Day 3

Agenda

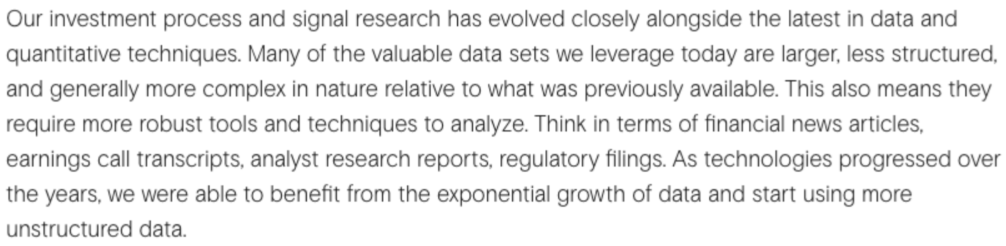
In this session, we will discuss:

- Overview of business use cases for RAG
- A workflow for RAG
- Understanding Embeddings
- Building and Managing Vector Databases
- Hands-on Implementation of Vector Database Applications

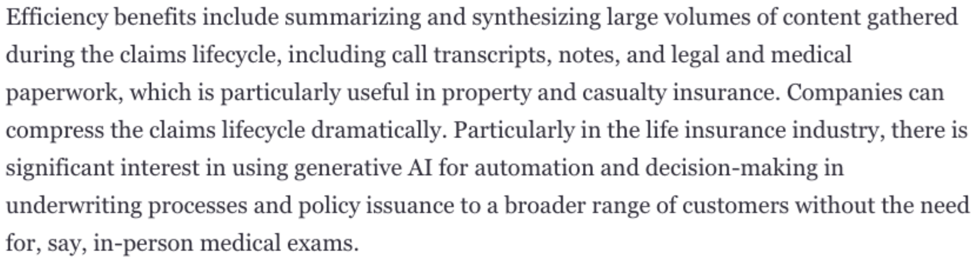
Business use cases for Retrieval-Augmented Generation (RAG)

Business Use Cases for RAG

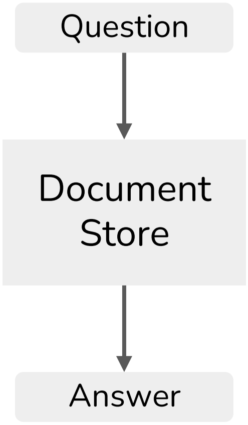
Generative AI has enabled document Q&A at scale


Our investment process and signal research has evolved closely alongside the latest in data and quantitative techniques. Many of the valuable data sets we leverage today are larger, less structured, and generally more complex in nature relative to what was previously available. This also means they require more robust tools and techniques to analyze. Think in terms of financial news articles, earnings call transcripts, analyst research reports, regulatory filings. As technologies progressed over the years, we were able to benefit from the exponential growth of data and start using more unstructured data.

Dennis Walsh, Goldman Sachs Asset Management


Efficiency benefits include summarizing and synthesizing large volumes of content gathered during the claims lifecycle, including call transcripts, notes, and legal and medical paperwork, which is particularly useful in property and casualty insurance. Companies can compress the claims lifecycle dramatically. Particularly in the life insurance industry, there is significant interest in using generative AI for automation and decision-making in underwriting processes and policy issuance to a broader range of customers without the need for, say, in-person medical exams.

Ernst & Young

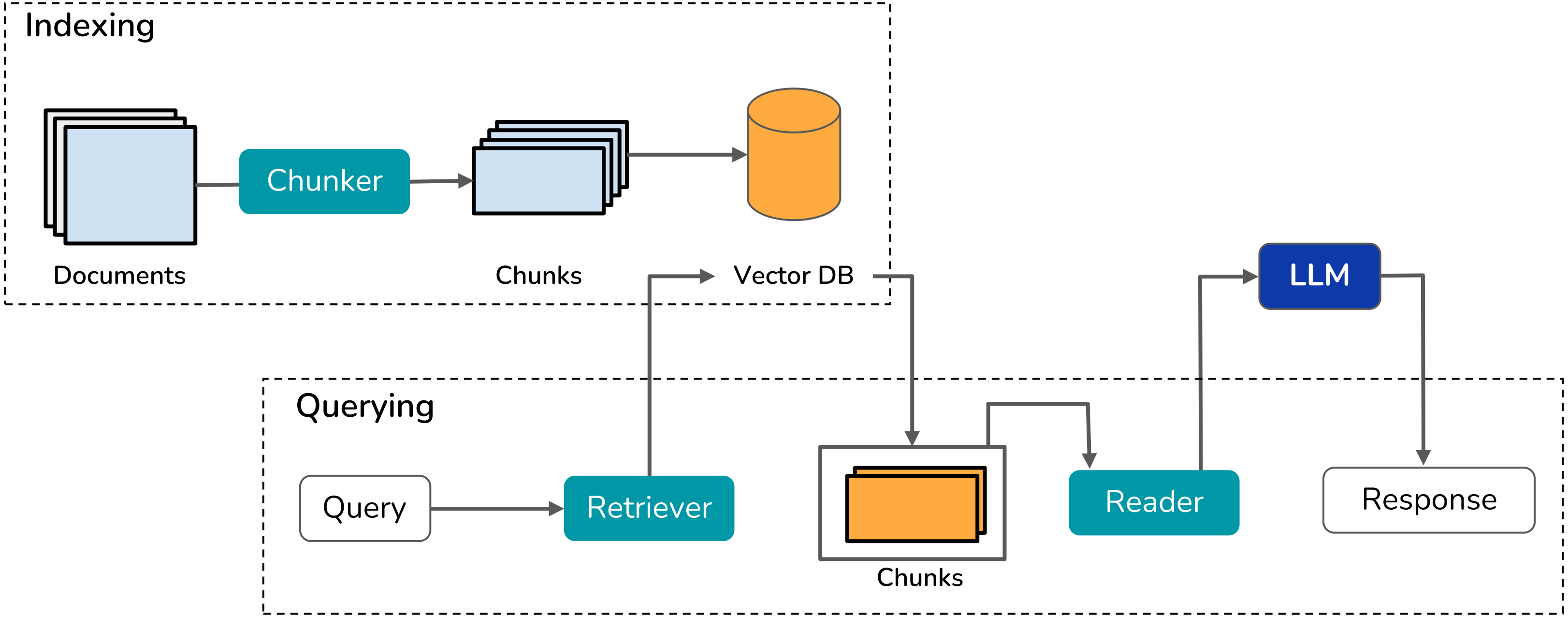


Generative AI is reducing the human effort required in synthesizing information from documents

A Workflow for RAG

A Workflow for Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) is a technique that enhances generative AI models by incorporating external data sources to improve accuracy and relevance in text generation



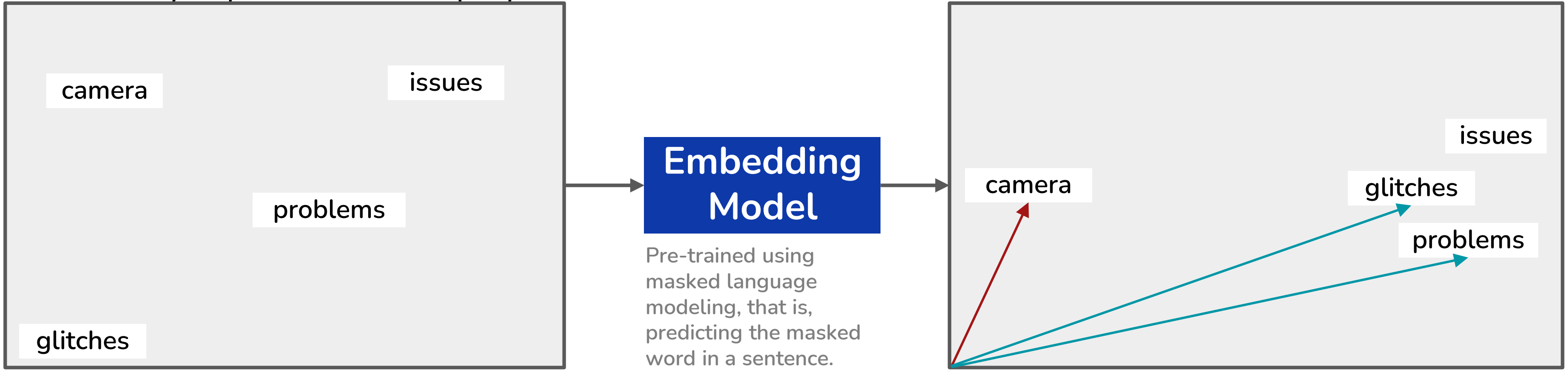
Indexing Documents for RAG

Representing Text

- The process of converting raw text data into a computer-readable format involves transforming it into numeric feature vectors.
- This conversion is known as text representation, which aims to capture both the linguistic information and the semantics of the text.
- Deep learning models are a popular method to create representations from input text referred to as *embeddings*.

Embeddings - An Introduction

- Embeddings are a type of word representation that allows words with similar meaning to have a similar representation.
- They capture semantic properties of words and relations with other words.



Pre-trained using masked language modeling, that is, predicting the masked word in a sentence.

Raw 2D Mathematical Space

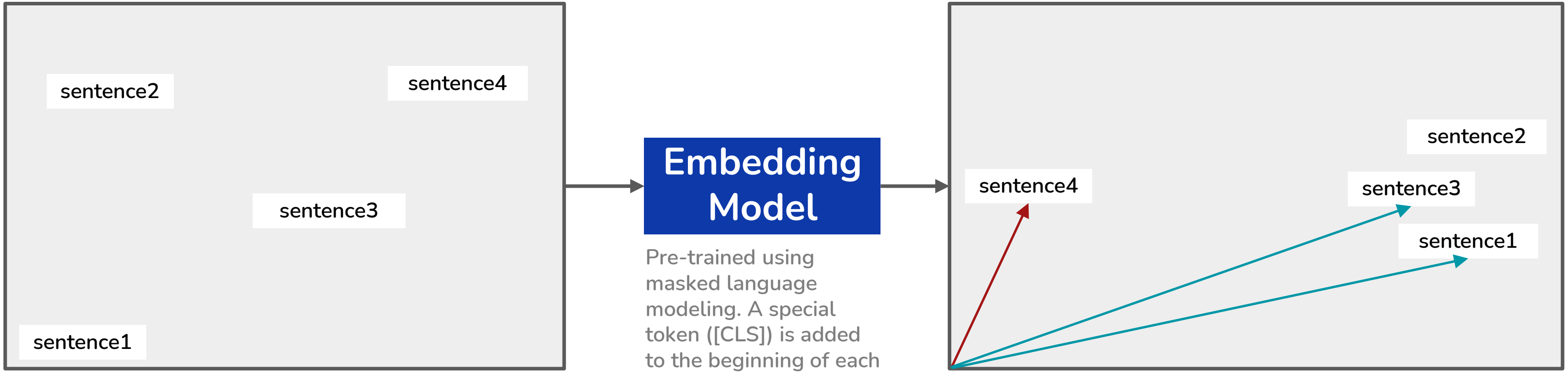
Each word is associated with a random list of two numbers. Words with similar meaning are scattered

Contextualized 2D Mathematical Space

Each word is associated with a specific list of two numbers. Words with similar meaning are clustered close to each other

Embeddings - An Introduction

- Sentence Embeddings are vector representations of whole sentences capturing their meaning.
- They are derived by averaging word embeddings or using specialized embedding models



Pre-trained using masked language modeling. A special token ([CLS]) is added to the beginning of each sentence during training.

Raw 2D Mathematical Space

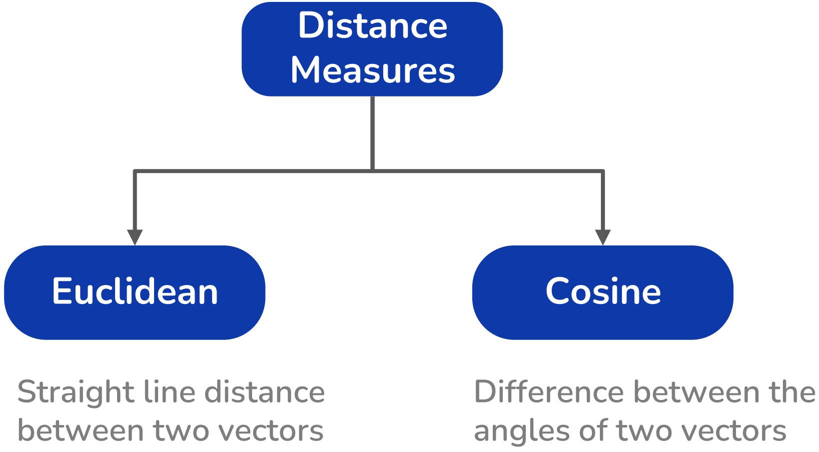
Each sentence is associated with a random list of two numbers. Sentences with similar meaning are scattered

Contextualized 2D Mathematical Space

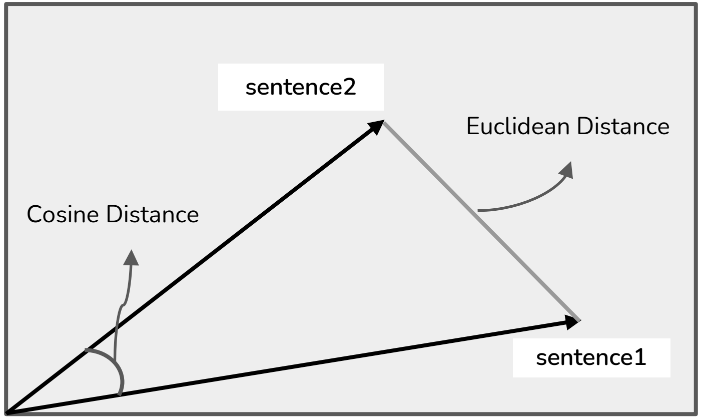
Each sentence is associated with a specific list of two numbers. Sentences with similar meaning are clustered close to each other

Using Embeddings for Similarity

A pair of texts is deemed to be similar if they are close to each other (i.e., less distant) in the embedding space.



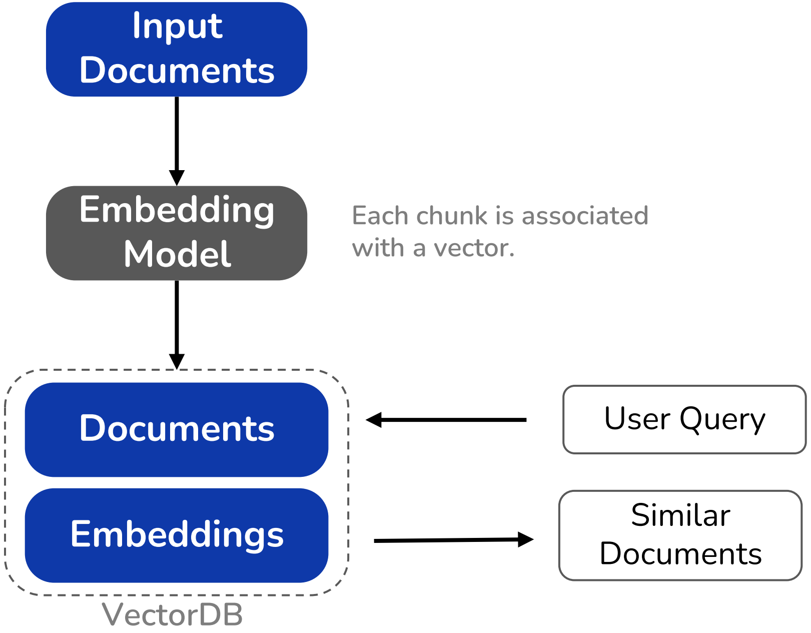
Distance Measures
Euclidean
Cosine
Straight line distance between two vectors
Difference between the angles of two vectors



Embedding Space

Vector Databases

Input documents are split into chunks of a certain size.



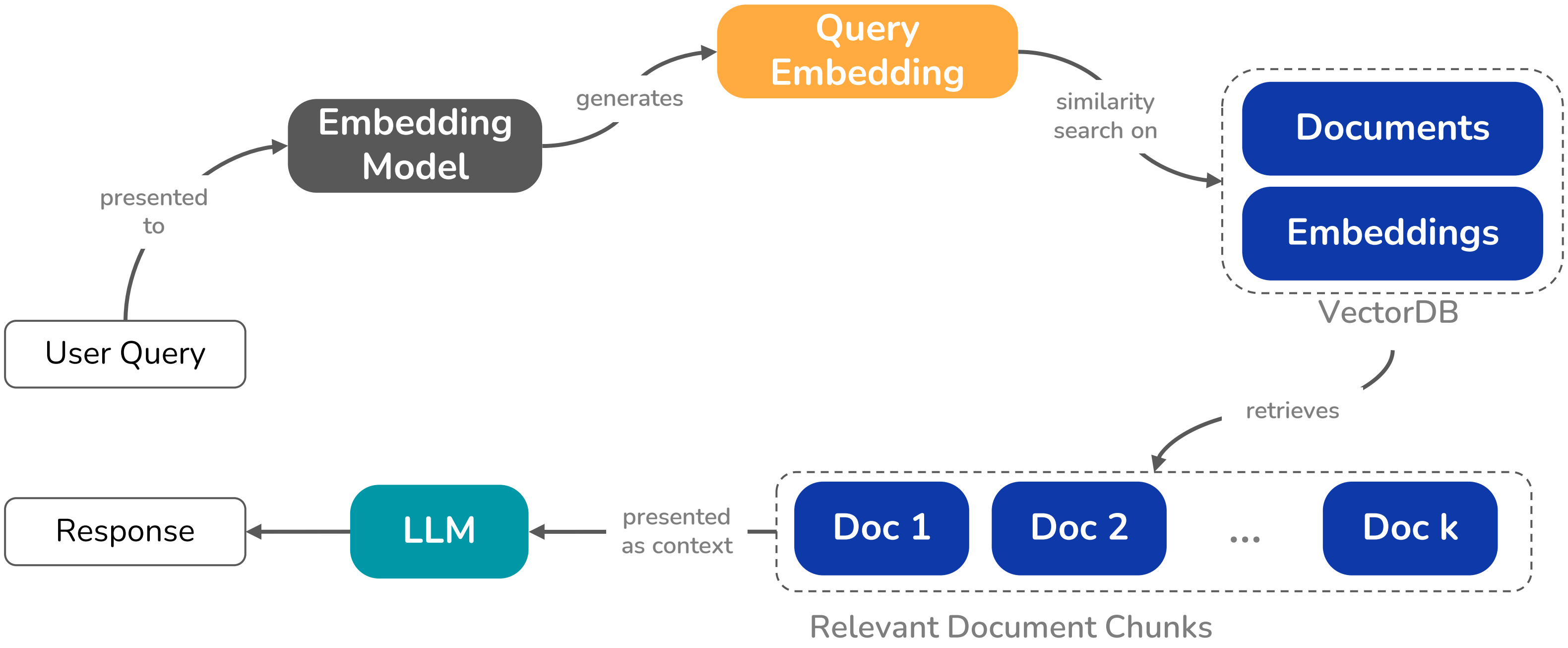
Each chunk is associated with a vector.

The vector database is pre-populated by indexing all the document chunks and the vectors created using the embedding. Indexes are organized into collections.

Retrieval & Generation

Retrieval & Generation

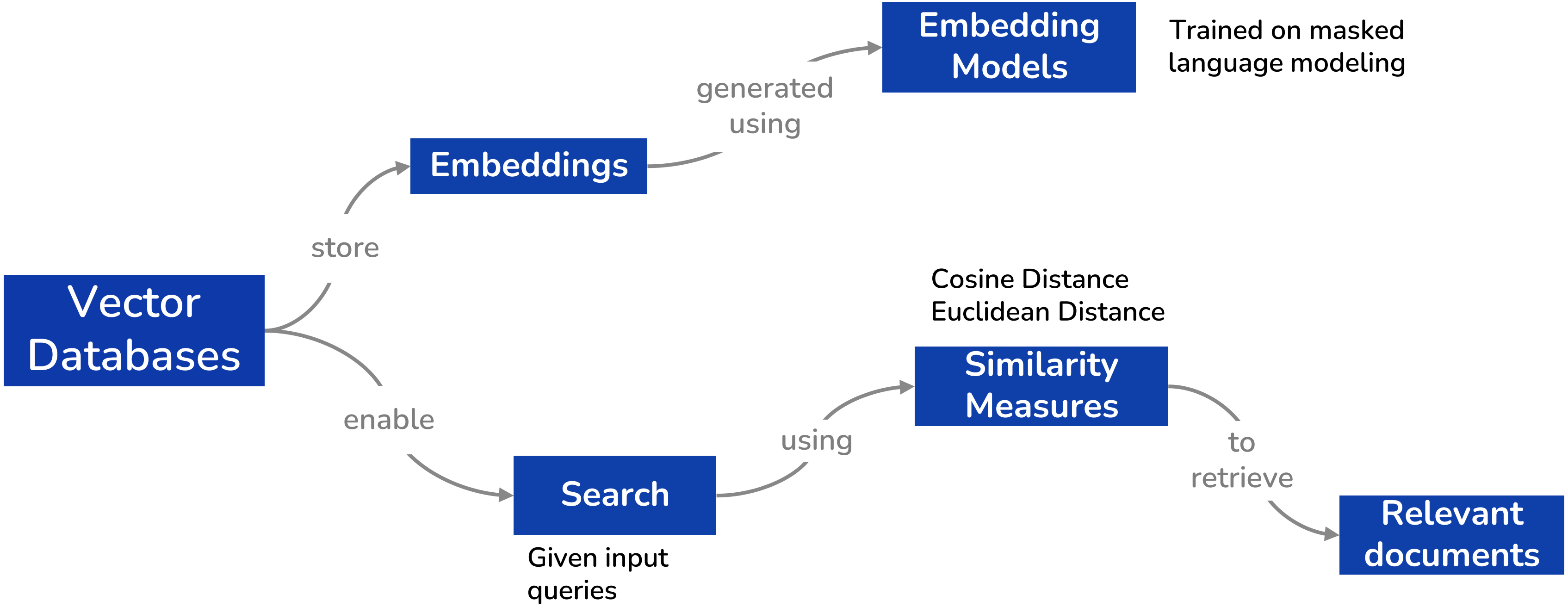
Vector databases are specialized in storing and retrieving vectors associated with unstructured data. Given input queries, the database can retrieve relevant documents using similarity search.



Vector Databases Hands-on

- Building and Managing a Vector DB
- Similarity Search

Summary



Vector Databases

store

Embeddings

generated using

Embedding Models

Trained on masked language modeling

enable

Search

Given input queries

using

Similarity Measures

Cosine Distance
Euclidean Distance

to retrieve

Relevant documents