🔥 Advanced Document Search System

This is a sophisticated hybrid document retrieval system that combines multiple AI-powered search techniques for maximum accuracy and relevance.

Core Architecture:

Frontend (React) ↔ FastAPI Backend ↔ Multiple Search Engines

🧠 How the Search Works:

1. Triple-Engine Approach:

- BM25 Engine (backend/retrieval/bm25\_engine.py): Traditional keyword-based search

- Dense Engine (backend/retrieval/dense\_engine.py): Semantic search using sentence transformers + FAISS

- Cross-Encoder Reranker (backend/retrieval/reranker.py): Neural reranking for final relevance scoring

2. Hybrid Fusion (backend/retrieval/hybrid\_engine.py):

- Combines all three engines using RRF (Reciprocal Rank Fusion)

- Configurable weights for BM25 vs dense search

- Final cross-encoder reranking for optimal results

📄 Document Processing Pipeline:

1. Multi-Format Support: PDFs, DOCX, TXT, MD, EPUB, RTF, HTML with OCR

2. Smart Chunking (backend/processing/chunker.py):

- Structure-aware segmentation preserving context

- Configurable strategies: fixed, semantic, or adaptive

3. Document Processor (backend/processing/document\_processor.py):

- Text extraction with OCR capabilities

- Language detection and boilerplate removal

🚀 Usage Flow:

1. Start: Run python start.py - launches both backend (port 8000) and frontend (port 3000)

2. Index: Upload documents through the web UI - processes and chunks them

3. Search: Query gets processed by all three engines simultaneously

4. Results: Hybrid fusion + reranking delivers the most relevant documents

5. Chat: RAG-style chat interface using search context

⚙️ Key Features:

- Real-time indexing status with progress tracking

- Comprehensive evaluation (Recall@k, MRR, nDCG, MAP metrics)

- RESTful API with FastAPI auto-documentation at /docs

- YAML configuration (configs/config.yaml) for fine-tuning

- Background processing for large document collections

- Context expansion for incomplete chunks

🎯 What Makes It Advanced:

- Vocabulary pruning in BM25 for efficiency

- FAISS indexing for fast dense retrieval

- Neural reranking with transformer models

- Dynamic chunking that preserves document structure

- Built-in benchmarking system for performance testing

The system essentially gives you Google-quality search for your private documents by combining the best of traditional and modern AI search techniques.