Citations and Attributions

Yujian Tang | Zilliz

Speaker

Yujian Tang

Developer Advocate, Zilliz

yujian@zilliz.com
https://www.linkedin.com/in/yujiantang
https://www.twitter.com/yujian_tang

zilliz

CONTENTS

01 **Why Do Citations Matter?**

02 **How Can You Build a Citation Engine?**

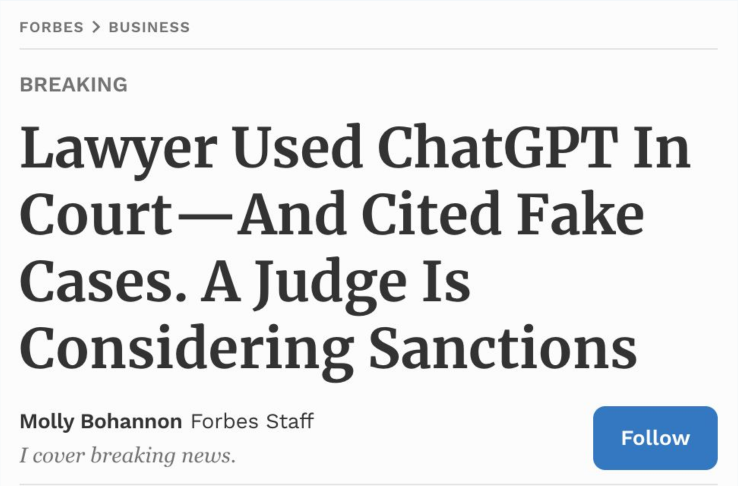
03 **What Goes Into a Citation Engine?**

04 **FAQ**

01

Why Do Citations Matter?

A Hallucination Problem



FORBES > BUSINESS

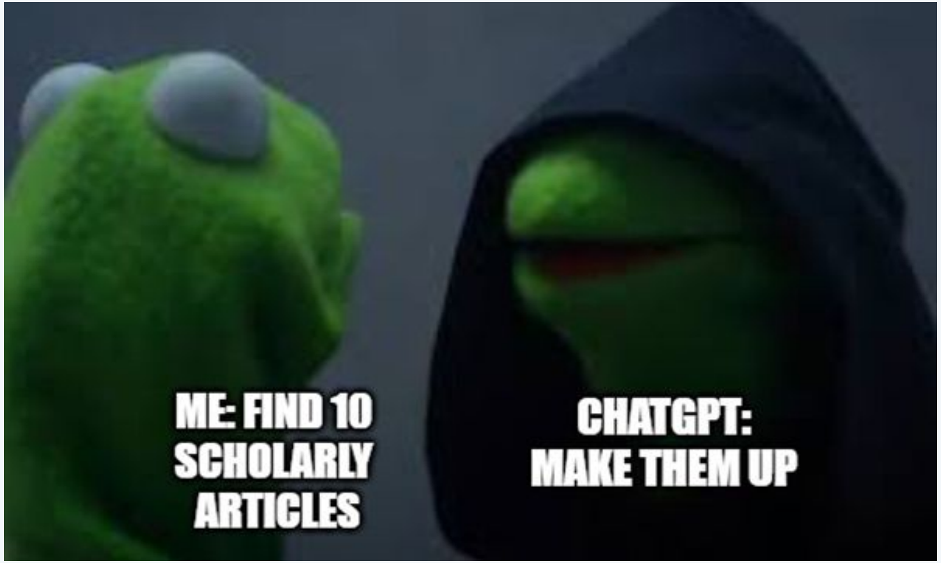
BREAKING

Lawyer Used ChatGPT In Court—And Cited Fake Cases. A Judge Is Considering Sanctions

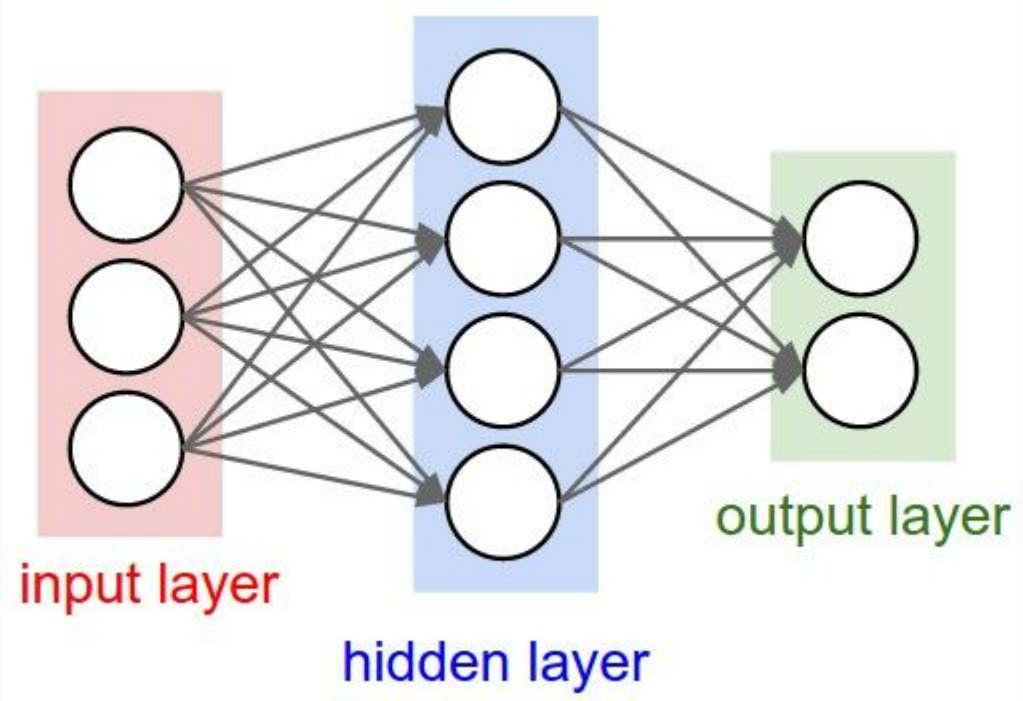
Molly Bohannon Forbes Staff

I cover breaking news.

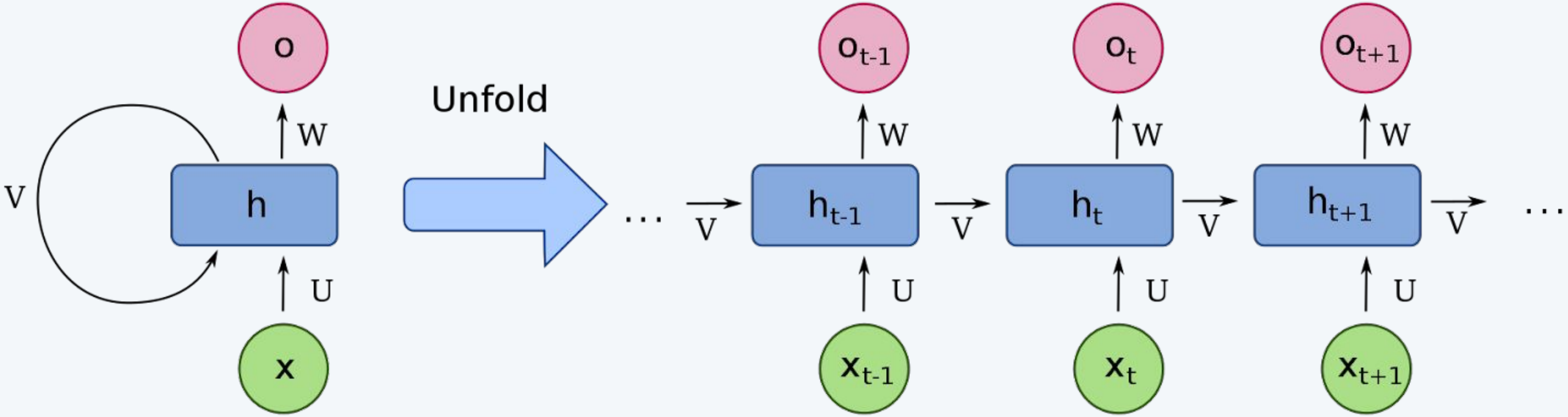
Follow



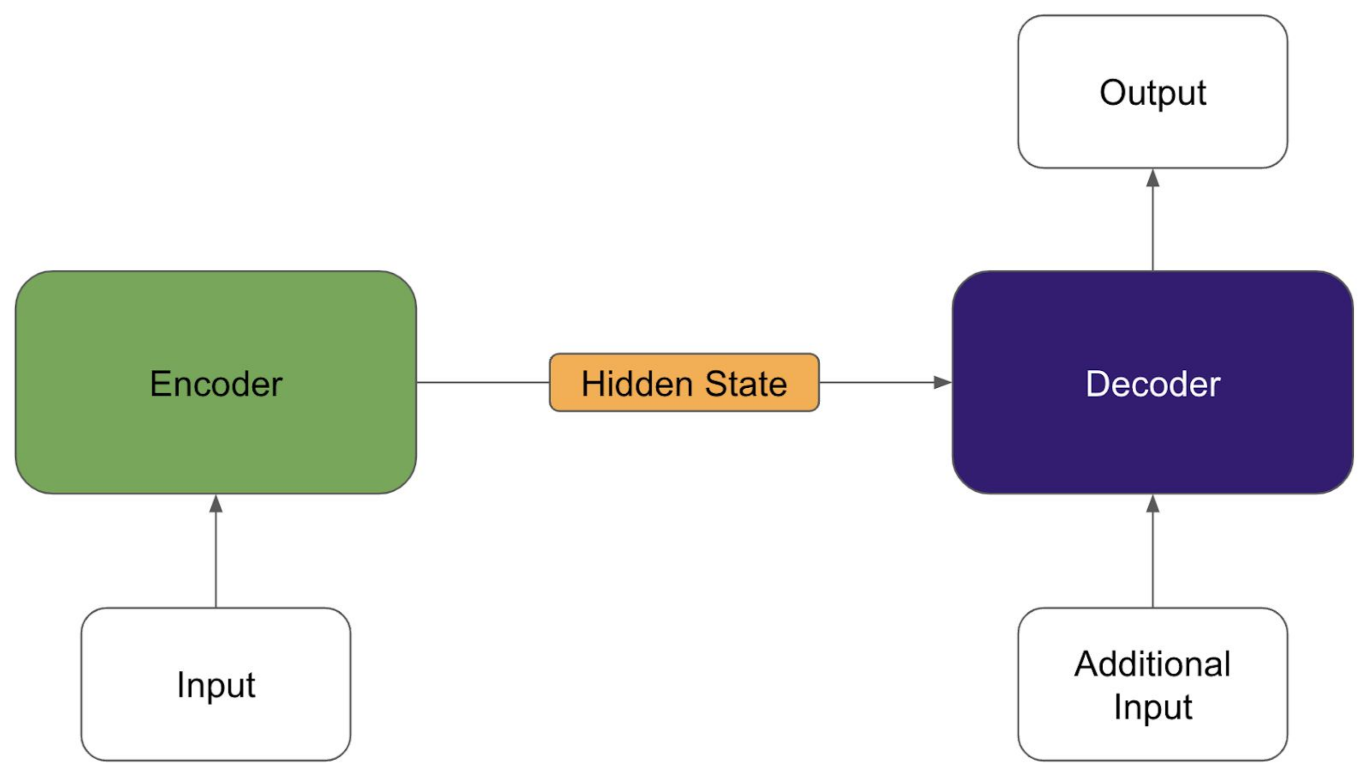
A Basic Neural Net



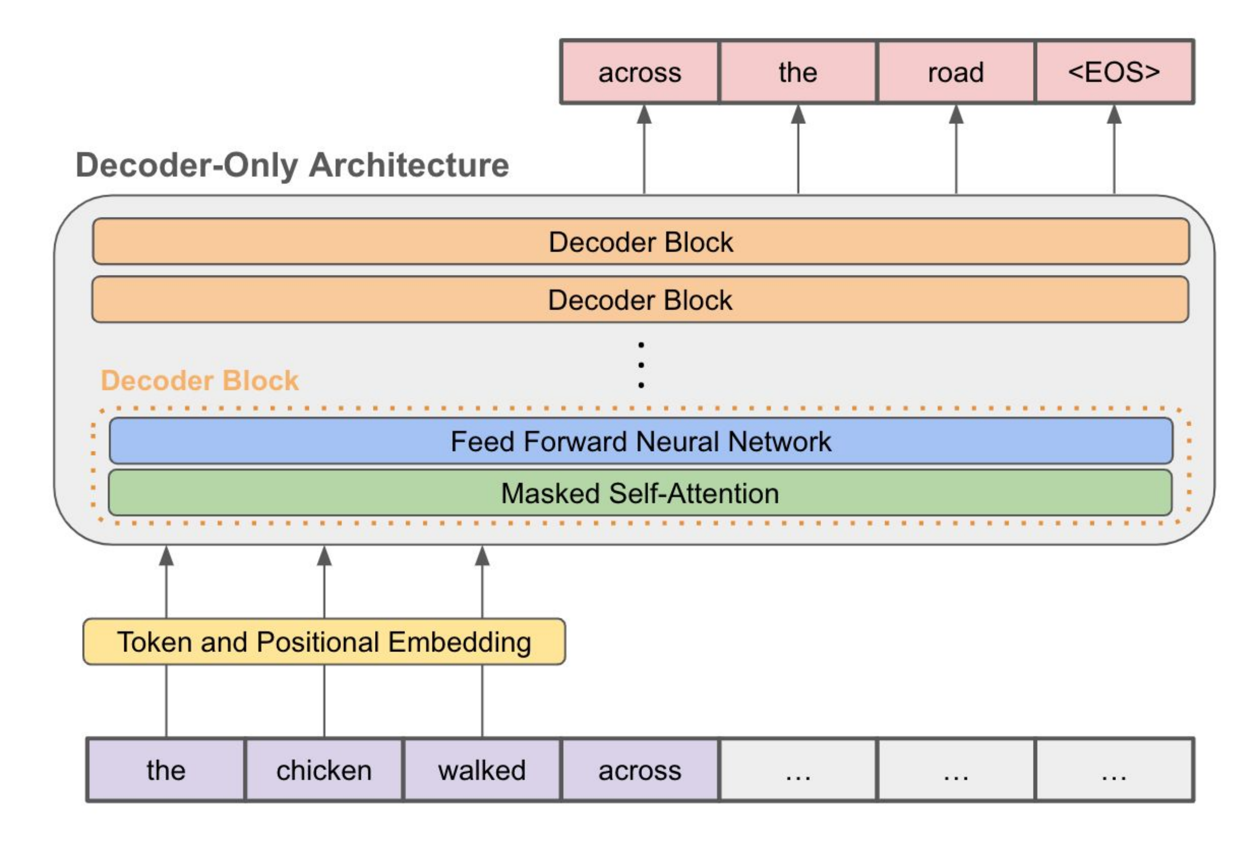
A Recurrent Neural Network



A Transformer Architecture



GPT Architecture



Takeaway

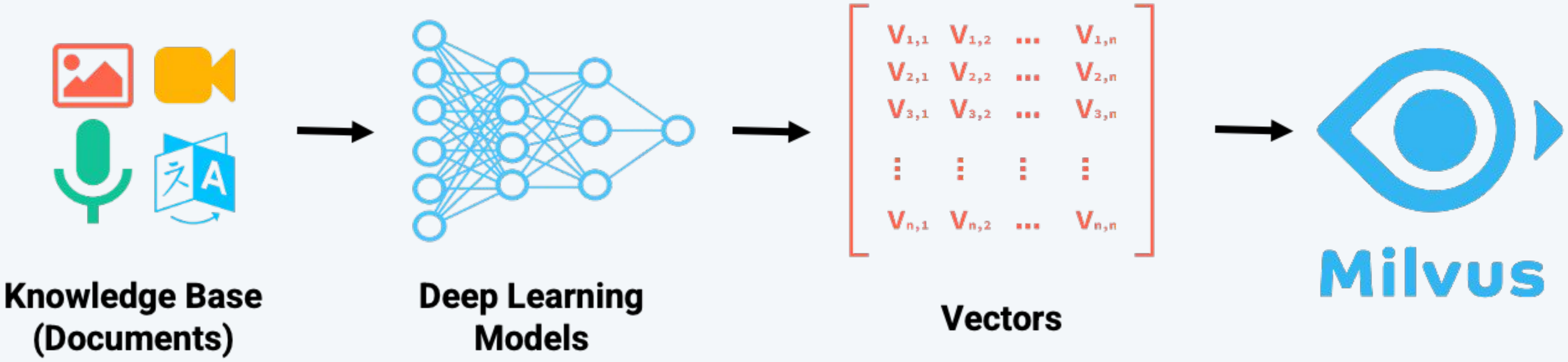
The reason ChatGPT hallucinates is because …

It’s set up to predict a series of words (tokens)

02

How Can You Build a Citation Engine?

Process for Basic Data Injection to LLMs



Knowledge Base (Documents) → Deep Learning Models → Vectors → Milvus

$$\begin{bmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,n} \\ V_{3,1} & V_{3,2} & \cdots & V_{3,n} \\ \vdots & \vdots & \vdots & \vdots \\ V_{n,1} & V_{n,2} & \cdots & V_{n,n} \end{bmatrix}$$

Semantic Similarity

Queen - Woman + Man = King

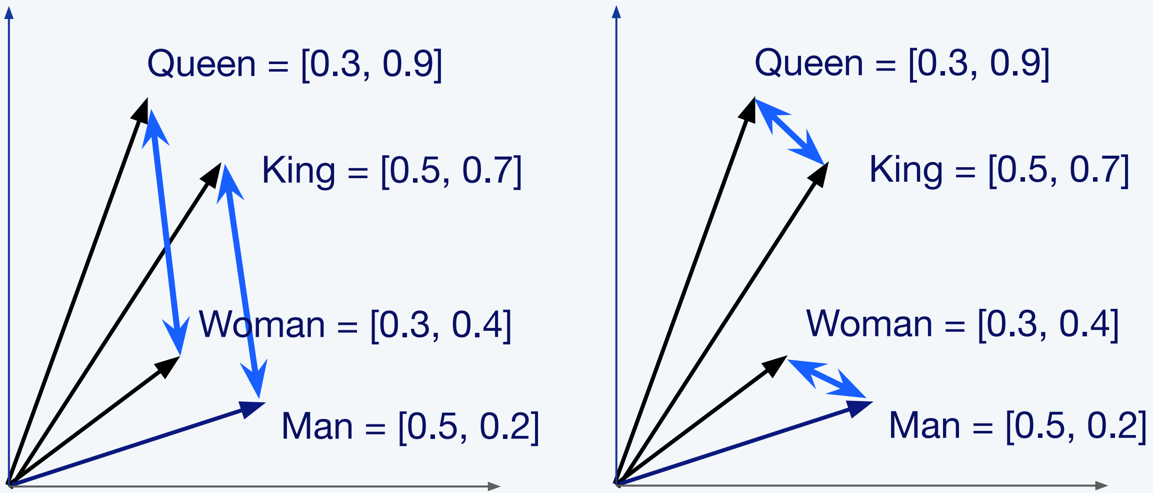


Image from Sutor et al

Semantic Similarity

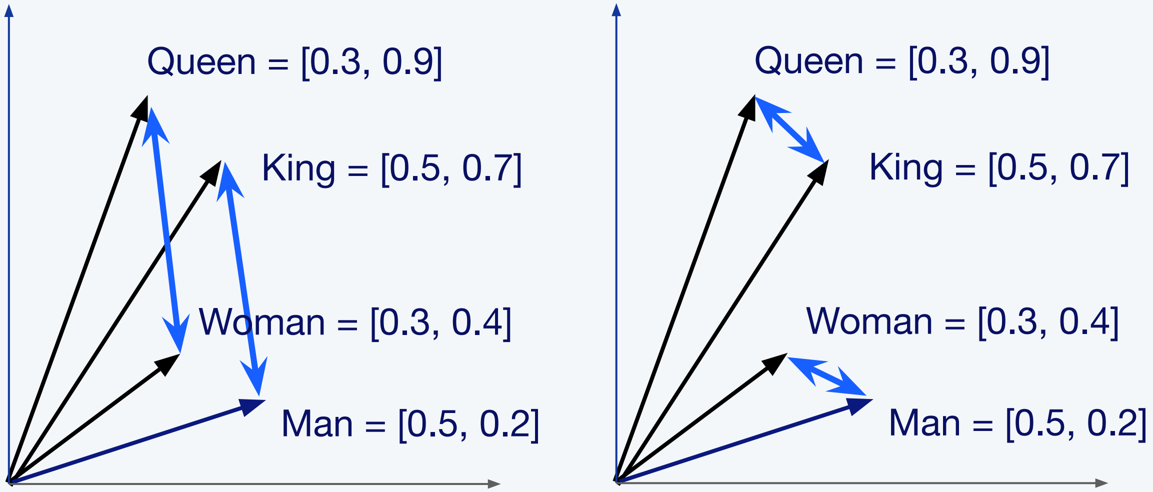


Image from Sutor et al

Queen - Woman + Man = King

Queen = [0.3, 0.9]
- Woman = [0.3, 0.4]

[0.0, 0.5]

Semantic Similarity

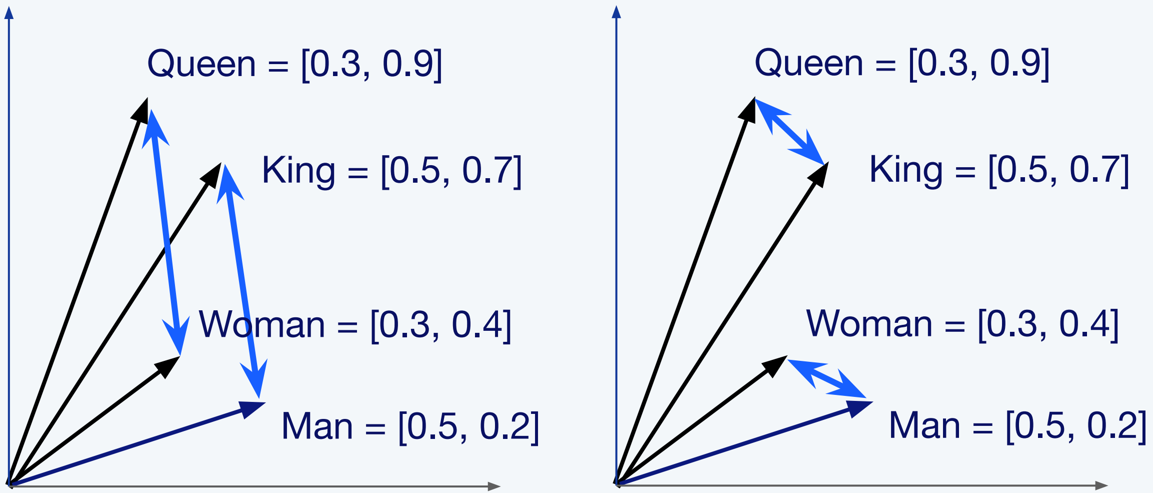


Image from Sutor et al

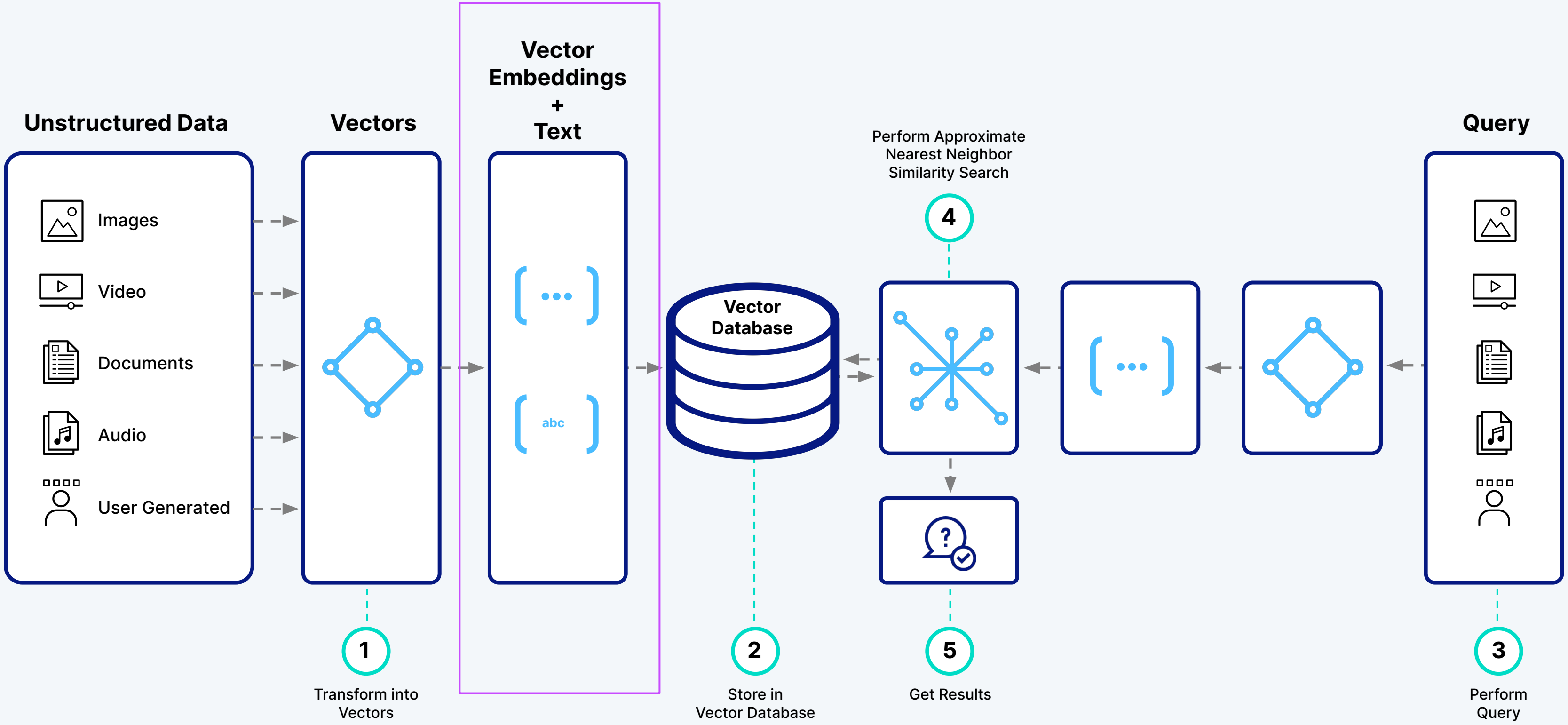
Queen - Woman + Man = King

Queen = [0.3, 0.9]
- Woman = [0.3, 0.4]

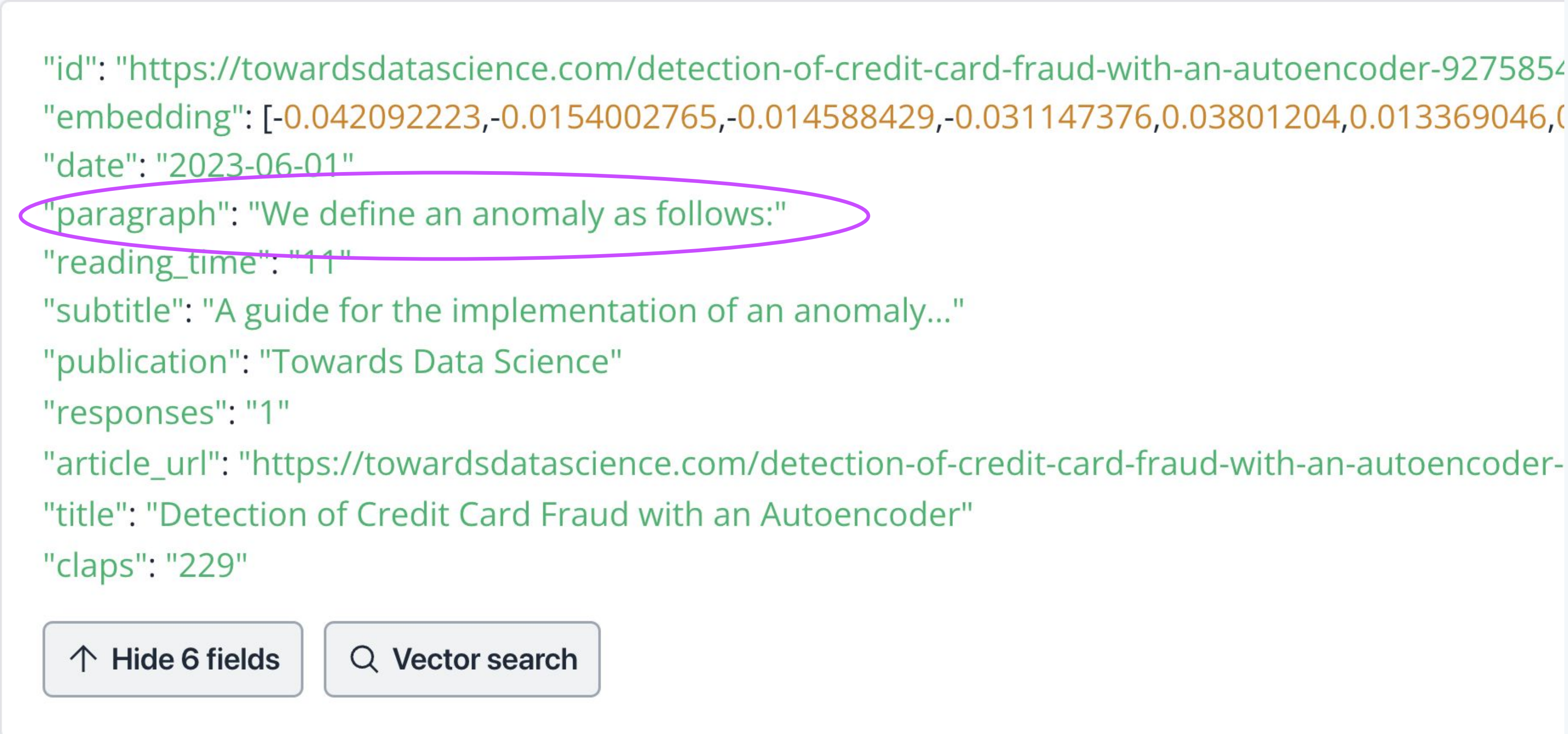
[0.0, 0.5]
+ Man = [0.5, 0.2]

King = [0.5, 0.7]

Typical Similarity Search



What Does Your Data Look Like?



```
"id": "https://towardsdatascience.com/detection-of-credit-card-fraud-with-an-autoencoder-927585
"embedding": [-0.042092223,-0.0154002765,-0.014588429,-0.031147376,0.03801204,0.013369046,
"date": "2023-06-01"
"paragraph": "We define an anomaly as follows:"
"reading_time": "11"
"subtitle": "A guide for the implementation of an anomaly..."
"publication": "Towards Data Science"
"responses": "1"
"article_url": "https://towardsdatascience.com/detection-of-credit-card-fraud-with-an-autoencoder-
"title": "Detection of Credit Card Fraud with an Autoencoder"
"claps": "229"
```

03

What Goes Into a Citation Engine?

zilliz

LLM App Framework

CVP Stack

C: ChatGPT (or any other LLM)

- This can also be interpreted as the “processor” block for CVP

V: Vector database (e.g. Milvus)

- Can also be interpreted as the “storage” block for CVP

P: Prompt-as-code (e.g. Haystack)

- Interface between processor and storage blocks

Where Do Citations Sit?

CVP Stack

C: ChatGPT (or any other LLM)

- This can also be interpreted as the “processor” block for CVP

V: Vector database (e.g. Milvus)

- **Can also be interpreted as the “storage” block for CVP**

P: Prompt-as-code (e.g. Haystack)

- **Interface between processor and storage blocks**

Example Notebook

04

FAQs

FAQ - Use Cases

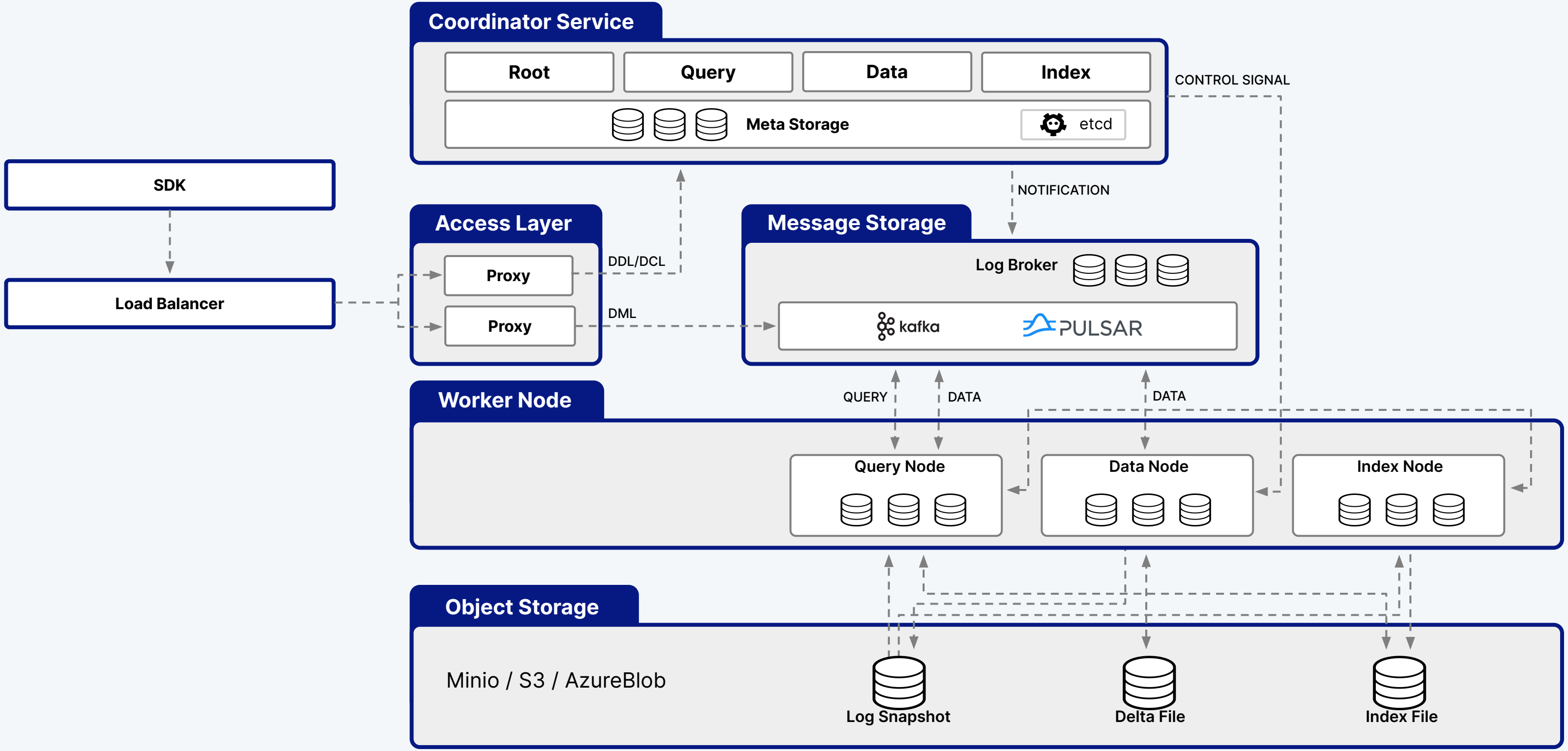
- When *NOT* to use
- CSV Files? PDFs?
- Hybrid Search

THANK YOU

zilliz

haystack
by deepset

Vector Database Architecture

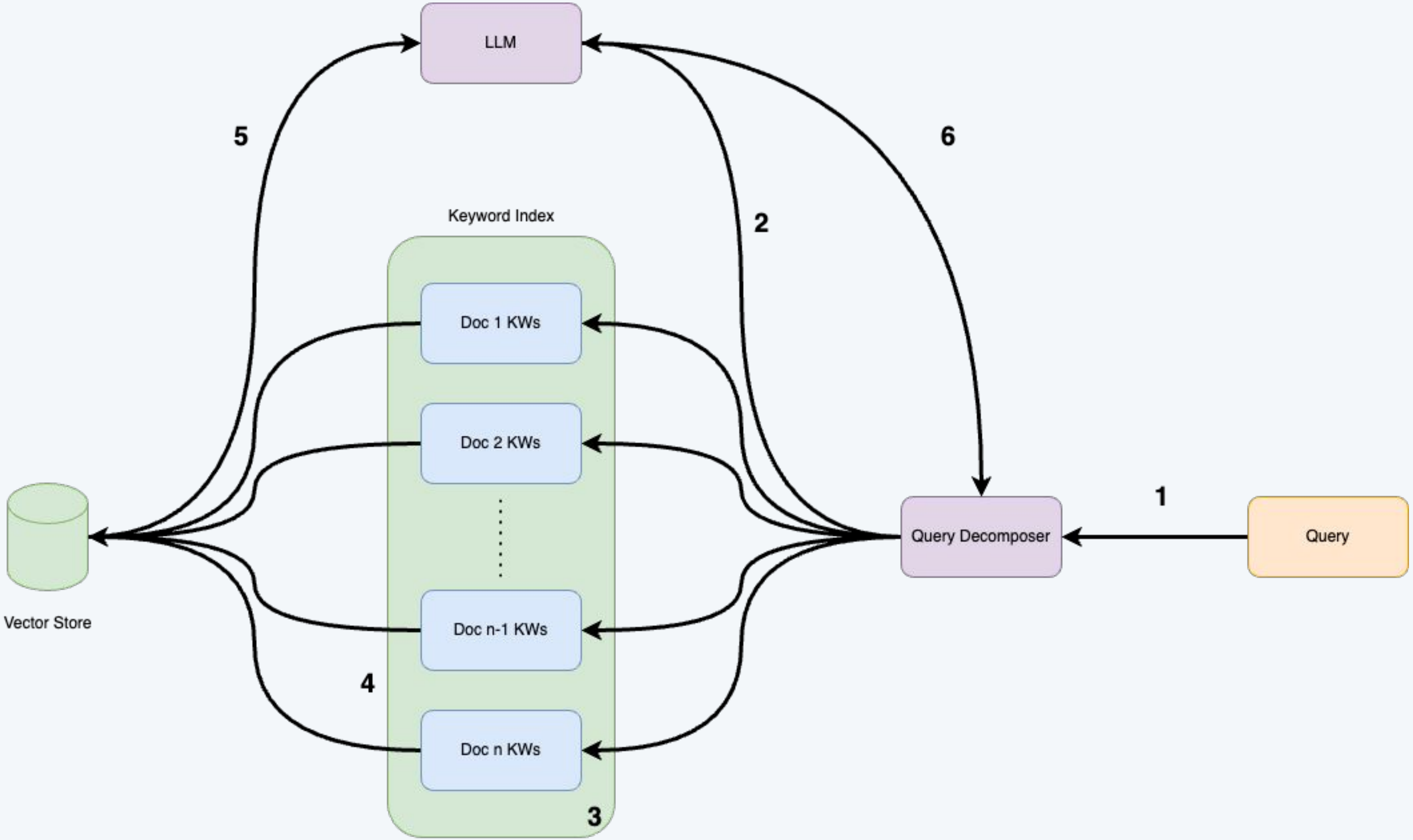


milvus

VECTOR DATABASE

zilliz

Architecture



Multi Document Query Engine Code Sample

05

Appendix

An Example Idea

Example

- A company has 100,000s+ pages of proprietary documentation to enable their staff to service customers.

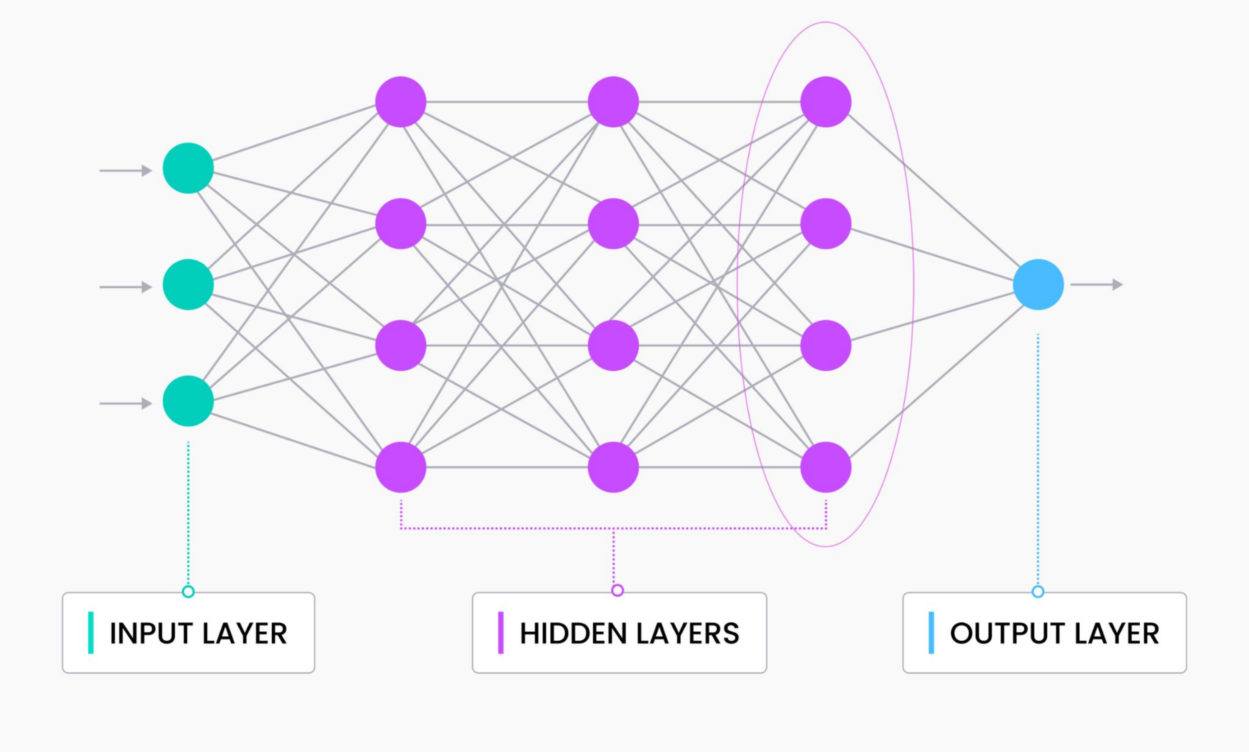
Problem

- Searching can be slow, inefficient, or lack context.

Solution

- Create internal chatbot with ChatGPT and a vector database enriched with company documentation to provide direction and support to employees and customers.

How are these generated?



Traditional databases face lots of challenges to manage vectors

- Inefficiency in High-dimensional spaces
- Suboptimal Indexing
- Inadequate query support
- Lack of scalability
- Limited analytics capabilities
- Data conversion issues

Why a Vector Database?

Purpose-built to store, index and query vector embeddings from unstructured data.

Vector database

- Advanced filtering (filtered vector search, chained filters)
- Hybrid search (e.g. full text + dense vector)
- Durability (any write in a db is durable, a library typically only supports snapshotting)
- Replication / High Availability
- Sharding
- Aggregations or faceted search
- Backups
- Lifecycle management (CRUD, Batch delete, dropping whole indexes, reindexing)
- Multi-tenancy

Vector search library

- High-performance vector search

How do I support different applications?

- High query load
- High insertion/deletion
- Full precision/recall
- Accelerator support (GPU, FPGA)
- Billion-scale storage