COSINE SIMILARITY IN MACHINE LEARNING

- Introduction to cosine similarity
- What are Vectors? (RECAP)
- What is Cosine Similarity?
- Design of chat-bots?
- Vector Databases

by Natnael Teklemariam

WHAT IS COSINE SIMILARITY?

- C**osine similarity** measures how similar two things are by calculating the angle between their vector representations—ignoring their size and focusing only on their direction

Real-World Examples:

1. **Daily.dev** – Recommends articles based on what you read (not just keywords, but meaning!)
2. **Spotify/Netflix** – Suggests songs/movies similar to your taste
3. **RAG Chatbots** – Retrieves the most relevant info before generating an answer

Problem:

* Machines don’t "understand" text like humans.
* We need a way to measure semantic similarity, not just exact word matches.

Solution: Cosine Similarity – A simple yet powerful math trick to compare meanings!

Why Cosine Similarity? (The "Before & After" Story)

Before (The Problem):
❌ Keyword search fails – "Python" could mean the snake or the language.
❌ Euclidean distance misleads – A long document isn't necessarily more relevant.
❌ Machines don't understand meaning – They need a way to compare semantics, not just words.

After (The Solution):
✅ Cosine similarity compares meaning – By measuring the angle between vectors.
✅ Works in high dimensions – Perfect for AI models like LLMs.
✅ Powering real-world AI – From Spotify playlists to ChatGPT's answers.

Analogy: "Think of it like comparing two people's music tastes. It's not about how many songs they've listened to (Euclidean distance), but how alike their preferences are (cosine similarity)."

VECTORS RECAP:

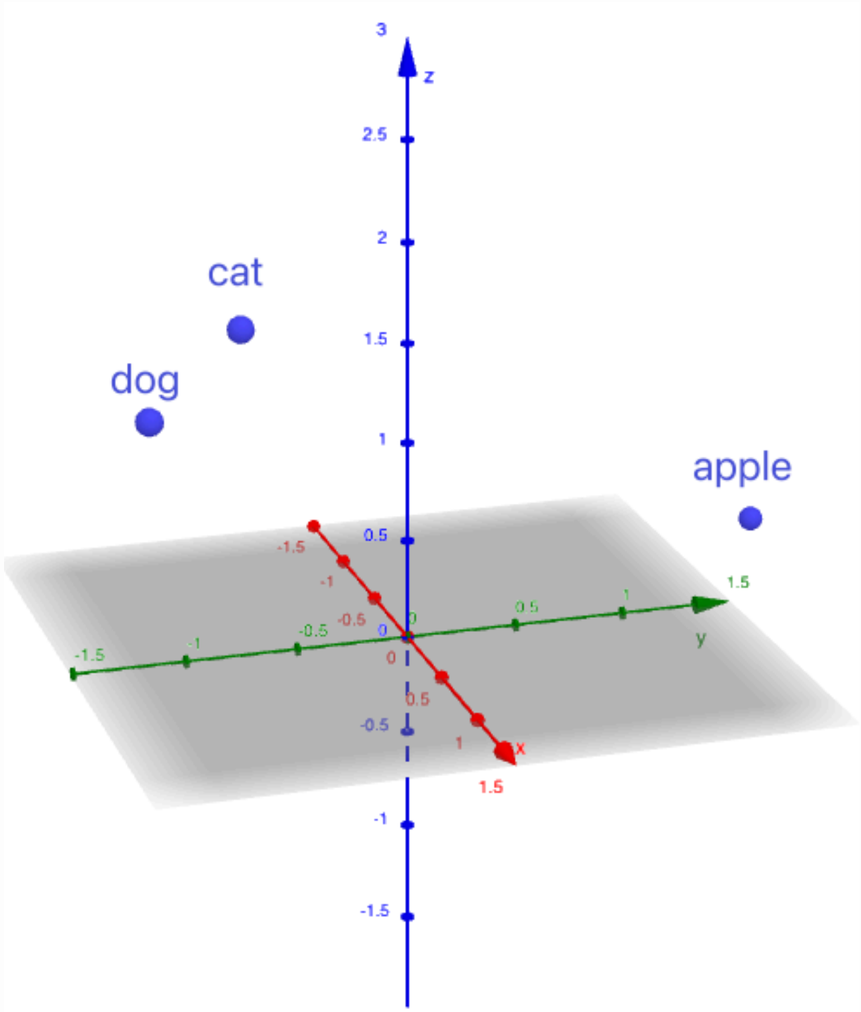
1. What’s a Vector?

Definition:
A list of numbers that represents data in multi-dimensional space.

- Analogy: Like GPS coordinates, animals, images, videos and more ...
  e.g., "Cat" = [0.7, -0.2, 0.4, ...]).

2. What’s a Vector Embedding?
Definition:
A vector that captures semantic meaning of text/images/etc., generated by AI models.

- Example:
  - "King" → [0.8, -0.3, 0.5]
  - "Queen" → [0.75, -0.25, 0.6] (close in space = similar meaning).
- Key Idea: Words with similar meanings cluster together.

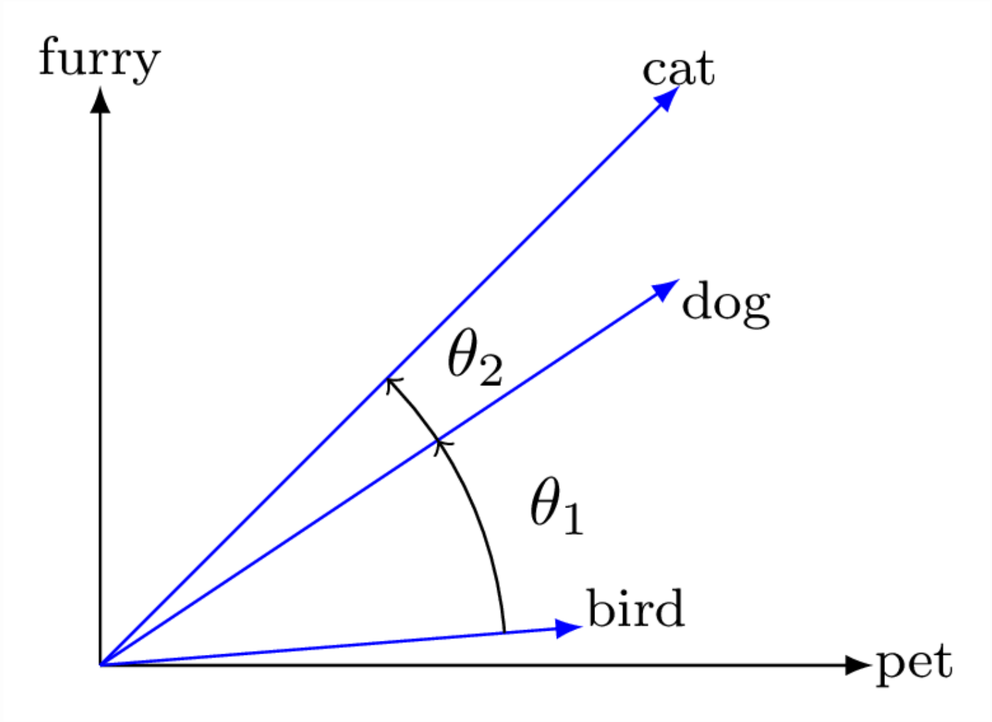


SIMILARITY BETWEEN TWO OBJECTS

Cosine similarity ignores vector length — so a short tweet and a long article can still be a match if their meaning aligns

similarity between two vectors:

- Similar: Arrows pointing nearly the same way (cosine ≈ 1).
- Dissimilar: Arrows at 90° (cosine = 0).
- Opposite: Arrows at 180° (cosine = -1)



. . . c o n t i n u e d

3. How Are Embeddings Created?

- Models: Word2Vec, BERT, OpenAI embeddings.
- Process:
    a. Model reads massive text (e.g., Wikipedia).
    b. Learns to map words/documents to vectors based on context.
- Visual: Show words plotted in 3D space (e.g., "dog" near "puppy", far from "car").

4. Why Do We Need Them?

- Machines don’t understand text → Embeddings convert words to math.
- Enables: Semantic search, recommendations, chatbots (RAG).

COSINE SIMILARITY IN ACTION: LION VS. CAT VS. DOG

1. Assign Embeddings (Simplified 2D Example)

Let’s pretend these are their vector coordinates:

- Cat: [0.8, 0.5]
- Dog: [0.7, 0.6] (similar to cat)
- Lion: [0.9, 0.2] (less similar direction)

(Note: Real embeddings have 100s of dimensions, but we’ll visualize in 2D for clarity.)

2. Calculate Cosine Similarity

Formula: $\cos(\theta) = (A \cdot B) / (\|A\| * \|B\|)$

Calculation Cosine Similarity calculation:

Cat vs Dog = $(0.8*0.7 + 0.5*0.6) / (\sqrt{(0.8^2+0.5^2)} * \sqrt{(0.7^2+0.6^2)})$

value = 0.98 (Nearly identical)

Cat vs Lion = (0.8*0.9 + 0.5*0.2) / (same denominators)

value = 0.85 (Similar but less so)

DESIGN A SMART FAQ CHATBOT WITH COSINE SIMILARITY

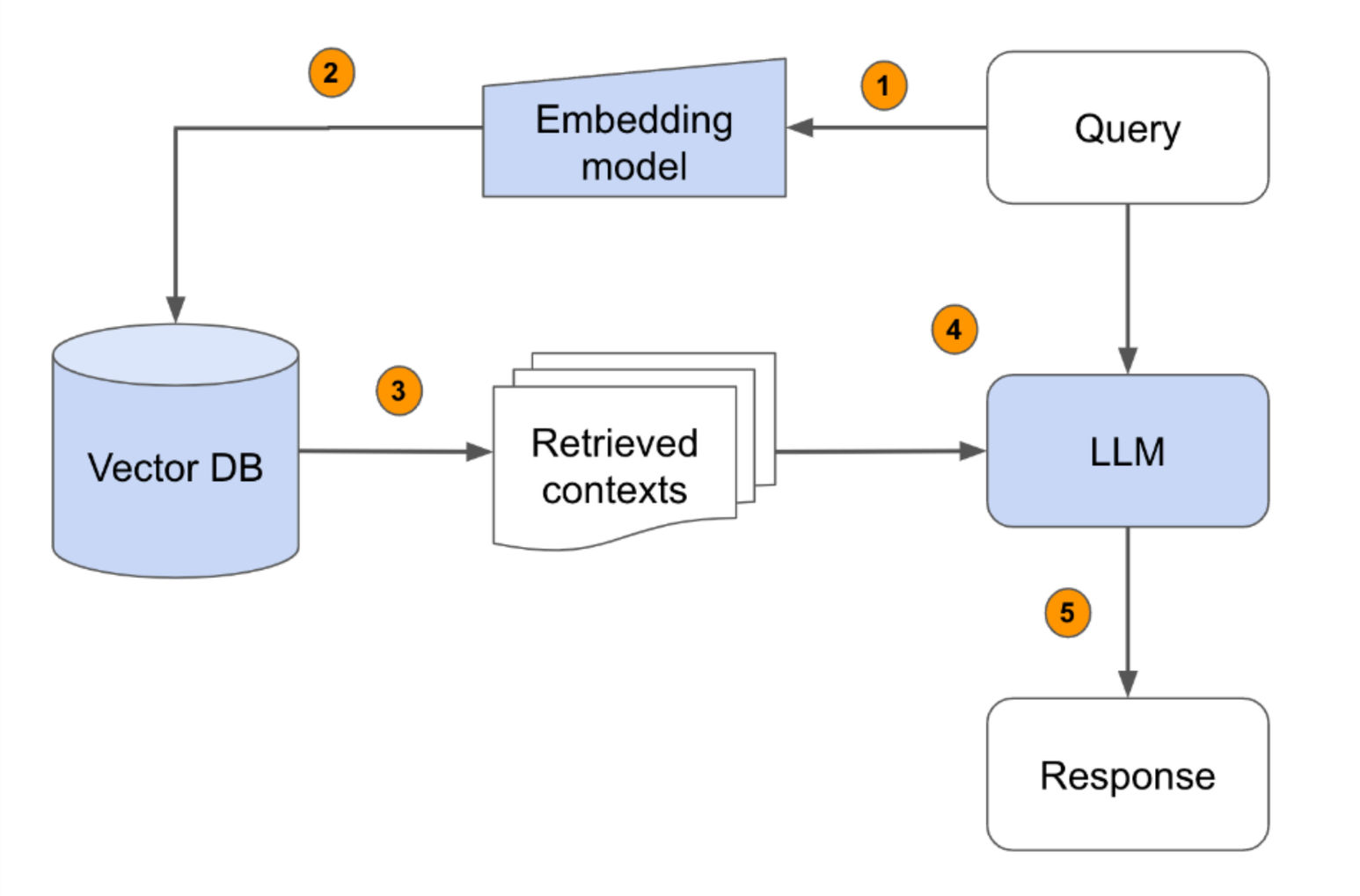
"Use vector embeddings and cosine similarity to match user questions with answers."

Step 1: Chatbot Blueprint

Architecture:

1. User Input: "How do I reset my password?"
2. Embedding Model: Convert question → vector
   (e.g., [0.4, -0.2, 0.8])
3. Vector Database: Pre-stored FAQ embeddings
   (e.g., "Reset password" = [0.38, -0.19, 0.82])
4. Cosine Similarity: Compare vectors to find closest match.
5. Response: Return best-matched answer.

CHATBOT ARCHITECTURE



VECTOR DATABASES

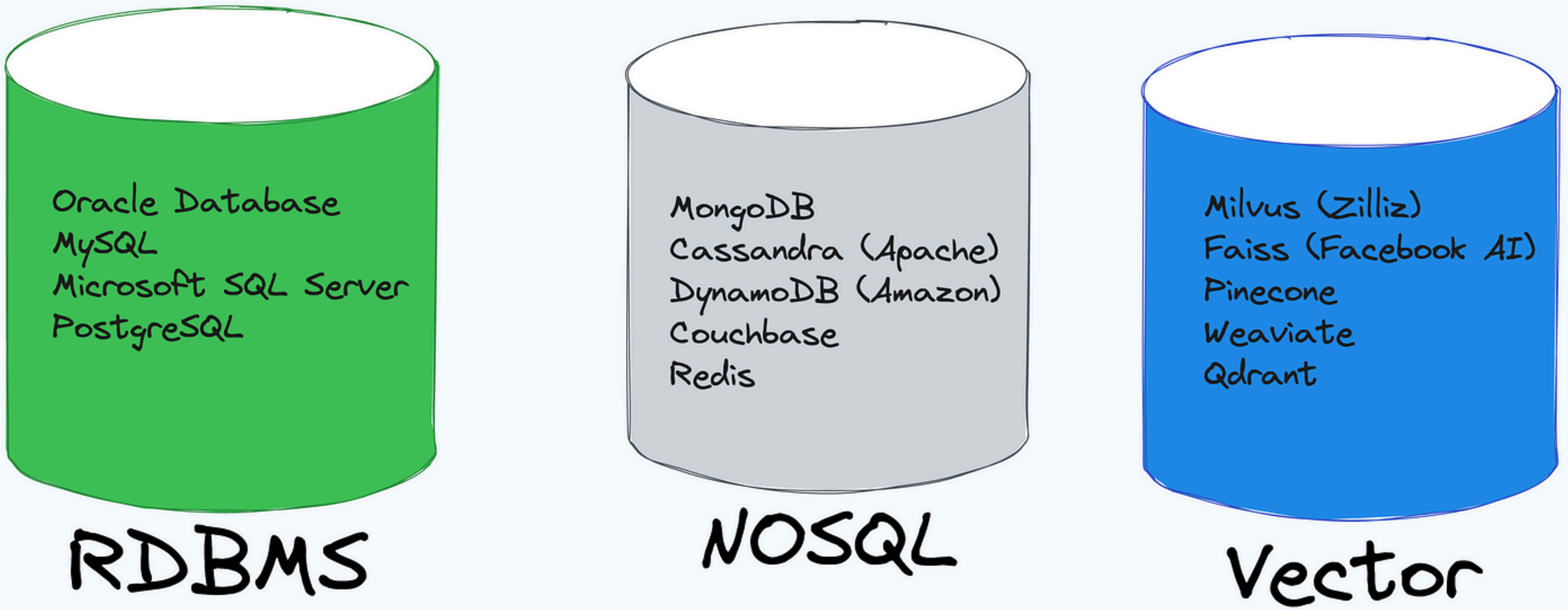
Definition:

A database optimized to store and query vector embeddings at scale.

Key Properties:

1. Native Vector Support: Handles high-dimensional data (e.g., 768D embeddings).
2. Similarity Search: Finds closest vectors via cosine/L2 distance.
3. Hybrid Storage: Can also store metadata (e.g., text, timestamps).

Databases


Oracle Database
MySQL
Microsoft SQL Server
PostgreSQL
RDBMS
MongoDB
Cassandra (Apache)
DynamoDB (Amazon)
Couchbase
Redis
NOSQL
Milvus (Zilliz)
Faiss (Facebook AI)
Pinecone
Weaviate
Qdrant
Vector

DRAWBACKS OF COSINE SIMILARITY

1. Magnitude Ignorance

    - Example: Short text ("cat") vs. long text ("a large domesticated feline") may have identical direction but different magnitudes.
    - Fix: Normalize vectors or combine with Euclidean distance.

2.High Dimensional Sparsity

    - Example: In very high dimensions (e.g., 1000D), random vectors can appear "similar" due to the curse of dimensionality.

Fix: Use dimensionality reduction (PCA, UMAP) or switch to inner product for normalized embeddings.

FINAL THOUGHT

"COSINE SIMILARITY TURNS DATA INTO MEANING, VECTOR DATABASES MAKE IT SEARCHABLE, AND RAG BRINGS IT TO LIFE—THIS TRIO IS RESHAPING AI’S FUTURE."

THANK YOU!