AI & RAG Systems Portfolio

Dr. Stephen Dietrich-Kolokouris | Production AI/ML Engineer

Professional Summary

Dr. Dietrich-Kolokouris designs and deploys production retrieval-augmented generation (RAG) systems, AI agent frameworks, and LLM-powered applications. His work spans the full stack from vector database architecture through prompt engineering to user-facing interfaces, with particular emphasis on grounded, citation-backed AI systems that minimize hallucination.

Production RAG Systems

Architecture & Design

Built end-to-end RAG pipelines using LangChain, FAISS, and ChromaDB for document-grounded question answering. Production systems feature SHA-256 manifest-based index persistence, automatic rebuild detection when source documents change, and MMR (Maximal Marginal Relevance) retrieval to balance relevance with diversity.

Technical Stack

- LangChain framework for chain orchestration, retrieval, and prompt management
- FAISS vector store with MMR retrieval (k=7, fetch_k=24, lambda_mult=0.5)
- ChromaDB for persistent, metadata-filterable vector storage
- OpenAI embeddings (text-embedding-ada-002, text-embedding-3-small)
- PyPDF, pdfplumber for document ingestion and text extraction
- RecursiveCharacterTextSplitter with optimized chunk_size=1100, overlap=160

Retrieval Quality Engineering

Implemented evidence pack formatting with source deduplication, page-level citation tracking, and automatic truncation to context window limits. Developed guardrail systems using anchored regex patterns to prevent the LLM from fabricating external citations or URLs not present in the source corpus.

LLM Application Development

Recruiter-Facing Portfolio Assistant

Designed and deployed a Streamlit-based AI assistant that represents professional qualifications to recruiters. Features include silent recruiter context extraction (roles, domains, location preferences), history-aware query rewriting, multiple action modes (chat, fit summary, outreach draft), and tone toggling between conversational and technical registers.

- GPT-4o for answer generation with GPT-4o-mini for cost-optimized auxiliary calls
- Streaming response delivery for real-time user experience
- Structured JSON extraction of recruiter intent across conversation turns
- Action modes: general chat, fit summary, outreach message drafting
- PDF transcript export with evidence file tracking via ReportLab

Agent Frameworks & Automation

Developed AI agent systems using OpenAI function calling and tool-use patterns. Projects include browser automation agents, YouTube script generation systems, and lifestyle productivity applications. Experience with multi-step agent orchestration, error recovery, and human-in-the-loop confirmation flows.

- OpenClaw framework integration for enhanced agent functionality
- Browser automation agents for web research and data collection
- YouTube Script Generator optimized for paranormal, true crime, and history content
- Multi-agent orchestration with tool selection and fallback logic

Prompt Engineering

Deep expertise in prompt design for production systems. Techniques include system prompt layering with mandatory constraints, action-mode switching, recruiter context injection, evidence pack formatting, and guardrail enforcement. Developed prompt architectures that maintain factual grounding while allowing natural conversational tone.

- System prompt architecture with layered constraints and tone control
- Few-shot and chain-of-thought prompting for complex reasoning tasks
- Guardrail design: regex-based output filtering, hallucination prevention
- Context window management: evidence truncation, history summarization
- Evaluation methodology: manual review, retrieval recall testing

Data Engineering & Vector Databases

Experience with the full data pipeline from raw document ingestion through embedding generation to indexed vector storage. Implemented manifest-based cache invalidation to avoid unnecessary re-embedding, reducing index build times by 90%+ on unchanged corpora.

- FAISS: local vector search with persistence and manifest-based rebuild logic
- ChromaDB: persistent collections with metadata filtering
- Document preprocessing: PDF extraction, recursive text splitting, deduplication
- Embedding model selection and cost optimization
- Index performance tuning: chunk size optimization, retrieval parameter sweeps