

NLP BASED AGENTIC RAG SYSTEM FOR DOCUMENT QUERY ANALYSIS

Presented By:

Dokala Manoj Kumar-AV.SC.U4AIE23110

Talluri Ranga Sai Varun - AV.SC.U4AIE23141

Telagamsetty Viswajith Gupta - AV.SC.U4AIE23144

INTRODUCTION

- With the rapid growth of unstructured textual data such as PDFs, reports, policies, and research documents, extracting accurate information has become challenging.
- Traditional keyword-based search systems fail to capture semantic meaning and contextual relevance.
- Recent advancements in Natural Language Processing (NLP) and Large Language Models (LLMs) have enabled intelligent document understanding.
- Retrieval-Augmented Generation (RAG) combines information retrieval with generative models to produce context-aware responses.
- This project focuses on building an NLP-based Agentic RAG System that can intelligently analyze documents and answer user queries.

PROBLEM STATEMENT

- Users often need precise answers from large and complex documents.
- Manual searching is time-consuming and inefficient.
- Existing document QA systems may generate incorrect or hallucinated answers
- Existing systems struggle with:
 - Understanding user intent
 - Handling long documents
 - Providing trustworthy, source-based answers

LITERATURE REVIEW

| Limitation of Exixsing Literature | How we overcomes that |
|---|---|
| Single Retrieval Method: Most papers use only vector search (semantic) or keyword search (BM25), not both | Implements hybrid retrieval combining BM25 (keyword matching) + vector search (semantic) + cross-encoder reranking for superior relevance |
| No Conversation Memory: Each query treated independently | Implements session-based memory with conversation history tracking (last 20 messages) |
| Static Responses: Systems generate single answer without quality verification | Implements self-critique agent loop (0-3 iterations) where critic model reviews and refines answers iteratively for accuracy |
| Limited Mode Options: Systems either strict (docs only) OR hybrid (docs + web), not both | Provides dual-mode architecture: Strict mode (document-only, less hallucination) and Hybrid mode (docs + web search) selectable per query |
| Simple Prompting: Basic prompt templates without security considerations | Uses XML-wrapped prompts (<user_query> tags) to prevent prompt injection attacks |
| Limited File Format Support: Supports only PDF or TXT | Supports multiple formats: PDF, TXT, CSV, XLSX, XLS, DOCX, MD |

KEY INNOVATIONS

- Hybrid Retrieval Pipeline: BM25 + Vector Search + Cross-Encoder Reranking (3-stage retrieval)
- Self-Critique Loop: Iterative answer refinement with dedicated critic model
- Dual-Mode Operation: Strict (zero hallucination) vs Hybrid (web-augmented) modes
- Multi-Agent Architecture: Specialized agents for chat, data analysis, process mapping, and critique
- Web Search Integration: DuckDuckGo search with LLM-powered query refinement
- Production-Ready UI: Complete Streamlit interface with session management and cost tracking
- Smart Query Contextualization: LLM rewrites follow-up questions.

System Workflow

MODEL ARCHITECTURE

- The system follows an Agentic Retrieval-Augmented Generation (RAG) architecture.
- Combines:
 - Parametric memory (LLM knowledge)
 - Non-parametric memory (document embeddings)
- Uses agent-based control for intelligent query handling.
- Designed to ensure:
 - High accuracy
 - Minimal hallucination
 - Scalability for large documents

1. DOCUMENT PROCESSING

- Formats: PDF, TXT, CSV, XLSX, XLS, DOCX, MD
- Chunking: 800 chars, 100 overlap (RecursiveCharacterTextSplitter)
- Embeddings: Google text-embedding-004 (primary) + sentence-transformers (fallback)
- Storage: ChromaDB persistent vector database

2. HYBRID RETRIEVAL

- BM25 Keyword Search: Exact term matching (rank_bm25)
- Vector Semantic Search: Cosine similarity on embeddings
- RRF Fusion: Combines rankings via $RRF_score = \sum 1/(60 + rank_i)$
- Cross-Encoder Reranking: ms-marco-MiniLM-L-6-v2 scores [query, doc] pairs

3. AGENTIC CONTROL & ANSWER GENERATION

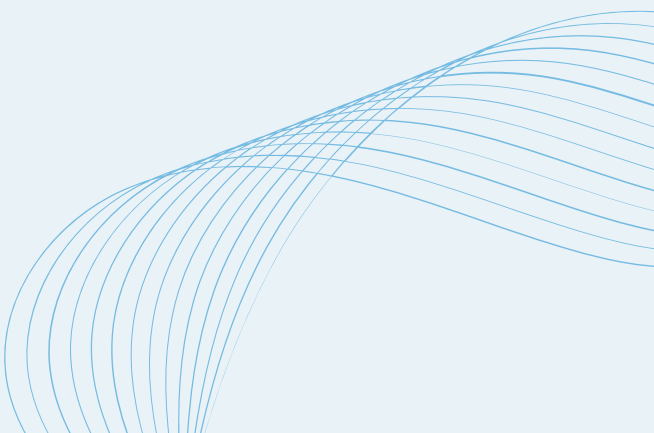
- Input Router Agent
 - Analyzes query intent
 - Routes to appropriate agent
 - Contextualizer
- Specialized Agents
 - Chat Agent - QA generation
 - Critic Agent - answer validation
 - Data Analyst - CSV/Excel analysis
 - Process Mapper - flowchart generation
- Self-Critique Loop
 - 0-3 iterations
 - Detects hallucinations and logical errors

4. MODE SELECTION, SECURITY & UI

- Dual-Mode Operation
 - Strict Mode: document-only, less hallucination
 - Hybrid Mode: documents + web search
- Security
 - XML-wrapped prompts prevent prompt injection
 - Session-isolated memory (UUID-based)
- User Interface
 - Streamlit chat UI
 - File upload, mode toggle
 - Streaming responses & cost tracking



1. WHY RAG OVER FINE TUNING?

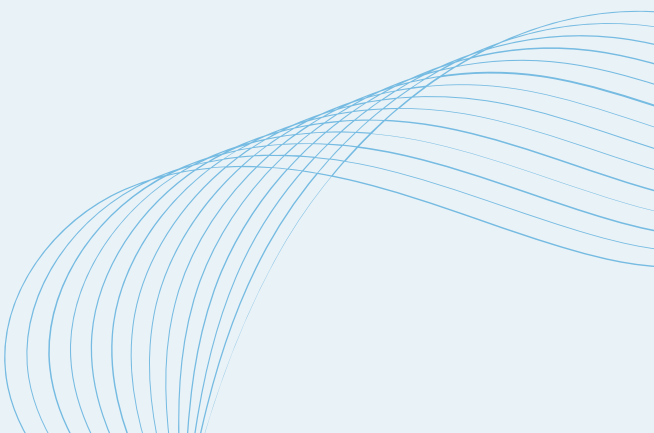
- Dynamic Adaptability: Updates instantly with new documents - no retraining required
 - Low Resource Requirements: No GPU training needed - runs on standard hardware
 - Minimal Hallucination: Grounded in actual documents + self-critique validation
 - Source Attribution: Can cite specific documents and page numbers
 - Easy Domain Transfer: Just upload new documents - works across any domain
 - Fast Implementation: Operational in hours, not days/weeks
 - Unlimited Scalability: Add unlimited documents without model capacity limits
- 

2. ADVANTAGES OF HYBRID RETRIEVAL

- Why 4-Stage Retrieval?
 - BM25 captures exact matches: Essential for technical terms, names, codes
 - Vector search captures semantics: Understands synonyms, paraphrases, concepts
 - RRF fusion optimizes ranking: Combines strengths of both methods
 - Cross-encoder reranking: Final precision layer, eliminates false positives
- Performance Benefits
 - Higher recall than single-method retrieval
 - Better precision through reranking
 - Robust across different query types

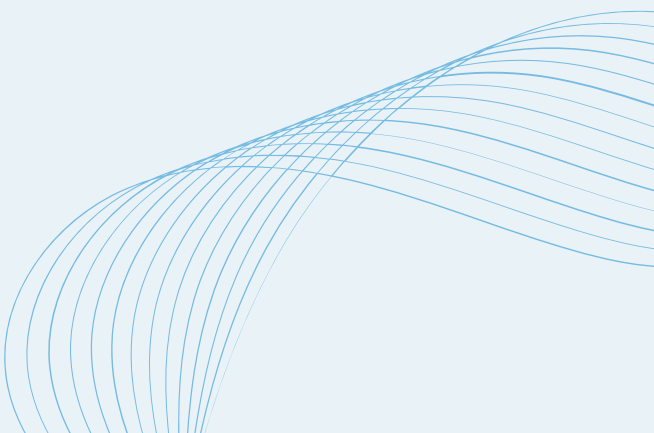


3. ADVANTAGES OF DUAL MODE ARCHITECTURE

- Strict Mode Use Cases
 - Legal document analysis (less hallucination required)
 - Compliance checking
 - Academic research (citation accuracy critical)
 - Hybrid Mode Use Cases
 - General research questions
 - Current events + historical context
 - Gap-filling when documents incomplete
 - Flexibility
 - User chooses mode per query based on needs
- 



4. ADVANTAGES OF SMART CONTEXT MANAGEMENT

- Problem
 - Fixed context windows waste tokens or miss information
 - Our Solution
 - Small files: Read entirely (comprehensive understanding)
 - Large files: Use RAG (efficiency)
 - Automatic detection and switching
 - Benefits
 - Optimal token usage
 - Better summaries for small documents
 - Scalability for large document collections
- 



DEMONSTRATION



THANK YOU