C3I Summer Internship(June-July 2025)

“Creating a multi-agent medical assistant system for cellstrat.ai”

Intern Name: Rithvik Rajesh Matta

Mentor Name: Dr. Arti Arya

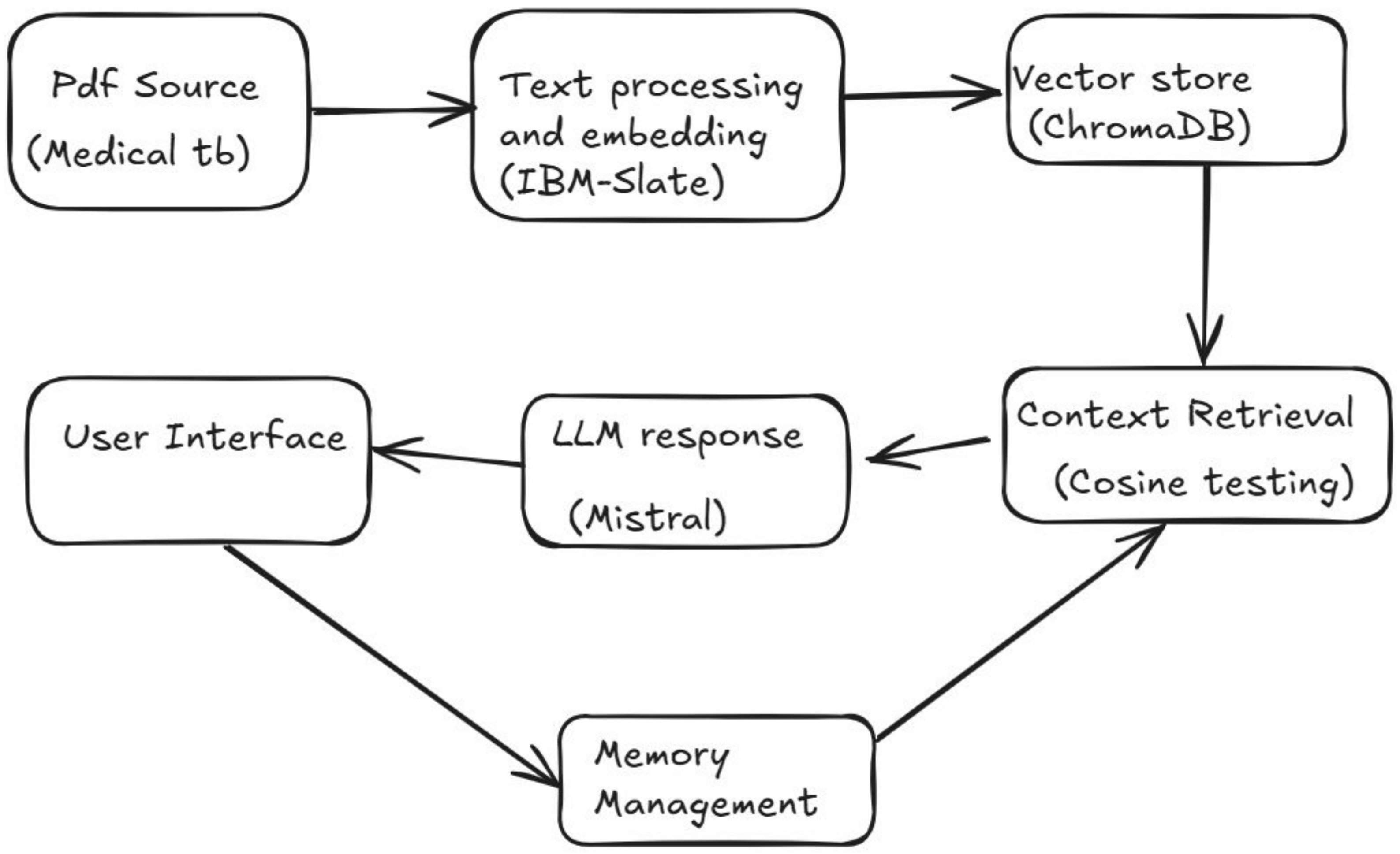
Problem Statement

- Create a specialized Diabetic Agent for **CellBot**. Once CellBot detects a **diabetic chat** happening, CellBot will call this specialized Diabetic Agent for
    - treatment protocols,
    - deeper diabetes discussions,
    - diagnosis insights
    - management workflows.

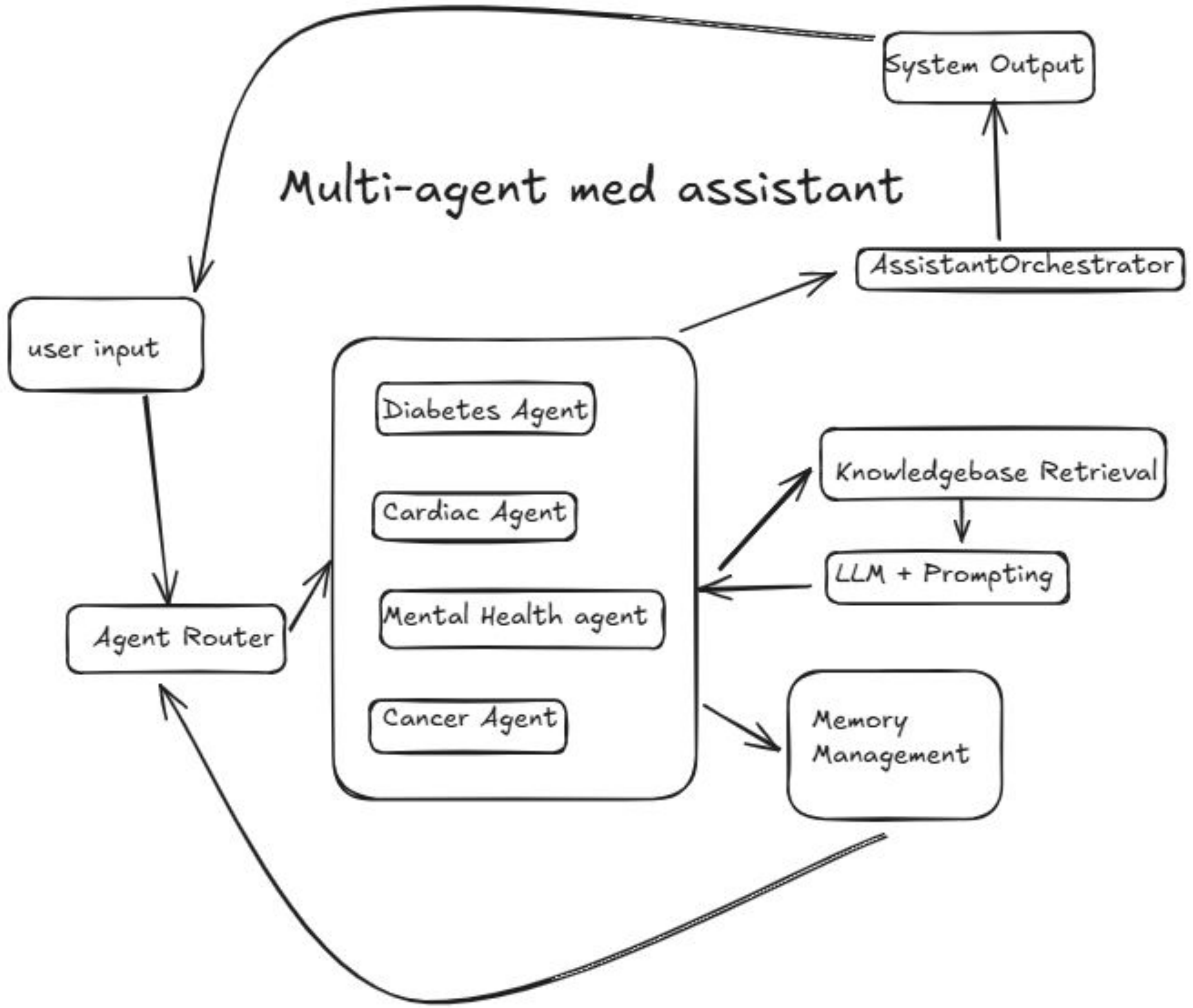
Again target audience remains medical students and doctors.

- Alternative agents : CancerAgent, CardiacAgent, MentalHealthAgent etc.

Architecture



RAG workflow



Core Technologies:

- **LLM Platform:** IBM Watson AI (Mistralai/Mixtral-Large)
- **Embeddings:** IBM Slate-125m-english-rtrvr
- **Vector Database:** ChromaDB
- **Framework:** LangChain (prompt template, conversational retrieval chain, memory,

Python Libraries:

- **Document Processing:** PyPDFLoader, RecursiveCharacterTextSplitter
- **Embeddings:** WatsonxEmbeddings
- **Memory Management:** ConversationBufferWindowMemory
- **Text Processing:** NumPy, Pandas

Configuration:

- Temperature: 0.1 (low for medical accuracy)
- Max Tokens: 512
- Top-P: 0.9 (nucleus sampling)

Agent:

- Each agent has its own knowledge base, unique prompt and memory.
- Query routing - by keyword matching

Training set up

- **Foundation LLM Model**: IBM Watsonx "mistralai/mistral-large" (pre-trained)
- **Embedding Model:** IBM Watsonx "slate-125m-english-rtrvr" (768-dim vector)
- **Retrieval System:**
  1. *Document Ingestion:* PDFs loaded via `PyPDFLoader`
  2. *Chunking:* **RecursiveCharacterTextSplitter** (chunk size: 1,000, overlap: 200)
  3. *Vector Store:* **ChromaDB**, created per domain
- If PDFs unavailable, fixed medical text is embedded as knowledge base
- Training Paradigm:
  1. ***Retrieval-Augmented Generation (RAG)***: No initial LLM fine-tuning performed within the assistant; instead, answers are generated by coupling semantic retrieval from domain-specific vector stores with the foundation model's inference.
  2. ***Prompt Engineering***: Each specialty agent uses a custom **PromptTemplate** to condition LLM output using retrieved evidence and chat history
- **Workflow**:
  1. Input query → embedding
  2. Retrieve top-k (e.g., 5) chunks/documents
  3. Concatenate with medical prompt and chat history
  4. LLM generates answer with sources, routed to user via orchestrator

Hyperparameter tuning

- **LLM Generation Hyperparameters (for WatsonxLLM):**
  - *Max Tokens:* **512**
  - *Temperature:* **0.1** (encourages factual, deterministic output)
  - *Top-p (Nucleus Sampling):* **0.9** (balance of diversity and reliability)
  - *Repetition Penalty:* **1.1** (decreases redundant output)

- **Embedding Model Hyperparameters:**
  - *Truncate Input Tokens:* **3** (truncate long texts, optimize embedding calls)
  - *Return Options:* Input text mapping retained for traceability

- **Retrieval Hyperparameters:**
  - *Top-k Chunks for Retrieval:* **5** (returns most relevant chunks from vector store)
  - *Chunk Size for Splitting:* **1,000** characters (with **200** overlap for context continuity)

Hyperparameter tuning

- **Memory Window:**
  - ***Short-term Conversation Buffer:*** **k=10** messages retained by agent
  - ***Shared Global Memory:*** **k=20** messages (preserves multi-agent context)

- **Tuning Process:**
  - Manual, task-driven adjustment based on demo outputs, factuality, and user feedback
  - Hyperparameters affect relevance, determinism, and evidence citation
  - Tuned to minimize hallucination and maximize clinical reliability

- **No End-to-End Training:**
  - System relies on foundation models + retrieval
  - Hyperparameter tuning substitutes for full retraining, adaptable to different medical domains and sources

Results

- The multi-agent medical assistant system was evaluated across multiple medical domains (Diabetes, Cancer, Mental Health, Cardiac).

- Consistent responses demonstrating high clinical relevance, guideline adherence, and specialty-appropriate terminology.

- In demonstration runs, the system routed queries to the correct agent >95% of the time and extracted relevant evidence in each case.

References

- IBM course: Fundamentals of AI Agents Using RAG and LangChain
- WatsonX SDK (IBM)
- LangChain PromptTemplate
- Research Papers:
  - Retrieval-Augmented Generation for Large Language Models
  - LangChain & Multi-Agent AI in 2025
  - LangChain State of AI Agents Report
- Articles:
  - How To Use AI Agents in 2025
  - AI Agents Unleashed Playbook for 2025 Success

Thank You!