Applied Deep Learning

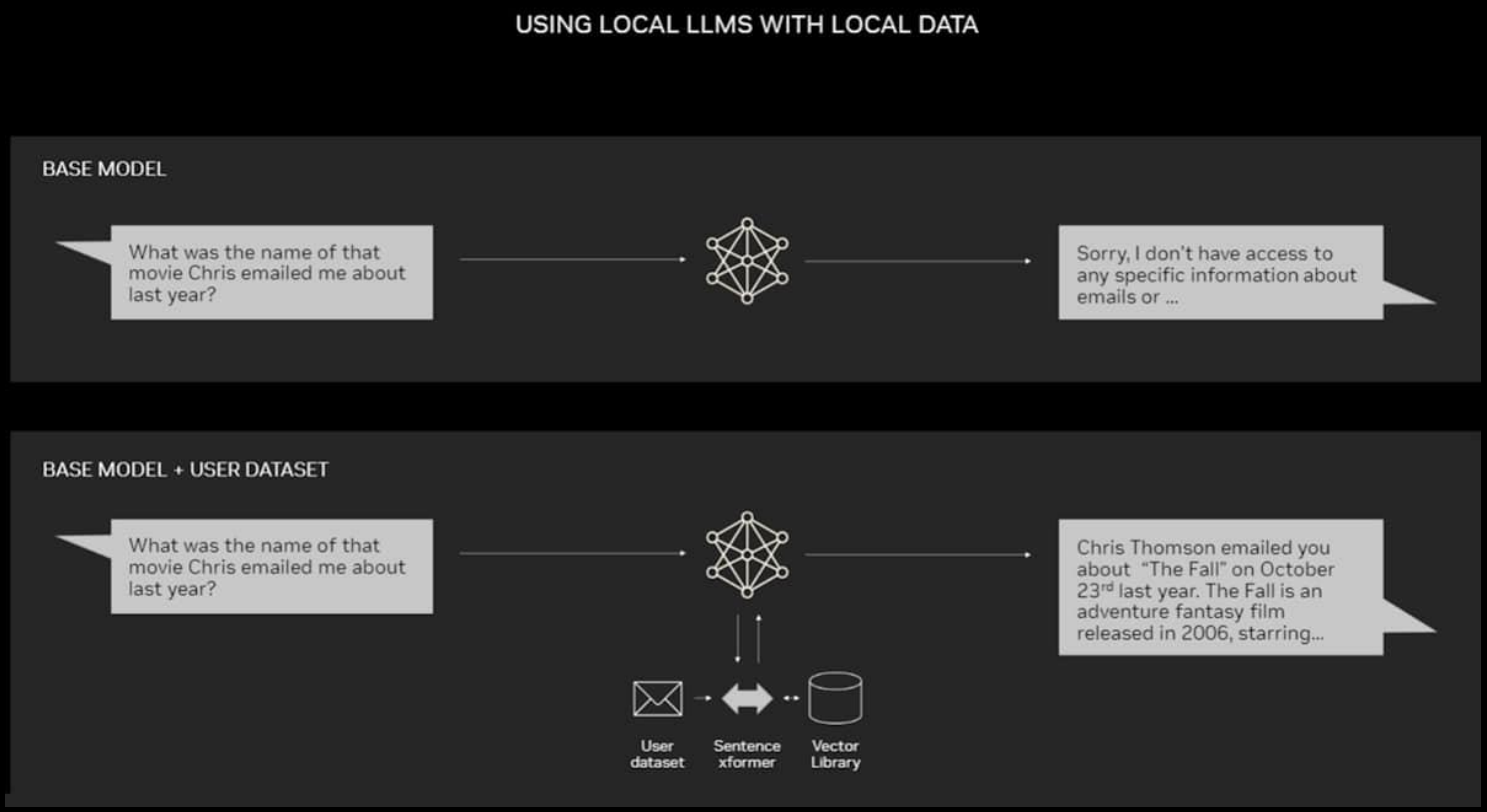
5350 Graph Retrieval Augmented Generation (RAG)

portilla@gmail.com

Jesus.portilla@Colorado.edu

https://github.com/iportilla/RAG

Retrieval Augmented Generation (RAG)



https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/

LLM Shortcomings

write a **get_embedding** function in python using the latest azure openai client

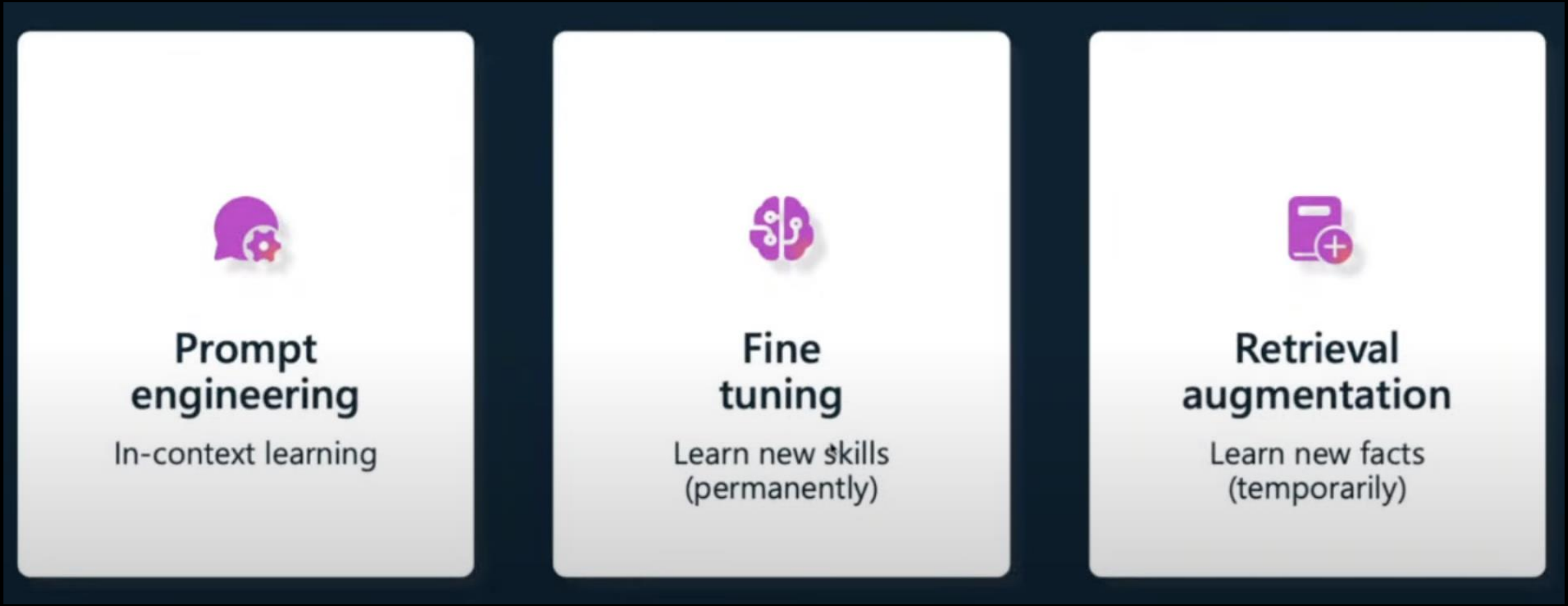
python

```python
def get_embedding(texts, model="text-embedding-ada-002"):
    """
    Retrieve embeddings for a list of texts using the specified model.

    Parameters:
    - texts (list of str): The texts to embed.
    - model (str): The model to use for embedding. Default is "text-embedding-ada-002".

    Returns:
    - embeddings (list of list of floats): The embeddings for each text.
    """
    response = openai.Embedding.create(
        input=texts,
        model=model
    )
    embeddings = [embedding['embedding'] for embedding in response['data']]
    return embeddings
```

Incorporate Domain Knowledge with LLM

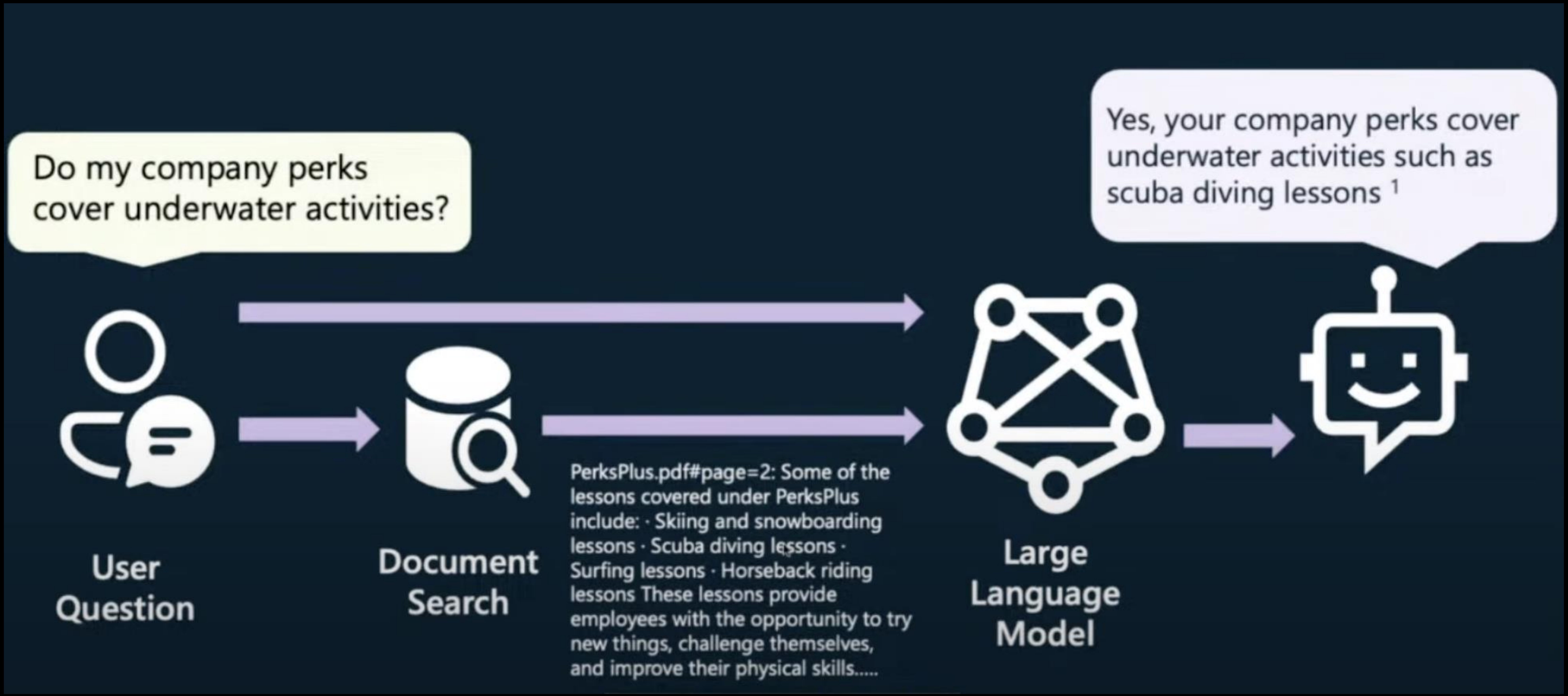


https://www.youtube.com/watch?v=vuOA13Y_Qzk

The Benefits of RAG

- Up-to-date public knowledge (AZ OpenAI documentation)
- Access to internal knowledge (Company HR docs)

RAG – Retrieval Augmented Generation



https://www.youtube.com/watch?v=vuOA13Y_Qzk

Robust retrieval for RAG

- Responses only as good as retrieved data
- Keyword search recall challenges
- Vector-based retrieval finds docs by Semantic similarity

Example

Question:

"Looking for lessons on underwater activities"

Won't match:

"Scuba classes"

"Snorkeling group sessions"

Vector embeddings

- An embedding encodes an input as a list of FP numbers
- “dog” -> [0.014, -0.05, ...]
- Different models output different embeddings (different lengths)

https://aka.ms/aitour/vectors

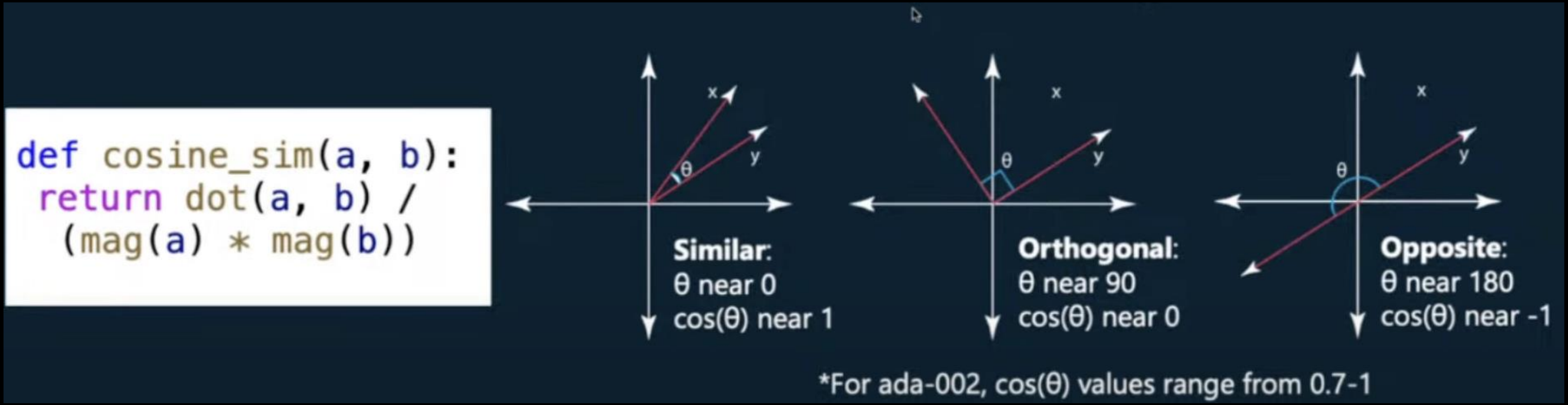
https://pamelafox.github.io/vectors-comparison/

https://pamelafox.github.io/vectors-comparison/movies.html

https://github.com/Azure-Samples/rag-with-azure-ai-search-notebooks/blob/main/vector_embeddings.ipynb

Vector similarity

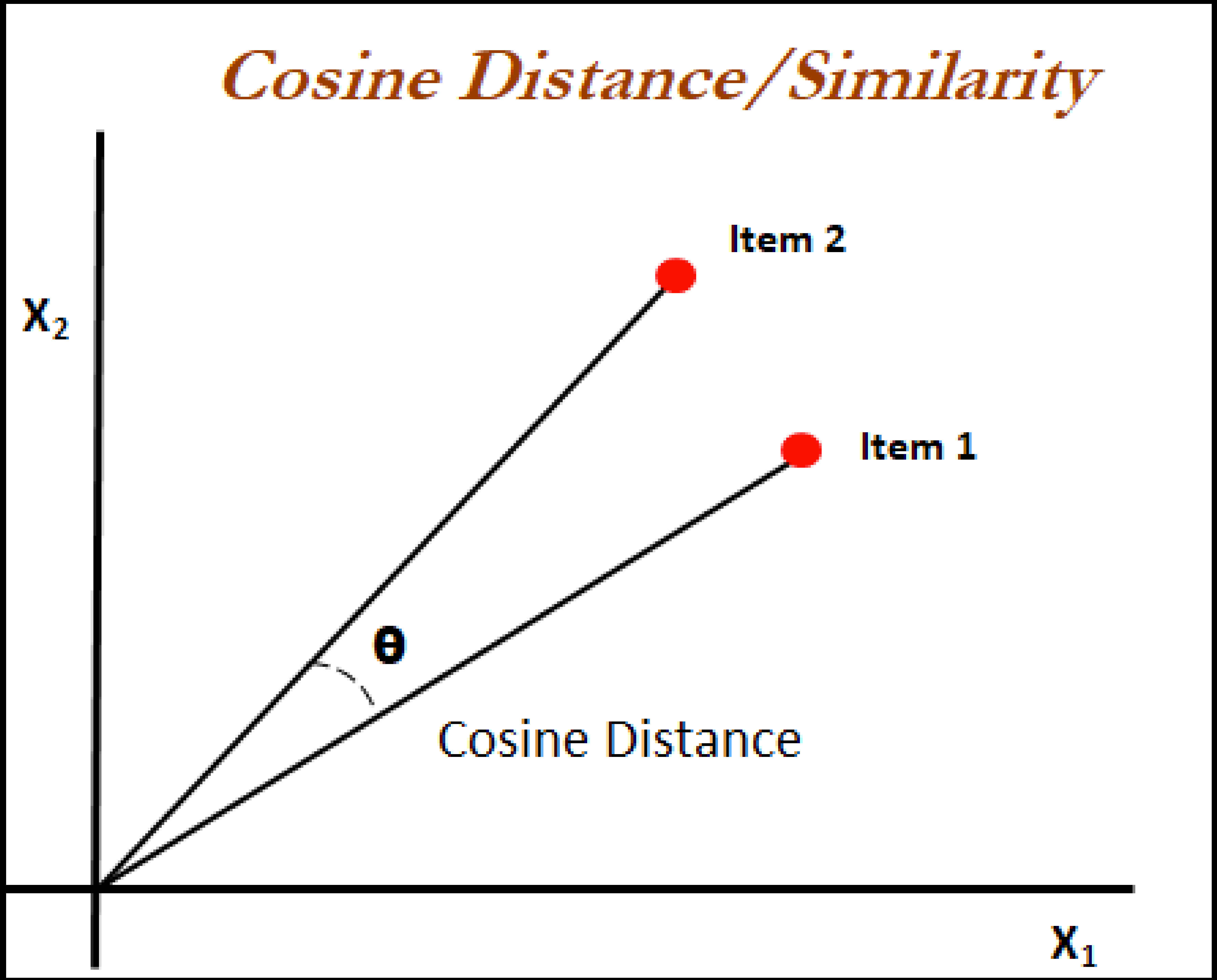
Embeddings are used to calculate similarity between inputs:
The most common distance measurement is cosine similarity

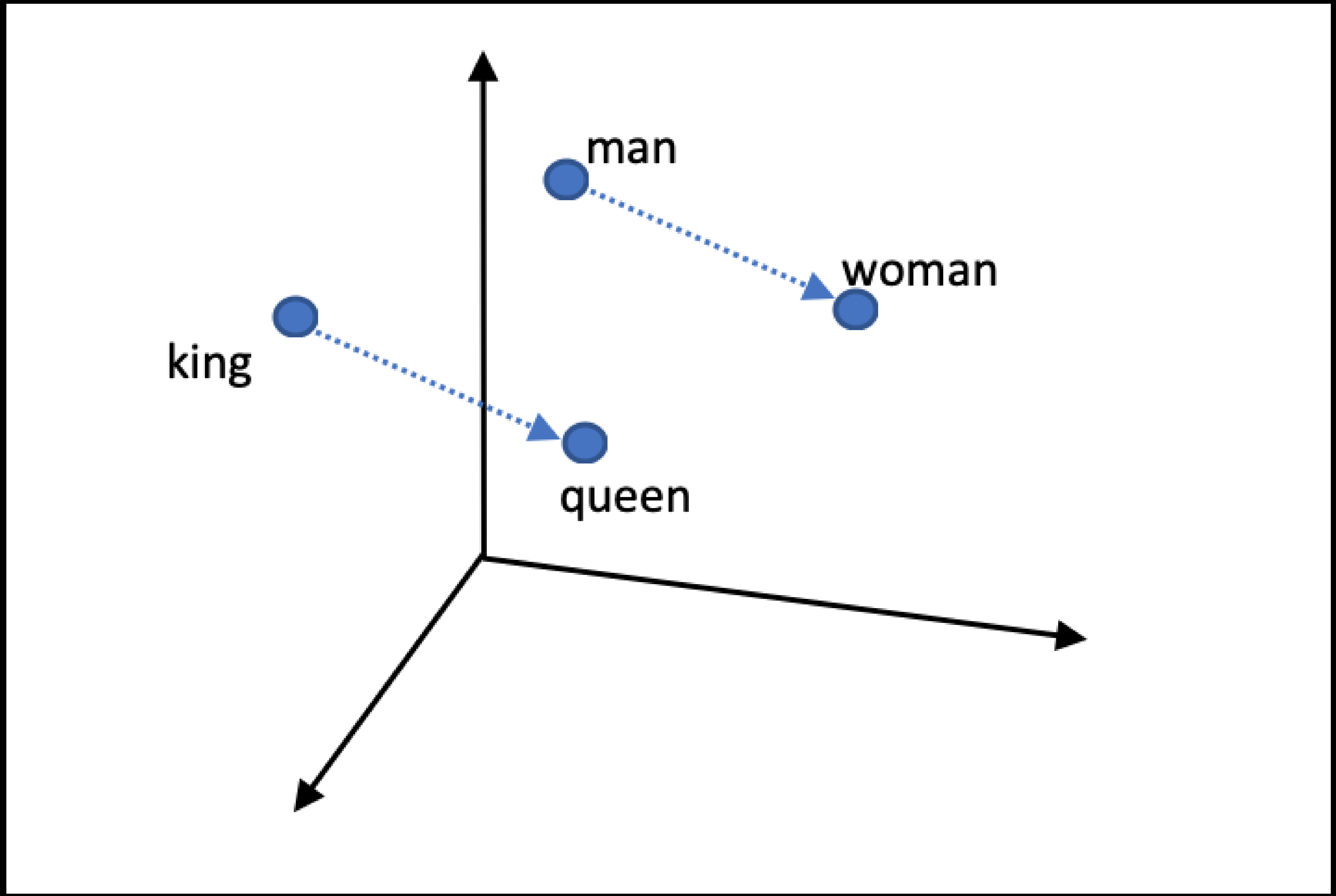


https://aka.ms/aitour/vectors

https://github.com/Azure-Samples/rag-with-azure-ai-search-notebooks/blob/main/vector_embeddings.ipynb

Vector embeddings





https://github.com/Azure-Samples/rag-with-azure-ai-search-notebooks/blob/main/vector_embeddings.ipynb

Vector Comparison

What is a vector?

Expore words from a dataset of 1000 words across two embedding models.

Target word: book Embedding model: Both (Comparison) Find word

Model: word2vec

Vector: 300 dimensions

0.044865, -0.010391, -0.017868, 0.027773, 0.055935, 0.01209, -0.017383, 0.097498, 0.034765, -0.020102, 0.09206, -0.029716, 0.08701, 0.01379, -0.057878, 0.022918, 0.002671, -0.002792, 0.052439, -0.100994, 0.057101, -0.055935, -0.014178, -0.08468, -0.098664, 0.01981, -0.036125, 0.057489, 0.022724, -0.041369, -0.078076, -0.081572, -0.10954, 0.012187, 0.080019, 0.069142, 0.036319, -0.040204, 0.090895, -0.016217, 0.010779, -0.000422, 0.010779, 0.135954, -0.052439

Most similar:

read	0.3893648604097623
paper	0.3634623893904801
write	0.35940013889130784

Model: openai

Vector: 1536 dimensions

-0.006843345705419779, -0.019184302538633347, -0.004917495418339968, -0.022664999589323997,

Most similar:

paper	0.8874017308879492
movie	0.8805337935966647
film	0.8711653176455576
letter	0.8632871648170634
record	0.8630170946356468
course	0.8629488396382509
bank	0.8628000814561154

https://pamelafox.github.io/vectors-comparison/

Movie title embeddings in OpenAI

Movie title embeddings in OpenAI

Expore embeddings for Disney movie titles from OpenAI ada-002 model.

Select a movie title: The Jungle Book See embedding

Movie title: The Jungle Book

Vector: 1536 dimensions

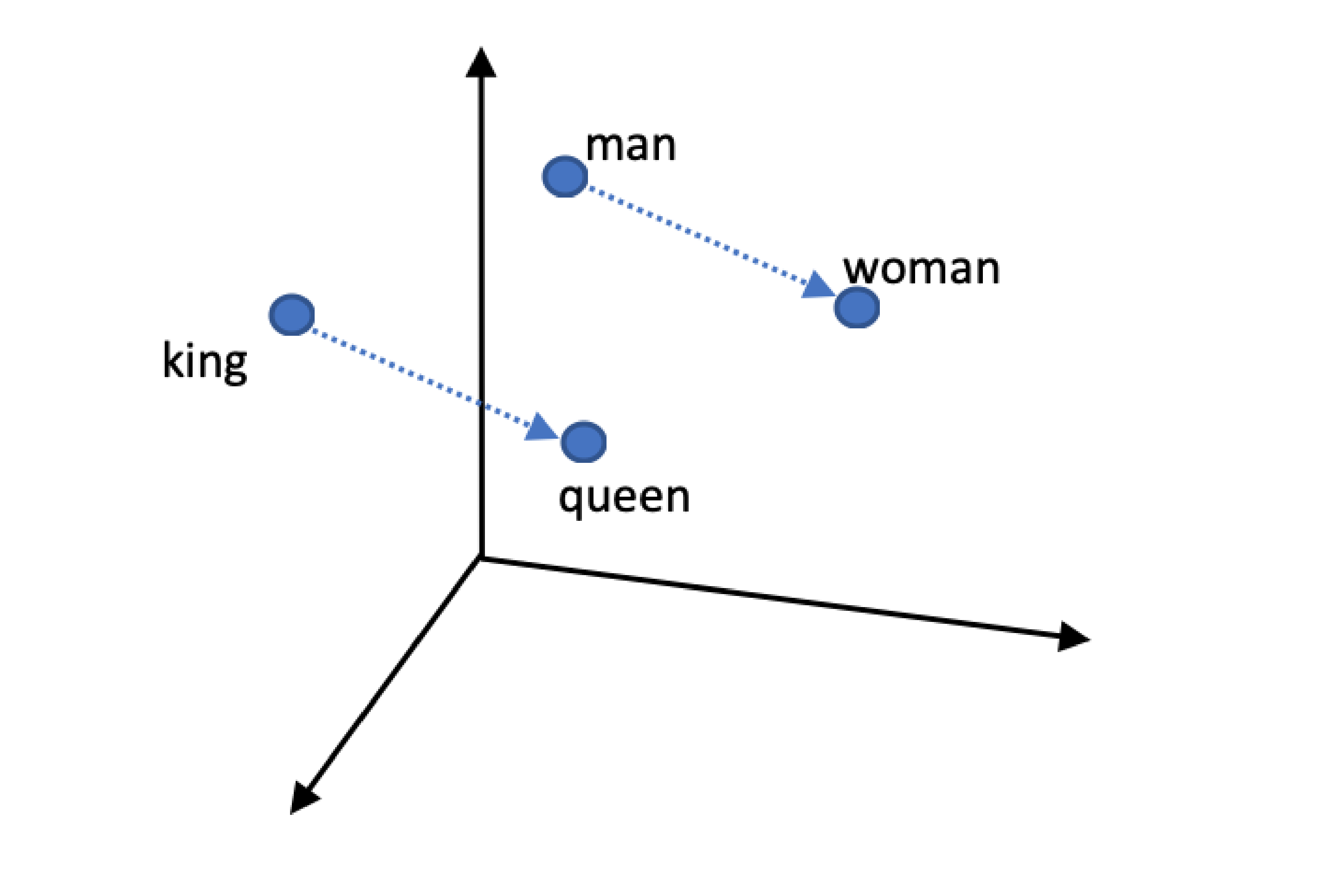
-0.009433940052986145, -0.0026398864574730396, 0.002852880861610174, -0.0006918430444784462, -0.01920369639992714, 0.017636556178331375, -0.013955017551779747, -0.024390187114477158,

Most similar:

The Jungle Book 2	0.9486278980316131
Jungle 2 Jungle	0.9236481731450379
The Lion King	0.9001141316128429
George Of The Jungle	0.8967382582947568
Tarzan	0.8928694263214043
The Fox and the Hound	0.8667384685848213
The Tigger Movie	0.8659348715821917

https://pamelafox.github.io/vectors-comparison/movies.html

Vector embeddings Lab

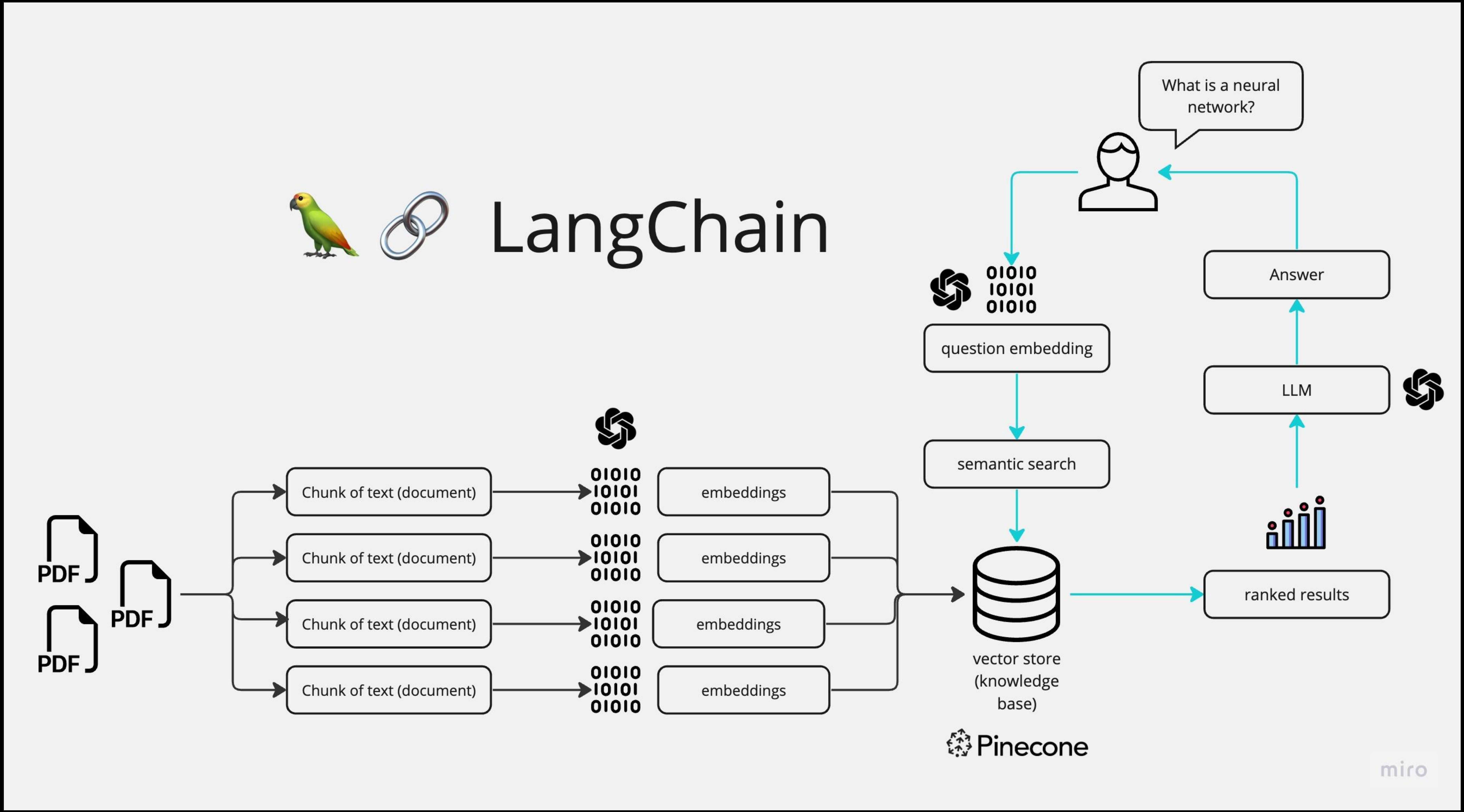


https://aka.ms/aitour/vectors

https://github.com/Azure-Samples/rag-with-azure-ai-search-notebooks/blob/main/vector_embeddings.ipynb

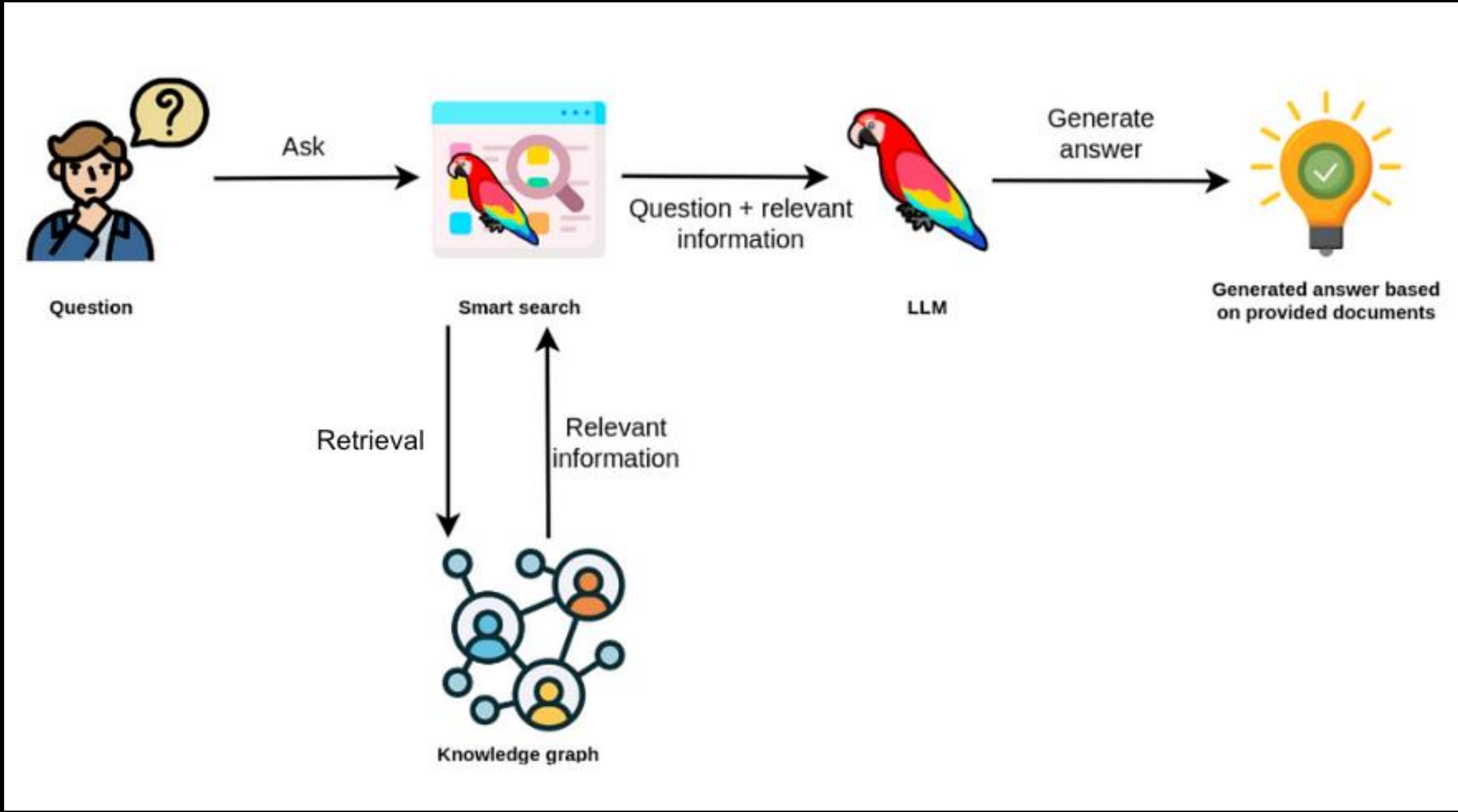
- Azure OpenAI
- RAG
- Exercise

RAG



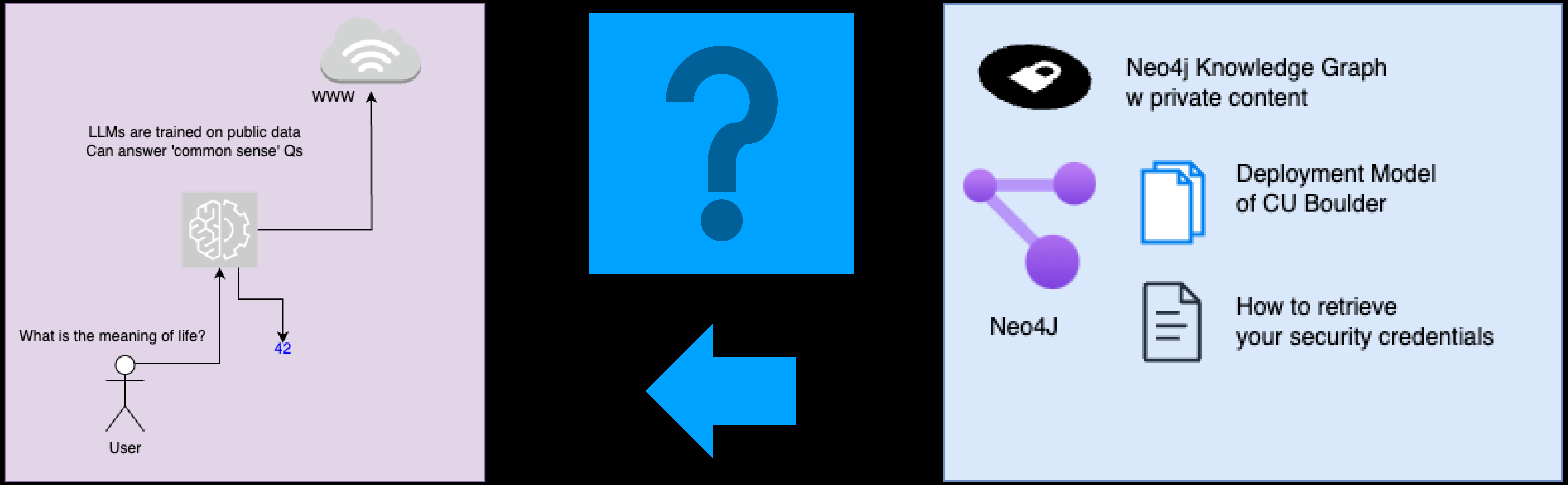
https://github.com/iportilla/ask-pdf

Graph RAG



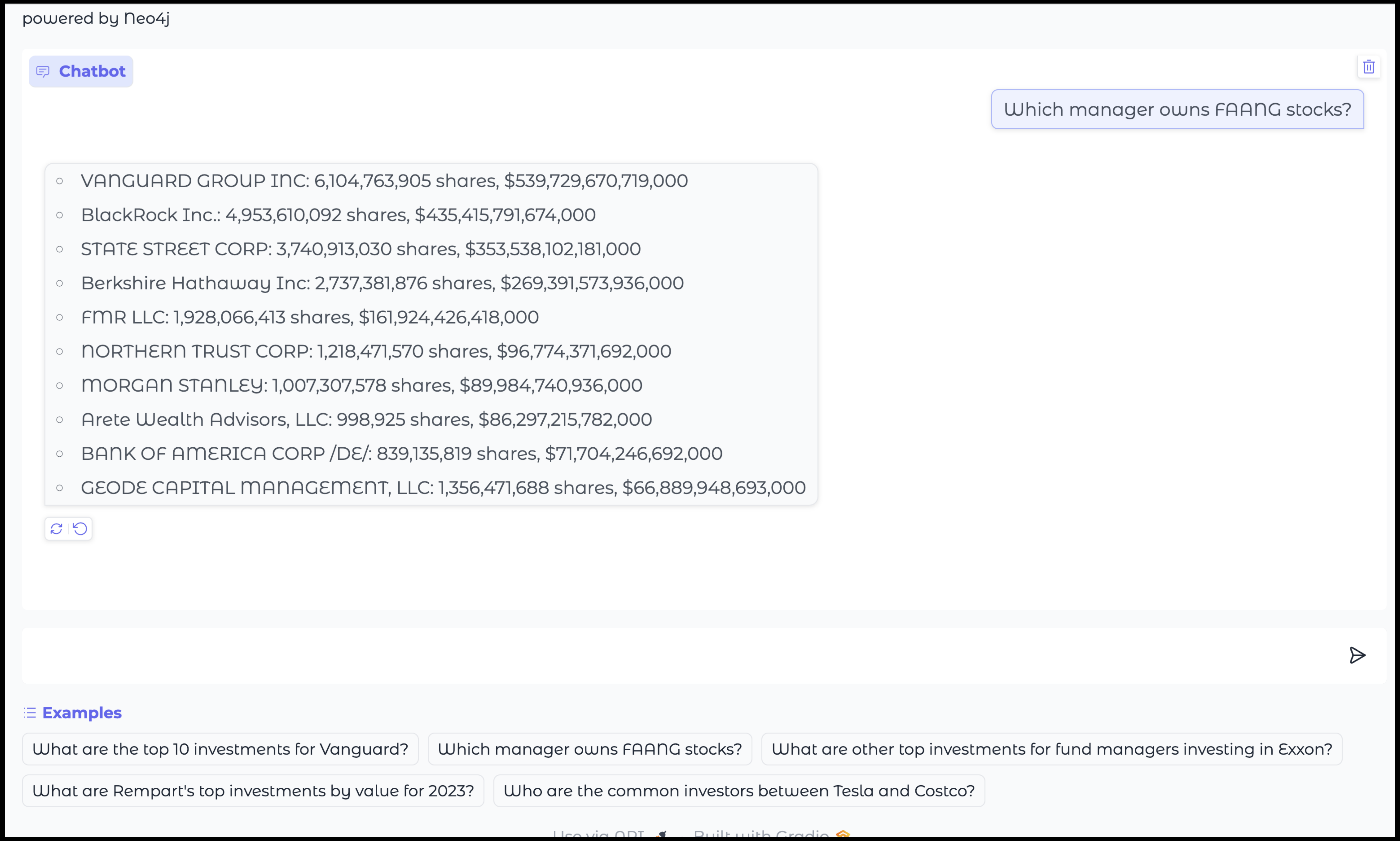
https://neo4j.com/developer-blog/global-graphrag-neo4j-langchain/

Graph RAG



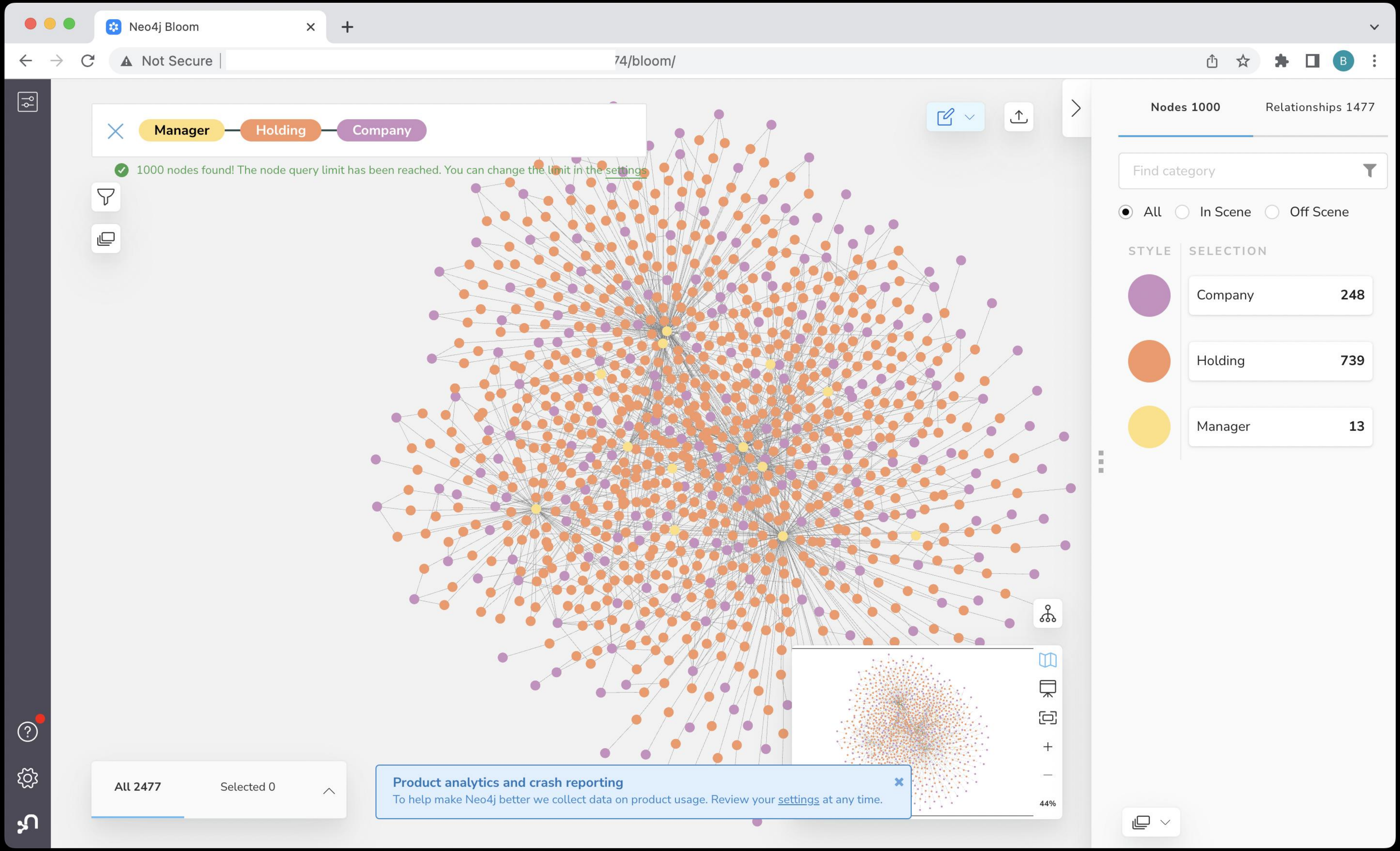
https://youtu.be/ftlZ0oeXYRE?si=XKE0oFxZg3Ox2Avt

Graph-RAG Lab



https://github.com/iportilla/genai-stack

Graph-RAG Lab



https://github.com/neo4j-partners/hands-on-lab-neo4j-and-azure