### 1. Introduction

This project implements a **hybrid retrieval system** that combines dense-vector semantic search with sparse keyword-based retrieval. The motivation is to leverage the advantages of both methods:

- Vector search captures semantic similarity and paraphrasing.
- Keyword search provides precision with explicit term matches.
- Hybrid retrieval fuses the two to achieve more robust and stable search results.

---

### 2. System Design

- **Indexing:**
  - Document metadata and chunks stored in SQLite.
  - Embeddings for each chunk stored in FAISS (cosine similarity).
- **Retrieval methods:**
  - *Vector-only (FAISS)* – semantic similarity search.
  - *Keyword-only (BM25/FTS5)* – sparse retrieval using keyword matches.
  - *Hybrid-sum* – weighted sum of normalized vector and keyword scores (default $\alpha$=0.6).
  - *Hybrid-rrf* – Reciprocal Rank Fusion (RRF) with constant C=60.
- **Implementation:**
  - Model: sentence-transformers/all-MiniLM-L6-v2
  - Chunk size: 500 tokens, overlap: 50
  - FastAPI endpoint /hybrid_search serving the four retrieval modes.

---

### 3. Evaluation Setup

- **Dataset:**
  - Three example documents: *01_transformers*, *02_bm25*, *03_faiss*.
- **Queries:**

- At least 10 test queries covering each document, including multi-answer queries.
- **Gold standard:**
  - Each query mapped to one or more relevant documents.
- **Metrics:**
  - Recall@k, MRR@k, nDCG@k (for k=1,3,5).
- **Alpha sweep:**
  - Evaluated weighted-sum fusion at α = 0.3, 0.5, 0.7.

---

**4. Results**

**4.1 Quantitative Results**

From scores_by_method_and_k.csv:

- **Vector-only:** Strong at capturing paraphrasing, but sometimes ranks irrelevant documents.
- **Keyword-only:** Accurate when exact terms appear, but brittle with synonyms.
- **Hybrid-sum and Hybrid-rrf:** Achieve consistently higher Recall@3 and nDCG compared to single methods.

📊 *Figures:*

- Bar charts of Recall/MRR/nDCG by method (k=1/3/5).
- Line chart of Recall@3 vs α (from alpha_sweep_k3.csv).
  - Performance is stable across α = 0.3–0.7, best around α=0.6.

---

**4.2 Qualitative Results**

Based on qualitative_examples.md:

| Query | Gold Doc | Vector-only | Keyword-only | Hybrid-sum |
|---|---|---|---|---|
| *what is attention in transformers* | 01_transformers | Hits correct doc at rank 1, but also includes irrelevant (faiss, bm25). | Hits correct doc (explicit terms), also includes noise. | Correct doc boosted to top with max score; noise down-weighted. |
| *why are transformers good for long texts* | 01_transformers | Captures semantic link (*long texts* vs *long sequences*). | Hits correct doc due to keyword, weaker on paraphrase. | Correct doc remains rank 1, irrelevant docs ranked lower. |
| *self-attention explained simply* | 01_transformers | Correct doc retrieved, but noise in other ranks. | Returns only the correct doc (low recall). | Combines both, keeps correct doc top-1 while balancing recall. |

**Observations:**

- Vector-only: good semantic recall, risk of drift.
- Keyword-only: precise but limited coverage.
- Hybrid: balances precision and recall, more stable top-k results.

---

### 5. Discussion

- **Strengths of hybrid:** Robust to synonyms and paraphrases while preserving keyword precision.
- **Limitations:** Current dataset is very small; real-world performance requires scaling.
- **Future improvements:** Larger corpus, advanced rank fusion (e.g., learning-to-rank), and neural rerankers (e.g., cross-encoders).

---

### 6. Reproducibility

- **Config:** Recorded in config.json (model, chunking, α values, db path).
- **Environment:**

- Python 3.10+
- OS: Windows 11 (based on db path)
- Dependencies: requirements.txt exported with pip freeze.

- **Steps:**

1. pip install -r requirements.txt
2. Prepare docs/ folder with input texts.
3. Run python build_index.py.
4. Start API with uvicorn api:app --reload.
5. Evaluate with python eval_hybrid_plus.py.
6. Visualize results with generated CSV/figures.

---

**7. Conclusion**

The hybrid retrieval system demonstrates clear advantages over vector-only and keyword-only methods. Both quantitative metrics (Recall/MRR/nDCG) and qualitative analysis show that fusion methods (sum, rrf) produce more reliable top-k results. The project highlights the effectiveness of combining dense and sparse retrieval for robust information access.