Post-Read Document

Advanced Exploration: Context Engineering, System Architecture, and RAG

This post-read is designed for participants who want to explore topics in greater depth beyond what could be covered during the training sessions. It provides structured directions for further study, practical experimentation, and architectural thinking.

---

1. Advanced Context Engineering

1.1 Context Window Optimization

Language models have token limits. Advanced systems must:

- Prioritize relevant information
- Remove redundant content
- Dynamically summarize history
- Compress long documents

Study topics:

- Context compression techniques
- Rolling memory vs summary memory
- Hierarchical context assembly

1.2 Dynamic Context Ranking

Instead of sending the top-k retrieved chunks directly to the model:

- Apply re-ranking models
- Use hybrid retrieval (keyword + vector)
- Add metadata filtering

Recommended areas to explore:

- Cross-encoder re-ranking
- BM25 + vector hybrid search
- Query expansion techniques

1.3 Multi-Step Context Pipelines

Advanced systems do not rely on a single retrieval step. They may:

1. Generate sub-questions
2. Retrieve for each sub-question
3. Merge and summarize
4. Produce a final answer

Study:

- Decomposition-based retrieval
- Multi-hop reasoning pipelines
- Tool-augmented reasoning

---

2. RAG Evaluation and Optimization

2.1 Retrieval Evaluation

You should measure:

- Recall (Did we retrieve relevant documents?)
- Precision (Are retrieved documents relevant?)
- MRR (Mean Reciprocal Rank)
- Top-k performance

Study:

- Offline evaluation datasets
- Synthetic query generation
- Retrieval benchmarking

2.2 Generation Evaluation

Evaluate model outputs using:

- Faithfulness
- Groundedness
- Hallucination rate
- Answer completeness

Advanced techniques:

- LLM-as-a-judge evaluation
- Automated citation verification
- Human-in-the-loop review

2.3 Reducing Hallucinations at Scale

Beyond basic prompting:

- Use strict answer templates
- Enforce citation extraction
- Add verification layers
- Implement retrieval confidence scoring

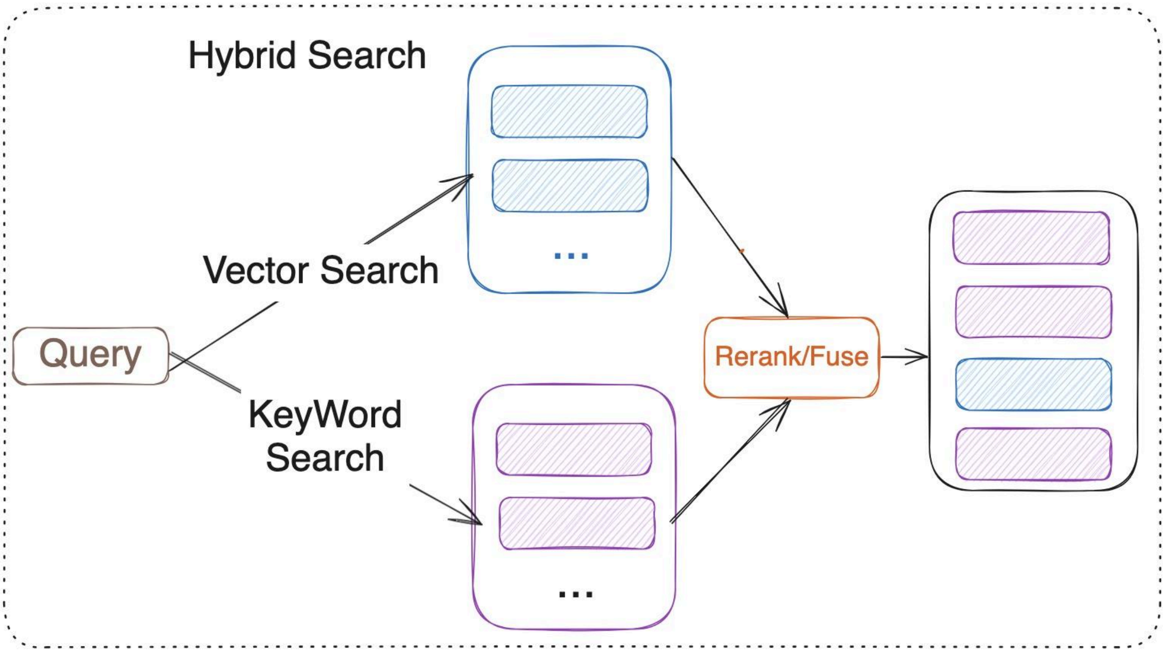
Study:

- Retrieval confidence thresholds
- Answer abstention strategies
- Response validation agents

---

3. Advanced RAG Architectures

3.1 Hybrid Search Systems



Hybrid search combines:

- Sparse retrieval (BM25, keyword)
- Dense retrieval (embeddings)

Why it matters:

- Improves recall
- Handles exact keyword matches
- Reduces retrieval misses

3.2 Multi-Modal RAG

RAG can extend beyond text:

- Images
- Tables
- PDFs with layout
- Audio transcripts

Explore:

- Multi-modal embeddings
- Table-aware retrieval
- Document layout parsing

3.3 Graph-Augmented RAG

Instead of flat documents, use:

- Knowledge graphs
- Entity linking
- Relationship-aware retrieval

Study:

- Graph databases
- Node embeddings
- Path-based retrieval

---

4. Production System Architecture

4.1 Scalability Considerations

Key areas to explore:

- Horizontal scaling of vector databases
- Caching retrieval results
- Streaming responses
- Load balancing

4.2 Cost Optimization

Advanced systems optimize:

- Embedding batch processing
- Context size
- Model selection (small vs large models)
- Retrieval frequency

Study:

- Tiered model architectures
- Caching embeddings
- Context pruning strategies

4.3 Observability and Monitoring

Production systems require:

- Query logging
- Retrieval logs

- Prompt version tracking
- Drift detection

Explore:

- RAG monitoring frameworks
- Performance dashboards
- Evaluation pipelines

---

5. RAG vs Fine-Tuning: Deeper Analysis

5.1 When to Combine Both

Some systems use:

- Fine-tuning for domain reasoning
- RAG for dynamic knowledge

Hybrid strategies include:

- Fine-tuned retrievers
- Fine-tuned instruction-following models
- RAG with domain-specialized LLMs

---

6. Agentic RAG Systems

Agentic systems extend RAG with:

- Tool calling
- Planning
- Iterative retrieval
- Self-correction

Explore:

- ReAct-style pipelines
- Planner–executor architectures
- Reflection and verification loops

---

7. Hands-On Practice Roadmap

To deepen expertise:

Step 1

Build a basic RAG system with:

- Simple chunking
- Top-k retrieval
- Direct context injection

Step 2

Improve it with:

- Overlapping chunking
- Hybrid retrieval
- Re-ranking

Step 3

Add evaluation:

- Measure retrieval quality
- Analyze hallucinations
- Tune chunk size

Step 4

Experiment with:

- Multi-hop queries
- Query decomposition
- Metadata filtering

---

8. Suggested Technical Topics for Independent Study

- Embedding model comparisons
- Vector indexing methods (HNSW, IVF, PQ)
- Cross-encoders vs bi-encoders
- Semantic caching
- Context compression algorithms
- Multi-query retrieval
- Knowledge distillation in RAG

---

9. Research Directions

For advanced learners, consider reviewing research on:

- Retrieval-augmented transformers
- Memory-augmented neural networks
- Long-context LLM architectures
- Self-reflective agents
- Retrieval-conditioned generation

Final Guidance

To move from foundational understanding to expertise:

1. Build systems rather than only reading about them
2. Measure performance quantitatively
3. Compare multiple retrieval strategies
4. Document architectural trade-offs
5. Iterate continuously