in Japan and around the world, and how his ideas were developed through  
27 his life experiences which culminated in the System of Profound Knowledge - a grand  
28 unifying theory of management. During the COVID-19 pandemic, the author was able to  
29 utilize the extra time to write and connect the unique moments in Deming's life to his  
30 Profound Knowledge.
```

```
31 ## Map_reduce  
32 This book is a portrait of W. Edwards Deming and John Willis, two soft-spoken experts  
33 in this century's biggest advancements. It captures the essence of how to improve most  
34 any business, with a focus on quality and understanding of the people responsible for  
35 that quality. It also gives a portrait of the hard-working, humble man from Wyoming  
36 who helped revolutionize manufacturing in Japan and around the world.
```

```
37 ## Chunks (chunk_size=1000, chunk_overlap=200)  
38
```

Untitled

Automatically reload preview

Preview HTML



## STUFF

The book is about W. Edwards Deming, a relatively quiet voice in one of this century's biggest advancements, and his System of Profound Knowledge. It connects the unique moments in Deming's life that culminated in his grand unifying theory of management that is the predictor of success or failure in every organization today. It was written by John Willis during the COVID-19 pandemic when he had extra time due to travel restrictions.

## Refine

The Phoenix Project is a modern retelling of the bestselling business book The Goal, written by Michael Lewis and published when W. Edwards Deming was ninety-three years old, just before his death. It focuses on software development and delivery, and includes DevOps in the subtitle. The DevOps Handbook, published in 2016 by Gene Kim, Patrick Debois, Jez Humble, and the speaker, provides a prescriptive solution to the topics discussed in The Phoenix Project. The two books have sold almost a million copies combined and contain important ideas about the right quality and uniformity being the foundation of commerce, prosperity, and peace. The book also provides a portrait of the hard-working, humble man from Wyoming who helped revolutionize manufacturing in Japan and around the world, and how his ideas were developed through his life experiences which culminated in the System of Profound Knowledge - a grand unifying theory of management. During the COVID-19 pandemic, the author was able to utilize the extra time to write and connect the unique moments in Deming's life to his Profound Knowledge.

## Map\_reduce

This book is a portrait of W. Edwards Deming and John Willis, two soft-spoken experts in this century's biggest advancements. It captures the essence of how to improve most any business, with a focus on quality and understanding of the people responsible for that quality. It also gives a portrait of the hard-working, humble man from Wyoming who helped revolutionize manufacturing in Japan and around the world.

# Generative AI Data Strategies (Observability and Tracing)

# MongoDB Metrics

Read Later Calendly - John W... Deming book (sha... ShareFile - Googl... Bookmarks John Willis - DEMI... YouTube Maps Home - Netflix Ikigai: The Japane...

Atlas Botchagal... Access Manager Billing All Clusters Get Help John

Project 0 Data Services App Services Charts

BOTCHAGALUPE TECHNOLOGIES > PROJECT 0 > CLUSTER 0 ac-sv4u64b-shard-00-00.xqonhqh.mongodb.net:27017 VERSION 6.0.9

DEPLOYMENT tabase ita Lake

SERVICES vice Sync triggers ita API ita Federation arch team Processing

SECURITY kickstart ckup itabase Access itwork Access ivanced

Status

GRANULARITY 1 hour ZOOM 2 months CURRENT DISPLAY 6/28/2023 12:00pm to 8/30/2023 12:00pm AT 1 HOUR GRANULARITY EXPORT

ADD CHART  DISPLAY OPCOUNTERS ON SEPARATE CHARTS  DISPLAY TIMELINE ANNOTATIONS

Opcounters ac-sv4u64b-shard-00-00.xqonhqh.mongodb.net:27017 0.4/S 0.2/S 0/S JUL 2023 AUG 2023

Connections 0 5 10 JUL 2023 AUG 2023

Network 0B/S 500B/S 1000B/S JUL 2023 AUG 2023 0.48/S 0.24/S 0/S

TOGGLE CHARTS - Opcounters - Connections - Network + Logical Size

The dashboard displays three metrics over a two-month period from July 28, 2023, to August 30, 2023. The Opcounters chart shows a sharp increase in operations per second (0.4/S) starting around August 1st. The Connections chart shows a peak of approximately 10 connections in early August. The Network chart shows a significant burst of data transfer (up to 1000B/S) occurring in late August.

# Langsmith Tracing

LangSmith

Projects 3

Datasets & Testing 1

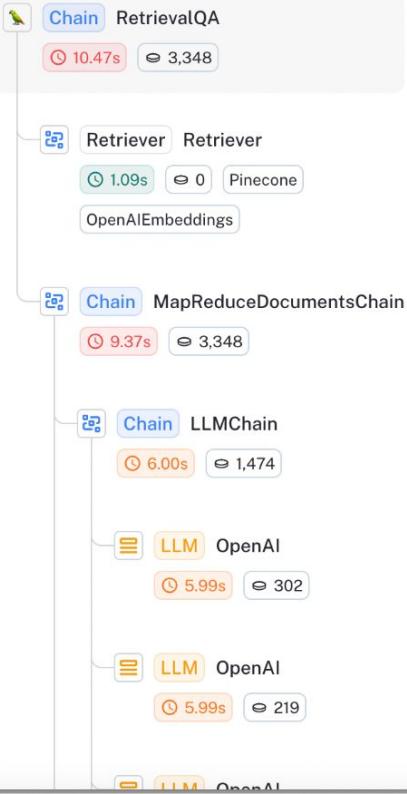
API Keys

Documentation

Personal  
botchagalupe@gmail.com

Personal > Projects > GAI-Demo-Agents\_Combined\_... > RetrievalQA

Trace



## RetrievalQA

Run Feedback Metadata

INPUT

Copy

1

query: Summarize the book

YAML

OUTPUT

Copy

1

result: " This book is a portrait of W. Edwards Deming and John Willis, two soft-spoken experts in this century's biggest advancements. It captures the essence of how to improve most any business, with a focus on quality and understanding of the people responsible for that quality. It also gives a portrait of the hard-working, humble man from Wyoming who helped revolutionize manufacturing in Japan and around the world."

2

source\_documents:

- metadata:

source:

/content/gdrive/MyDrive/GAI/transcripts/DEM\_Interior.6.24.23.pdf

Share

Rate Run

Add to Dataset

Open in I

START TIME

08/29/2023, 10:41:47 AM

END TIME

08/29/2023, 10:41:58 AM

STATUS

Success

TOTAL TOKENS

3,348 tokens

LATENCY

10.47s

TYPE

Chain

# WhyLabs Monitoring

WhyLabs Platform +

hub.whylabsapp.com/assets/model-18/profiles?dateRange=2023-02-15-to-2023-02-16&includeType=discrete&includeType=non-discrete&profile=1676419200000&profile=1676505600000&sortModelBy=LatestAlert...

WHYLABS Project Dashboard / embeddings\_mnist 02/15/2023 to 02/16/2023

Select project: embeddings\_mnist Summary Monitor Manager Profiles Inputs Outputs Segments Performance Tracing Explainability

Search features ? 2 filters applied Total features ? 12 Discrete Non-discrete

Profiles ? Select up to 3 dataset profiles for comparison

Profile 1: 2023-02-15 00:00:00 UTC

Profile 2: 2023-02-16 00:00:00 UTC

Feature name ? Est. mean ? Est. std dev ? Min ? Est. median ? Max ?

Feature name	Est. mean	Est. std dev	Min	Est. median	Max
pixel_values.6_distance	2027.13	259.73	1219.30	2010.38	3370.77
P1					
P2					
pixel_values.7_distance	2012.96	349.89	1049.50	1997.56	3441.86
P1					
P2					
pixel_values.8_distance	1937.80	263.32	1253.61	1893.26	3139.48
P1					
P2					
pixel_values.9_distance	1943.58	263.72	1258.78	1903.46	3124.56
P1					
P2					
pixel_values.closest	0.00	0.00	0.00	-	0.00
P1					
P2					

Rows per page: 30 1-12 of 12 | < < > >|

pixel\_values.closest Distribution analysis Click legends to toggle distributions

Frequent items data Click to toggle: Profile 1 Profile 2

Item count

Frequent items Show Histogram data

Item	Profile 1	Profile 2
0	5905	642
1.00	9218	721
2.00	5516	614
3.00	6670	734
4.00	6205	699
5.00	5191	609
6.00	6061	652
7.00	6172	673
8.00	5581	616

# Demo



# Resources

- <https://platform.openai.com/docs/guides/embeddings/limitations-risks>
- <https://python.langchain.com/docs/modules/chains/>
- <https://smith.langchain.com/>
- <https://www.mongodb.com/products/platform/atlas-vector-search>
- [https://python.langchain.com/docs/integrations/vectorstores/mongodb\\_atlas](https://python.langchain.com/docs/integrations/vectorstores/mongodb_atlas)
- <https://techstrong.ai/author/john-willis/>

