Generative AI

1. Prompt Engineering Realities
- Zero-shot isn't just "ask and get" - it's about crafting precise instructions
- Few-shot patterns need carefully curated edge cases
- Chain-of-thought prompting can hurt performance in simple tasks
Pro tip: A well-maintained prompt library is worth its weight in gold

2. RAG Architecture Insights
- Vector DB performance depends heavily on data preparation
- Chunk size optimization > embedding model selection
- Effective metadata filtering reduces hallucinations
Game-changer: Hybrid search often outperforms pure semantic search

3. Parameter Optimization Truths
- temperature is context-dependent; one size doesn't fit all
- presence_penalty shapes conversation flow more than you think
- max_tokens management is crucial for cost control
Reality check: Production systems rarely need high temperature values

4. Embedding Strategy
- Model choice should match your data characteristics
- Caching strategies are crucial for performance
- Batching embeddings can significantly reduce costs
Critical insight: Simple similarity metrics often outperform complex ones

5. Architecture Decisions
- Start simple: direct API calls
- Scale up: add middleware when needed
- Complex frameworks aren't always the answer
Hard truth: The best architecture is often the simplest one

6. Context Management
- Quality of context > Quantity of tokens
- Strategic information filtering beats compression
- Context window management affects both performance and costs
Pro move: Design your context strategy before scaling

Key Principle: Effective GenAI isn't about complexity - it's about strategic simplicity.

Revision #3
Created 8 February 2025 21:24:43 by sedawk
Updated 10 February 2025 16:36:19 by sedawk