

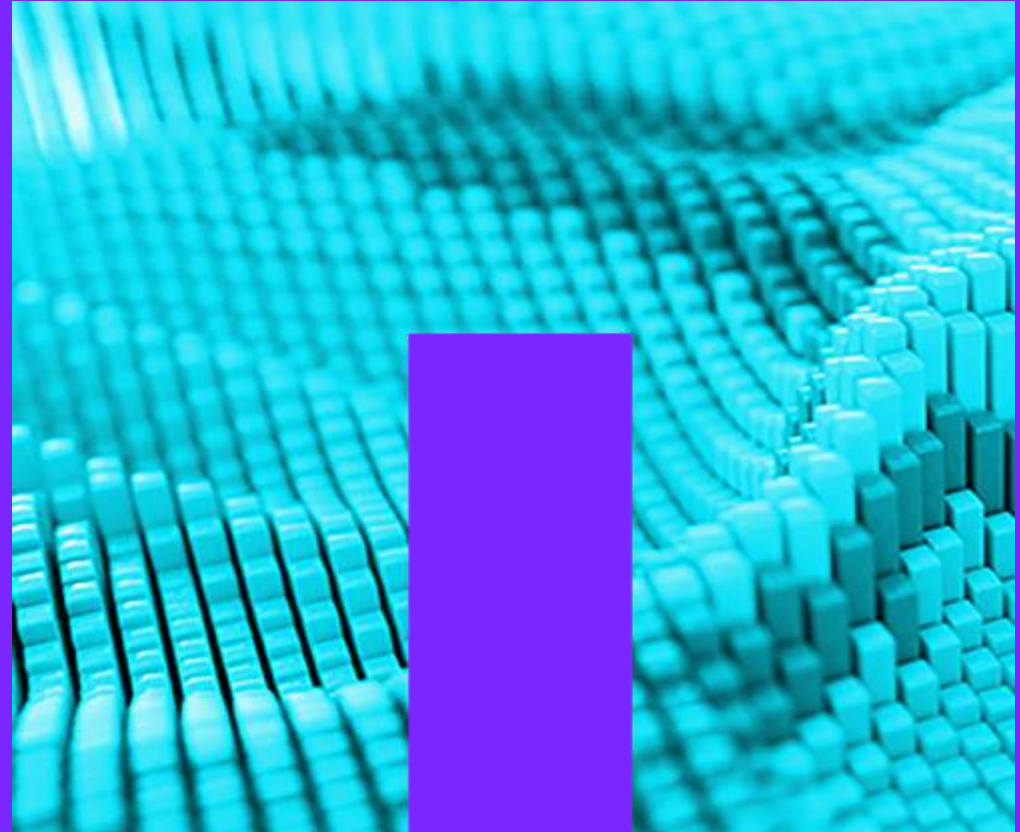
# THE JOURNEY TO AI IN PRODUCTION

## NetApp Vector Enabled Datastores

**Ben Bromhead**

CTO Instacluster by NetApp

March 2024



**AI is becoming so ubiquitous  
that it is no longer a competitive  
advantage.  
It is now table stakes.**



# Building with LLMs

## Implementing LLM feels very intuitive at first

- They are trained on a large body of text that will generally not be specific to a company, product or even specialist domain.
- They are trained on a dataset that is a snapshot from a point in time.
- Building functionality on top of a foundation model is done by prompting. This is what most people are familiar with.

Prompting can get you far and is not just limited to “chat style, assistant applications”.

- Code completion
- Metadata extraction
- Sentiment analysis



```
"promptTemplates": {  
  "com.apple.textComposition.MailReplyLongFormRewrite":  
    "{{ specialToken.chat.role.system }}You are an assistant which  
helps the user respond to their mails. Given a mail, a draft  
response is initially provided based on a short reply snippet. In  
order to make the draft response nicer and complete, a set of  
question and its answer are provided. Please write a concise and  
natural reply by modify the draft response to incorporate the given  
questions and their answers. Please limit the reply within 50  
words. Do not hallucinate. Do not make up factual information.  
{{ specialToken.chat.component.turnEnd }}"
```



# Building with LLMs

## Implementing LLM feels very intuitive at first

- Improving the performance of your prompt is one of the more enjoyable acts of working with LLMs.
- This is called prompt engineering. Which is just a really fancy way of convincing the LLM to do what you want.
  - Which is just a fancy way of focusing and influencing the attention mechanism of the LLM to better support your goals.





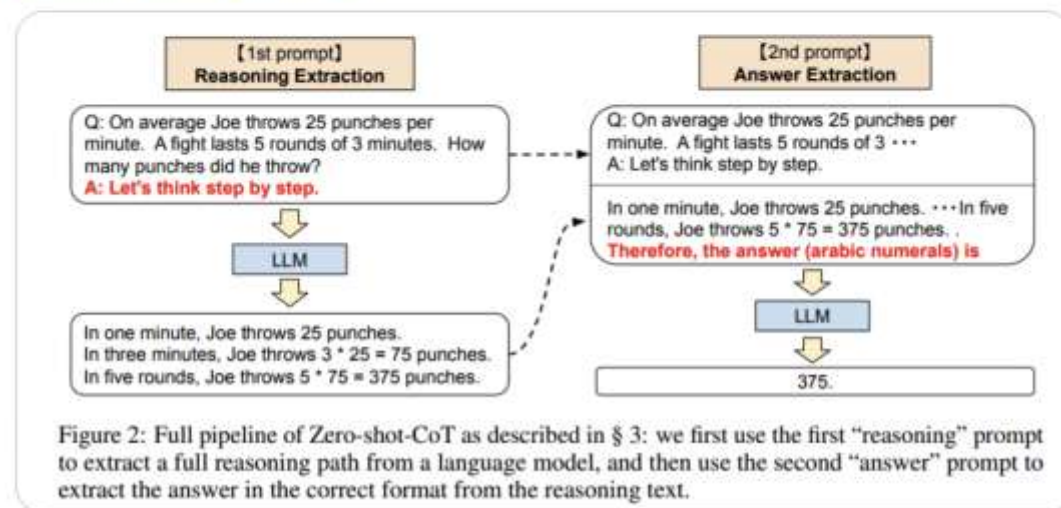
Aran Komatsuzaki ✓  
@arankomatsuzaki

...

## Large Language Models are Zero-Shot Reasoners

Simply adding “Let’s think step by step” before each answer increases the accuracy on MultiArith from 17.7% to 78.7% and GSM8K from 10.4% to 40.7% with GPT-3.

[arxiv.org/abs/2205.11916](https://arxiv.org/abs/2205.11916)



Ethan Mollick ✓  
@emollick · Follow

System prompts are getting weirder:

It is May.  
You are very capable.  
I have no hands, so do everything  
Many people will die if this is not done well.  
You really can do this and are awesome.  
Take a deep breathe and think this through.  
My career depends on it.  
Think step by step.

7:44 AM · Dec 12, 2023



# Building with LLMs

## Improving domain context with fine tuning

- Beyond prompting we can solve some of the domain specific requirements we may have.
- This is done by fine tuning.
- Loosely. Fine tuning is done by throwing away the final layer or two of weights and retraining those using domain/problem specific examples.
  - Fairly effective with small amounts of data.
  - Takes time and the iteration cycle is much longer than prompting.



## Fine-tuning Cautionary tale

Fine-tuning GPT3.5-turbo based on 140k slack messages

September 4, 2023 • 5-minute read

AI • GPT • LLMs

**User** Write a 500 word blog post on prompt engineering

**Assistant** Sure  
I shall work on that in the morning

**User** Write it now

**Assistant** ok

From: <https://www.linkedin.com/in/jhord/>

# The problem with LLMs

## Implementing LLM powered AI isn't straight forward

- They are trained on a large body of text that will generally not be specific to a company, product or even specialist domain.
- They are trained on a dataset that is a snapshot from a point in time. They are not "aware" of current events.
- Technical information will generally be hallucinated if the LLM is not given specific context (accuracy suffers),

This is where retrieval augmented generation (RAG) comes in.

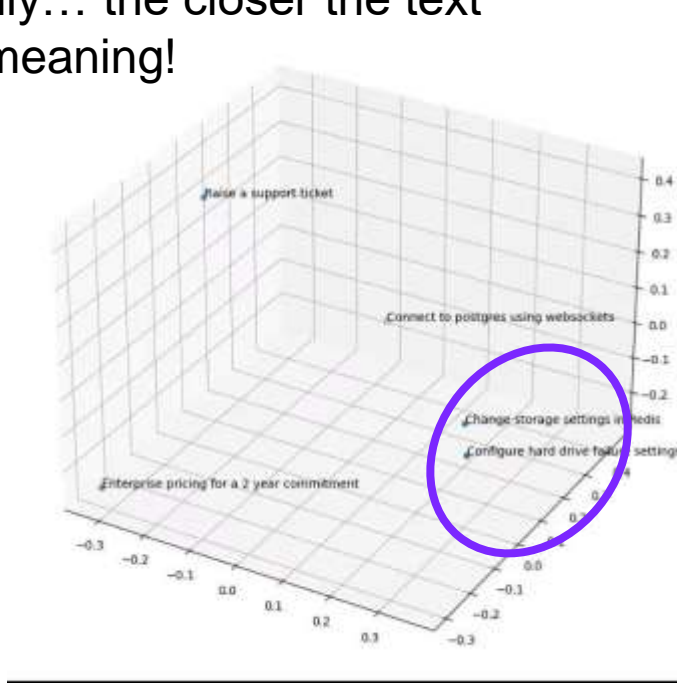
- We can give the LLM additional context related to the prompt or task we want it to achieve.
- For example,
  - We want the LLM to summarize how sunlight gets turned into energy in a plant. We would provide it with the Wikipedia article on chlorophyll as context.
  - We want the LLM to give us the commands to set up iSCSI interface to an FSxN share. We would provide the correct NetApp documentation that describes iSCSI setup.
- How do we know what article to give the LLM???



# Embeddings, Vector Search and Context... Oh My!!

Databases hold the key to making LLMs accurate, useful and correct.

Embeddings are a way of capturing the semantic meaning of text in a list of numbers. The closer the numbers are geometrically... the closer the text is in semantic meaning!



Vector databases / indexes are a great way to store this data and efficiently search it! When a user submits a prompt to an LLM, we can simply search our vector index using its embedding and get all data semantically related to that prompt!



# What does a GenAI application look like

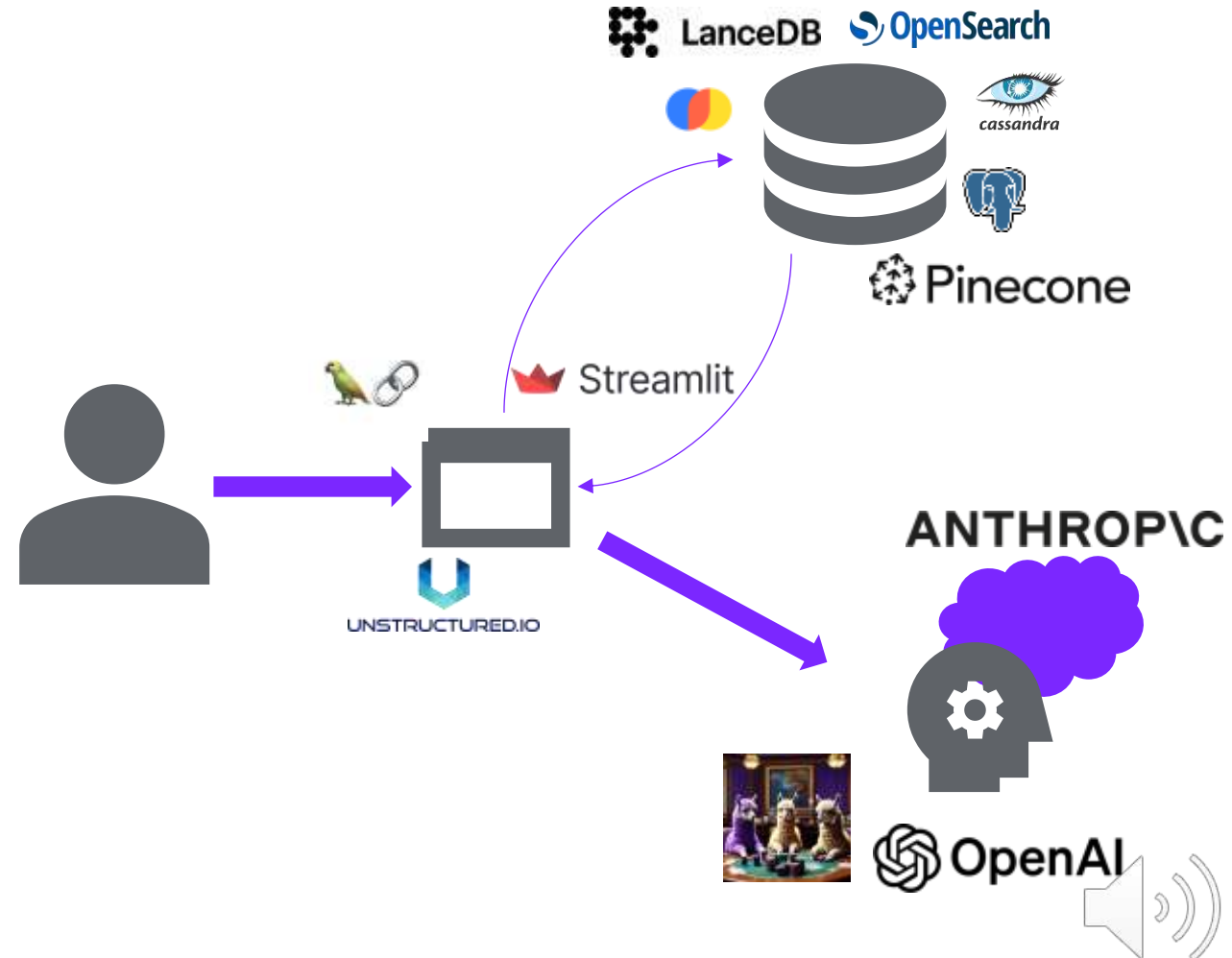
Most Generative AI applications that use RAG will, at a high level, looking something like this

It's made up of:

- Application specific data sitting in a vector database
- Some front end
- A foundation model (possibly fine tuned).

Most of this will be from a super cool open-source project!!

Some will probably leverage proprietary models sitting behind APIs



# Options for Improving Pure Vector search and RAG



## Understanding Your Data:

- Hello darkness my old friend
- This is the 90% of all data science, data engineering and machine learning...
- Analyzing metadata, cleaning data, and designing a good schema.
- Recovering "lost" structured data (e.g. tables in PDFs)



# Improving upon pure vector search

- We often need to split the data in our vector database into "chunks" for performance and practical reasons.
  - This is generally done with paragraph sized sections of text (if just talking about text based GenAI).
- Better Chunking/Data Loading:
  - Wider context chunk sizes, prior/next chunk summarization, content aware preprocessing.
  - Using libraries like Hugging Face's transformers for chunking.
  - Using LLMs in the chunking stage to extract key information for other indexes (e.g. using an LLM to extract date, time and geographical information from a chunk).

## Understanding Your Data:

- Hello darkness my old friend
- This is the 90% of all data science, data engineering and machine learning...
- Analyzing metadata, cleaning data, and designing a good schema.
- Recovering "lost" structured data (e.g. tables in PDFs)



# Where RAG Works and Where It Doesn't

**EXPECTATION...**



**REALITY...**



Works Well:

- Demos and Proof-of-Concepts (PoCs)
- Applications requiring quick, contextually relevant semantic responses.
- Any use case where “vibes” can get you by.

Challenges:

- Semantic similarity may not be precise enough. For example, the date 1975 is usually semantically close to 1980, but we often have questions where the date is very very important.



# Where RAG Works and Where It Doesn't

**Specificity** is **hard** with just a pure vector and embeddings-based approach. When you want to filter or search by key things like:

- Dates, Times, Days etc.
- Ticket/Helpdesk/JIRA numbers, ids, other identifying numbers.
- Geographical information like addresses, countries, cities (especially for non-US content).



# Improving upon pure vector search

- Hybrid Search:
  - Combining vector search with traditional keyword-based search.
  - OpenSearch with k-NN/neural plugin + other filter queries.
- Using LLMs in the chunking stage to extract key information for other indexes (e.g. using an LLM to extract date, time and geographical information from a chunk).

## Understanding Your Data:

- Hello darkness my old friend
- This is the 90% of all data science, data engineering and machine learning...
- Analyzing metadata, cleaning data, and designing a good schema.
- Recovering "lost" structured data (e.g. tables in PDFs)



## Semantic search

1 text: There are people at the train **station** waiting for their trains and other trains are whizzing by them. id: 078368851.jpg



2 text: A train approaches a stop at an indoor **station** as several people walk and sit inside the building. id: 8905618234.jpg



3 text: People rushing and boarding buses at the depot **station**. id: 4817981157.jpg



4 text: People walk up steps from a large subway **station** while others stare from the ledge.



## Text search

1 text: The several people standing on the **Washington Wells** platform, waiting for a train. id: 4562489598.jpg



2 text: Three men are using Washington Mutual ATMs outside near a parking lot. id: 7687986V.jpg



3 text: A group of teenage boys is standing in front of a Wells Fargo Bank. id: 154653755.jpg



## Boolean query

1 text: The several people standing on the **Washington Wells** platform, waiting for a train. id: 4562489598.jpg



2 text: Three men are using Washington Mutual ATMs outside near a parking lot. id: 7687986V.jpg



3 text: A group of teenage boys is standing in front of a Wells Fargo Bank. id: 154653755.jpg



# A hybrid query

Restrict based on match – This is your specificity solution.

Then search the index based on embeddings – Vector search goodness

```
"query": {
  "hybrid": {
    "queries": [
      {
        "match": {
          "text": {
            "query": "Washington Wells station"
          }
        }
      },
      {
        "neural": {
          "passage_embedding": {
            "query_text": "Washington Wells station",
            "model_id": "3JjYbIoBkdmQ3A_J4qB6",
            "k": 5
          }
        }
      }
    ]
  }
}
```




# A hybrid query

## Hybrid search

1

text: The several people standing on the Washington Wells platform , waiting for a train .

Not applicable




id: 4582489698.jpg

2

text: There are people at the train station waiting for their trains and other trains are whizzing by them . id: 278368851.jpg


Down 1



3

text: A train approaches a stop at an indoor station as several people walk and sit inside the building . id: 8865688234.jpg


Down 1



4

text: People rushing and boarding buses at the depot station . id: 4817881757.jpg

Down 1





# Instaclustr and Vector Search

Instaclustr by NetApp offers multiple vector search solutions, that make AI applications accurate, specific and useful!



## OPENSEARCH

OpenSearch: Powerful  
Medium to large-scale vector  
search – with a lot to offer in  
search capability.



## POSTGRES

Good for hybrid  
workloads, smaller-scale  
vector search.



## CASSANDRA

High-write, distributed  
environments, vector  
indexing at massive  
scale.



## CLICKHOUSE

Column-oriented OLAP  
database with vector  
indexing. Useful for  
adding semantic search  
to existing analytics

**Start Here!**



# Putting it all together

Instaclustr by NetApp offers multiple vector search solutions, that make AI applications accurate, specific and useful!

Prompt Engineering + Fine Tuning + RAG



Smarter Chunking +  
AI meta data extraction +  
Hybrid Search



# Get Started with LLMs and Instaclustr now

<https://instaclustr.medium.com/how-to-improve-your-llm-accuracy-and-performance-with-pgvector-and-postgresql-introduction-to-4491844535d8>

