Deep Research System Design

Preserving Domain-Specific Expert Thinking

Designing a Deep Research Assistant for Financial Services

1. Problem Framing & Understanding

The Challenge

📊 **Problem**: Senior analysts spend **35-40%** of time re-discovering existing insights

💡 **Opportunity**: Leverage **15 years** of proprietary analytical frameworks and expert reasoning

🎯 **Critical Success Factor**: System must preserve domain-specific thinking, not just retrieve documents

✅ **Available Asset**: ~**2,500** expert-curated Q&A pairs (golden dataset)

Success Criteria

1. ✅ Preserve and surface unique analytical frameworks
2. ✅ Answer complex queries requiring synthesis across reports and time periods
3. ✅ Handle ambiguous queries with proactive clarification
4. ✅ Deliver accurate, well-sourced answers maintaining expert rigor
5. ✅ Provide consistent, trustworthy responses for client-facing work

2. AI-Assisted Design Process

LLM Usage & Transparency

What We Used LLMs For:

- 🔍 Research and synthesis of state-of-the-art approaches
- 🏗️ Architecture design brainstorming and validation
- 📊 Evaluation framework design
- 📝 Document structure and content organization

Example Effective Prompts

1. *"Research state-of-the-art knowledge graph approaches for RAG systems in 2024, including Think-on-Graph, KGoT, and hallucination mitigation strategies"*
2. *"Design a multi-stage evaluation framework for financial research systems using offline golden sets and online LLM-as-judge approaches"*

What LLM Got Wrong

- ❌ **Over-simplification**: Missed importance of versioning and temporal reasoning
- ❌ **Generic solutions**: Had to redirect from generic RAG to domain-specific framework preservation
- ❌ **Evaluation gaps**: Lacked specificity for financial domain metrics

What Couldn't Be Delegated

- 🎯 Domain-specific assumptions and risk assessment
- ⚖️ Trade-off decisions based on firm priorities
- ❓ Critical questions for leadership
- 🏛️ Final architecture decisions requiring business context

3. System Architecture Overview

📄 **Document Ingestion Layer**

Research Reports | Due Diligence | Market Analysis | Decision Memos

⬇️

🔍 **Document Processing & Extraction**

Framework Extraction | Entity/Relationship Extraction | Temporal Metadata

⬇️

🕸️ **Knowledge Graph Construction**

Entities & Relationships | Frameworks (Proprietary) | Temporal Dimensions

⬇️

🔎 **Hybrid Retrieval System**

Semantic Search (50%) | Graph Traversal (35%) | Keyword Matching (15%)

⬇️

🤖 **Multi-Agent Query Processing**

Understanding → Decomposition → Retrieval → Synthesis → QA

⬇️

✍️ **Answer Generation & Context**

Framework-Aware Generation | Citations & Sourcing

⬇️

✅ **Quality Assurance & Guardrails**

4. Document Processing & Knowledge Extraction

Processing Pipeline

Stage 1: Document Ingestion

- Format normalization (PDF, Word, Markdown)
- Metadata extraction (author, date, framework type, version)
- Quality filtering (duplicates, corrupted files)

Stage 2: Framework Extraction

- Identify proprietary analytical frameworks
- Extract framework parameters and methodologies
- Capture reasoning patterns and decision trees

Stage 3: Entity & Relationship Extraction

- Named Entity Recognition (NER) for financial entities
- Relationship extraction (ownership, analysis_of, influences)
- Temporal entity extraction (events, predictions with dates)

Quality Metrics

Metric	Target
Extraction Accuracy	>90%
Framework Coverage	>85%
Temporal Alignment	>95%
Processing Time	<2 min/doc

5. Information Retrieval System Design

Hybrid Retrieval Architecture

1. Semantic Search (50%) - Fine-tuned embeddings, FAISS vector store
2. Knowledge Graph Traversal (35%) - Cypher/SPARQL queries, multi-hop reasoning
3. Keyword Matching (15%) - BM25 algorithm, metadata filtering

Retrieval Fusion & Metrics

Weighted Combination: Semantic 50% + Graph 35% + Keyword 15%
Re-ranking: Cross-encoder model, framework relevance, temporal recency

Metric	Target
Recall@10	>90%
Precision@5	>85%
MRR	>0.8
Framework Match	>75%
Latency (P95)	<500ms

6. Context Engineering & Answer Generation

Context Structuring

Multi-Source Context:

1. Primary sources (top retrieved documents)
2. Framework context (analytical framework definitions)
3. Temporal context (historical evolution of queries)
4. Expert reasoning (captured patterns from corpus)

Context Prioritization: Recency → Authority → Framework alignment → Citation network

Answer Generation Approach

Framework-Aware Generation:

- Inject framework definitions, use framework-specific templates
- Maintain expert voice and terminology

Structured Output:

1. Executive Summary
2. Key Findings (with citations)

Quality Metrics

Metric	Target
Answer Relevance	>4.0/5.0
Citation Accuracy	>95%
Framework Adherence	>80%
Expert Voice	>4.0/5.0
Confidence Calibration	>0.7

7. Agentic Multi-Step System Design

Multi-Agent Architecture Flow

Query Input → **Understanding Agent** (Intent classification, ambiguity detection)
↓
Decomposition Agent (Break into sub-queries, parallel vs. sequential)
↓
Retrieval Orchestrator (Coordinate multi-modal retrieval, merge results)
↓
Synthesis Agent (Combine information, apply framework reasoning)
↓
Quality Assurance Agent (Fact-checking, hallucination detection)
↓
Final Answer

Agent Responsibilities & Metrics

Agent	Responsibilities
Understanding	Intent classification, ambiguity detection
Decomposition	Break queries into sub-queries
Retrieval Orchestrator	Coordinate multi-modal retrieval
Synthesis	Combine information, apply frameworks
QA	Fact-checking, hallucination detection

Reliability Metrics:

- Task Completion >90%
- Coordination Accuracy >95%
- Error Recovery >80%

8. Quality Assurance & Hallucination Mitigation

Multi-Stage Guardrails

Stage 1: Pre-Generation

- Verify query answerable, check context retrieval

Stage 2: During Generation

- Grounding verification, framework consistency, temporal consistency

Stage 3: Post-Generation

- Fact extraction & KG verification, citation accuracy, confidence calibration

Hallucination Detection Strategies

1. KG Verification - Extract claims → Verify entities/relationships in KG
2. Source Attribution - Every claim must have citation, verify sources contain claim
3. Consistency Analysis - Compare with retrieved documents, detect contradictions
4. Self-Evaluation - LLM self-assessment, uncertainty

Golden Dataset & Metrics

Training & Evaluation:

- Fine-tune on golden Q&A pairs, learn patterns, calibrate thresholds

Metric	Target
Hallucination Rate	<5%
False Positive Rate	<2%
Citation Accuracy	>98%
Verification Coverage	>90%

9. Evaluation Framework

Offline Evaluation (Golden Set)

Dataset Split:

- Training 2,000 (80%)
- Validation 250 (10%)
- Test 250 (10%)

Metrics per Component:

- **Retrieval**: Recall@10, Precision@5, MRR, Framework match rate
- **Answer Generation**: BLEU/ROUGE, Citation accuracy, Framework adherence
- **Overall**: End-to-end accuracy, Hallucination rate, User satisfaction

Online Evaluation (LLM-as-Judge)

Judge Model: GPT-4 or Claude (fine-tuned on financial domain)

Evaluation Criteria (5-point scale):

Domain Expertise Preservation Metrics

Framework Usage Rate >75%
Methodology Consistency Agreement with history
Expert Reasoning Capture Human evaluation
Ontology Coverage >90%

10. System Performance & Monitoring

Performance Characteristics

Latency:

- Simple <3s
- Complex <15s
- Very Complex <45s (P95)

Throughput:

- 100+ concurrent users
- 10,000+ queries/day
- 5x peak capacity

Accuracy:

- Quality score >4.0/5.0
- Hallucination <5%
- Citation accuracy >95%

Monitoring & Observability

System Health:

Business Metrics

- Query volume by type
- User adoption
- Answer quality trends
- Framework usage

Failure Mode Detection & Recovery

Error Detection:

- Low confidence → Flag
- High hallucination spike → Alert
- Retrieval failures → Fallback
- Agent failures → Escalate

Recovery:

- Automatic retry with backoff
- Fallback to simpler methods
- Human-in-the-loop escalation
- Circuit breakers

11. Key Trade-offs & Design Decisions

Decision	Rationale	Trade-off
Hybrid Retrieval (Semantic + Graph + Keyword)	Maximizes recall while maintaining precision	Increased complexity but better results
Multi-Agent Architecture	Better handling of complex queries and error isolation	Coordination overhead vs. modularity benefits
Multi-Stage Guardrails (moderate thresholds)	Balance between safety and usability	Some valid answers may be flagged vs. higher safety
Automated Framework Extraction (with human validation)	Scalability with quality assurance	Some frameworks may need manual correction
Hybrid Evaluation (LLM + Human)	Scalable evaluation with human oversight	LLM evaluation may miss nuances vs. cost-effectiveness
Quality over Speed	Financial research prioritizes accuracy	Slightly slower responses vs. higher trust

12. Complex Query Examples

Example 1: Exploratory Query

Query: *"What do we know about renewable energy investments in Southeast Asia?"*

- Identified as exploratory synthesis query
- Decomposition into sub-queries (regions, sectors, time periods)
- Graph traversal + semantic search finds 12 reports (2020-2024)
- Synthesis with framework (Risk-Return Analysis)
- Output: Structured answer with citations and temporal evolution

Example 2: Methodological Query

Query: *"How have we historically evaluated regulatory risk in emerging markets?"*

- Identifies as framework/methodology query
- Retrieves framework definitions and historical

Example 3: Ambiguous Query

Query: *"Tell me about the European market"*

- Ambiguity detected (which market? what time period? what aspect?)
- Clarification generated: *"Which European market? (Equity, Fixed Income, Credit, etc.) What time period? What specific aspect?"*
- System waits for clarification before proceeding

13. Implementation Roadmap

Phase 1: Foundation (Months 1-3)

Deliverable: MVP with basic query answering

- Document processing pipeline, Basic knowledge graph construction
- Simple retrieval system (semantic search), Initial golden dataset annotation

Phase 2: Intelligence (Months 4-6)

Deliverable: System with framework awareness

- Framework extraction and ontology construction, Hybrid retrieval system
- Basic answer generation with citations, Offline evaluation framework

Phase 3: Advanced Capabilities (Months 7-9)

Deliverable: Production-ready system for pilot

- Multi-agent architecture, Advanced hallucination detection
- Temporal reasoning, Online evaluation (LLM-as-judge)

Phase 4: Production & Scale (Months 10-12)

Deliverable: Production system with full observability

- Performance optimization, Monitoring and alerting

14. Assumptions & Critical Questions

Key Assumptions (8)

1. Data Quality: Corpus well-structured with metadata
2. Expert Availability: Senior analysts available for annotation/validation
3. Infrastructure: Scalable cloud infrastructure available
4. Adoption: Analysts willing to use system once trust established
5. Regulatory Compliance: System can meet financial services requirements
6. Golden Dataset Quality: 2,500 Q&A pairs adequately represent domain
7. LLM Capabilities: Foundation models fine-tunable for financial domain
8. Framework Stability: Analytical frameworks remain relatively stable

Critical Questions for Leadership

1. What hallucination rate is acceptable for different use cases?
2. How to handle conflicting information from different time periods?
3. What is acceptable latency for complex queries?
4. What are compliance requirements for AI-generated client content?
5. What is budget for ongoing human annotation and validation?

Key Risks & Mitigation

Hallucination → Multi-stage guardrails

15. Conclusion & Next Steps

Key Design Principles

1. **Domain Expertise First**: Every component preserves expert thinking
2. **Quality over Speed**: Accuracy and trust prioritized
3. **Transparency**: Clear citations and confidence scores
4. **Continuous Improvement**: Evaluation-driven development

Success Factors

- Comprehensive evaluation framework
- Strong hallucination mitigation
- Framework-aware processing
- Multi-stage quality assurance

Immediate Next Steps

1. Validate assumptions with leadership
2. Begin golden dataset expansion and annotation
3. Build document processing MVP
4. Design detailed knowledge graph schema
5. Set up evaluation infrastructure