Retrieval-Augmented Generation

RAG combines the power of large language models with external knowledge retrieval. Instead of relying solely on parametric knowledge stored in model weights, RAG systems can access and incorporate up-to-date information from external sources.

The RAG Pipeline:

1. Query Encoding: Convert user question to vector embedding
2. Retrieval: Find relevant documents using semantic search
3. Context Augmentation: Combine retrieved context with query
4. Generation: LLM produces response grounded in retrieved facts

Benefits include reduced hallucinations, verifiable sources, and the ability to update knowledge without retraining.

Vector Databases for RAG

Vector databases are essential infrastructure for RAG systems. They store high-dimensional embeddings and enable efficient similarity search.

Popular Vector Databases:

- ChromaDB: Lightweight, Python-native, great for prototyping
- Pinecone: Managed service with enterprise features
- Weaviate: Open-source with hybrid search capabilities
- Milvus: High-performance, distributed architecture

Key features to consider:

- Indexing algorithms (HNSW, IVF, PQ)
- Metadata filtering capabilities
- Scalability and sharding options
- Integration with embedding models