Embeddings and Semantic Retrieval Foundations

This short pre-read introduces the core concepts required to understand embeddings and their role in modern AI systems.

---

1. What Are Embeddings?

Embeddings are dense numerical vector representations of data (such as text or images) that capture semantic meaning.

Instead of representing words or sentences as raw text, models convert them into high-dimensional vectors. In this space:

- Similar meanings → closer vectors
- Different meanings → farther vectors

Embeddings enable semantic search, clustering, recommendation systems, and retrieval-based AI.

---

2. Token vs Sentence vs Document Embeddings

Token Embeddings

- Represent individual words or subwords
- Used internally inside transformer models
- Contextual in modern LLMs

Sentence Embeddings

- Represent entire sentences
- Useful for semantic similarity and search

Document Embeddings

- Represent larger chunks of text
- Often created by pooling sentence-level embeddings

The choice depends on the retrieval granularity required.

---

3. Semantic Similarity

Semantic similarity measures meaning similarity rather than keyword overlap.

Example:

- “Car insurance policy”
- “Vehicle coverage document”

Though words differ, meaning is similar. Embeddings capture this relationship.

---

4. Vector Space Intuition

Embeddings exist in high-dimensional vector space.

Key intuition:

- Each text input becomes a point in space
- Similar points cluster together
- Distance between points reflects similarity

The model learns this geometry during training.

---

5. Cosine Similarity and Distance Metrics

To compare embeddings, we measure similarity using mathematical metrics.

Cosine Similarity

Measures the angle between vectors:

- 1 → highly similar
- 0 → unrelated
- -1 → opposite

Most common metric for text embeddings.

Other metrics:

- Euclidean distance
- Dot product

Choice depends on system design and embedding normalization.

---

6. Embedding Models and Use Cases

Embedding models convert text into vectors.

Common use cases:

- Semantic search
- Document retrieval
- Clustering
- Recommendation systems
- RAG pipelines

Model selection affects retrieval accuracy and performance.

---

7. Chunking Strategies

Large documents must be split before embedding.

Common strategies:

- Fixed-size chunks
- Overlapping chunks
- Section-based splitting
- Semantic splitting

Chunk size impacts retrieval precision and context completeness.

---

8. Embedding Pipelines

A typical embedding pipeline:

1. Document ingestion
2. Cleaning and preprocessing
3. Chunking
4. Embedding generation
5. Vector storage

Good pipeline design improves retrieval quality.

---

9. Embedding Quality Considerations

Embedding quality depends on:

- Model architecture
- Training data
- Domain alignment

- Chunk size
- Preprocessing

Poor embeddings lead to weak retrieval performance.

---

10. Multimodal Embeddings (Overview)

Modern embedding models can represent:

- Text
- Images
- Audio

In shared vector spaces.

This enables:

- Text-to-image search
- Image caption retrieval
- Cross-modal similarity

Before the Session

Ensure you understand:

- Basic vector concepts
- Similarity metrics
- How embeddings enable semantic search