RAG with LangChain

Teaching Handbook for Production Systems

A concept-first guide to building Retrieval-Augmented Generation systems with annotated insights and production-ready strategies.

- Sivakumar Rajendran

Understanding RAG: The Complete Pipeline

Ingestion

Load and prepare source documents

Chunking

Split into meaningful segments

Embeddings

Convert to vector representations

Retrieval

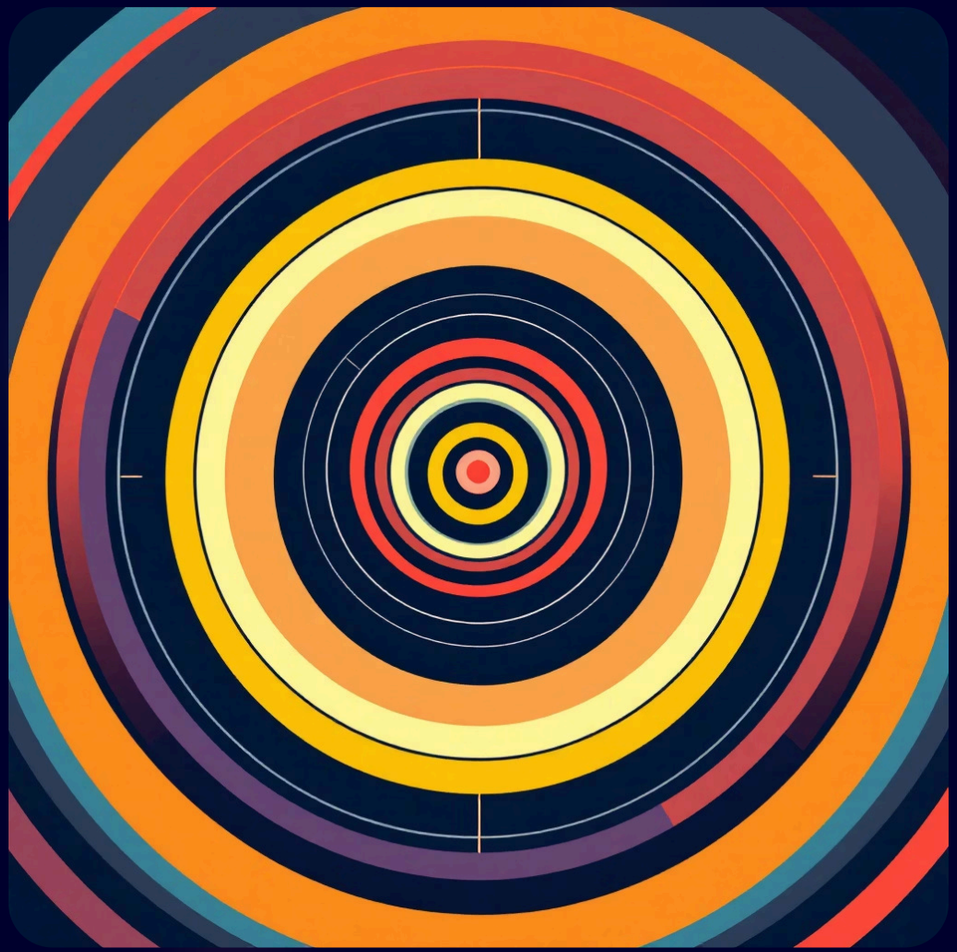
Find relevant context

Generation

Produce grounded answers

RAG grounds large language models on authoritative context, dramatically reducing hallucinations while enabling accurate citations and verifiable responses.

Core RAG Objectives



Ground Responses

Anchor LLM outputs in factual, authoritative source material

Reduce Hallucinations

Minimize fabricated information through verified context

Enable Citations

Provide traceable sources for every claim made

Critical Architecture Trade-offs

Chunk Size & Overlap

Balancing semantic coherence with retrieval precision. Smaller chunks improve specificity but may lose context. Typical range: 500-1000 tokens with 10-20% overlap.

Embedding Model Selection

Choose between speed and accuracy. Sentence transformers offer excellent balance. Domain-specific models improve relevance but increase latency.

ANN Recall vs Latency

Approximate nearest neighbor algorithms trade exactness for speed. FAISS and HNSW parameters directly impact both retrieval quality and response time.

Building Your First RAG System

01

Document Loading & Chunking

Load source documents and split into manageable segments using recursive text splitters that respect semantic boundaries.

02

Embedding & Vector Storage

Convert chunks to embeddings with HuggingFace models and store in FAISS for efficient similarity search.

03

Retriever Configuration

Configure retrieval parameters to return top-k most relevant chunks based on semantic similarity.

04

Prompt Engineering

Craft prompts that instruct the LLM to use only provided context and cite sources appropriately.

05

Chain Assembly

Connect components into a unified pipeline using LangChain's expression language for seamless data flow.

Advanced Query Optimization

Multi-Query Expansion

Generate multiple diverse queries from a single user question to increase retrieval recall. The LLM produces 3-5 variations that capture different aspects of the intent.

Cross-Encoder Re-Ranking

Apply a second-stage model to re-score retrieved documents. Cross-encoders compute query-document relevance with higher precision, dramatically improving top-k accuracy.

3x

Recall Improvement

With multi-query expansion

40%

Precision Gain

Using re-ranking models

Evaluation Strategies

Token Overlap Heuristics

Quick sanity check measuring answer fidelity by calculating the proportion of answer tokens present in retrieved context. Useful for rapid iteration.

RAGAS Framework

Comprehensive evaluation measuring faithfulness, answer relevance, and context precision. Automated metrics that correlate well with human judgment.

Human Gold Sets

Curate question-answer pairs with expert annotations. Essential for validating automated metrics and catching edge cases.

Production Deployment Essentials

Hybrid Retrieval

Combine BM25 keyword search with dense vector retrieval for superior recall across diverse query types.

Metadata Filtering

Implement namespaces and filters for multi-tenant architectures and domain-specific routing.

Caching Strategy

Cache embeddings and frequent responses. Batch encode documents and quantize models to reduce latency.

Observability

Trace requests end-to-end. Monitor latency, failure modes, and retrieval quality continuously.

Security & Compliance

Implement PII redaction, row-level security, and encryption at rest and in transit for production safety.

Data Scientist Production Checklist

Pre-Deployment

- Data deduplication and canonicalization
- PII detection and handling protocols
- Embedding model selection and validation
- ANN parameter tuning (nlist, nprobe, efSearch)
- Retrieval optimization (k-value, filters, re-ranking)

Post-Deployment

- Embedding drift monitoring
- Cost and latency budget enforcement
- Concurrency and cold start management
- Comprehensive logging and metrics
- A/B testing framework and rollback strategy

Beyond Basic RAG: Advanced Techniques

Fusion-in-Decoder (FiD)

Encode multiple documents independently, then let the decoder attend across all contexts simultaneously for richer reasoning.

GraphRAG

Build knowledge graphs from your corpus and retrieve relevant subgraphs to provide structured, relational context for complex queries.

Agent-Based Systems

Position retrieval as one tool among many—SQL queries, web search, APIs—allowing the LLM to orchestrate multi-step reasoning.

ColBERT Late Interaction

Token-level matching with late interaction for superior retrieval performance, especially on specialized domains requiring fine-grained relevance.