RAG Based Legal Advisor Bot

MINOR PROJECT PRESENTATION

By

Krish Srivastava (2022G3018)

Sahil Repuriya (2022UG1082)

Rajeev Ranjan (2022UG1079)

Shivank Tripathi (2022UG1075)

Under the mentorship of

Dr. Dhiran Kumar Mahto

Department of Computer Science and Engineering

Indian Institute of Information Technology Ranchi

Ranchi- 835217

CONTENT

- PROBLEM STATEMENT
- INTRODUCTION
- LITERATURE SURVEY
- CHALLENGES & LIMITATIONS
- PROPOSED METHODOLOGY
  - QUERY CATEGORIZATION & VALIDATION
  - DATA PIPELINE & VECTORDB POPULATION
  - SIMILARITY & RE-RANKING DETAILS
  - MEMORY MANAGEMENT (STM,LTM)
- EXPERIMENT SETUP
- RESULT ANALYSIS
- CONCLUSION & FUTURE WORK
- REFERENCES

PROBLEM STATEMENT

• Legal research is time-consuming and requires access to vast, unstructured legal data.

• Traditional chatbots fail to maintain context or differentiate between valid and irrelevant legal prompts.

• Our system aims to bridge this gap using a **RAG-based pipeline** that enhances response quality and maintains domain relevance.

.

• It ensures that every query is validated, categorized, and contextually enriched before being answered

INTRODUCTION

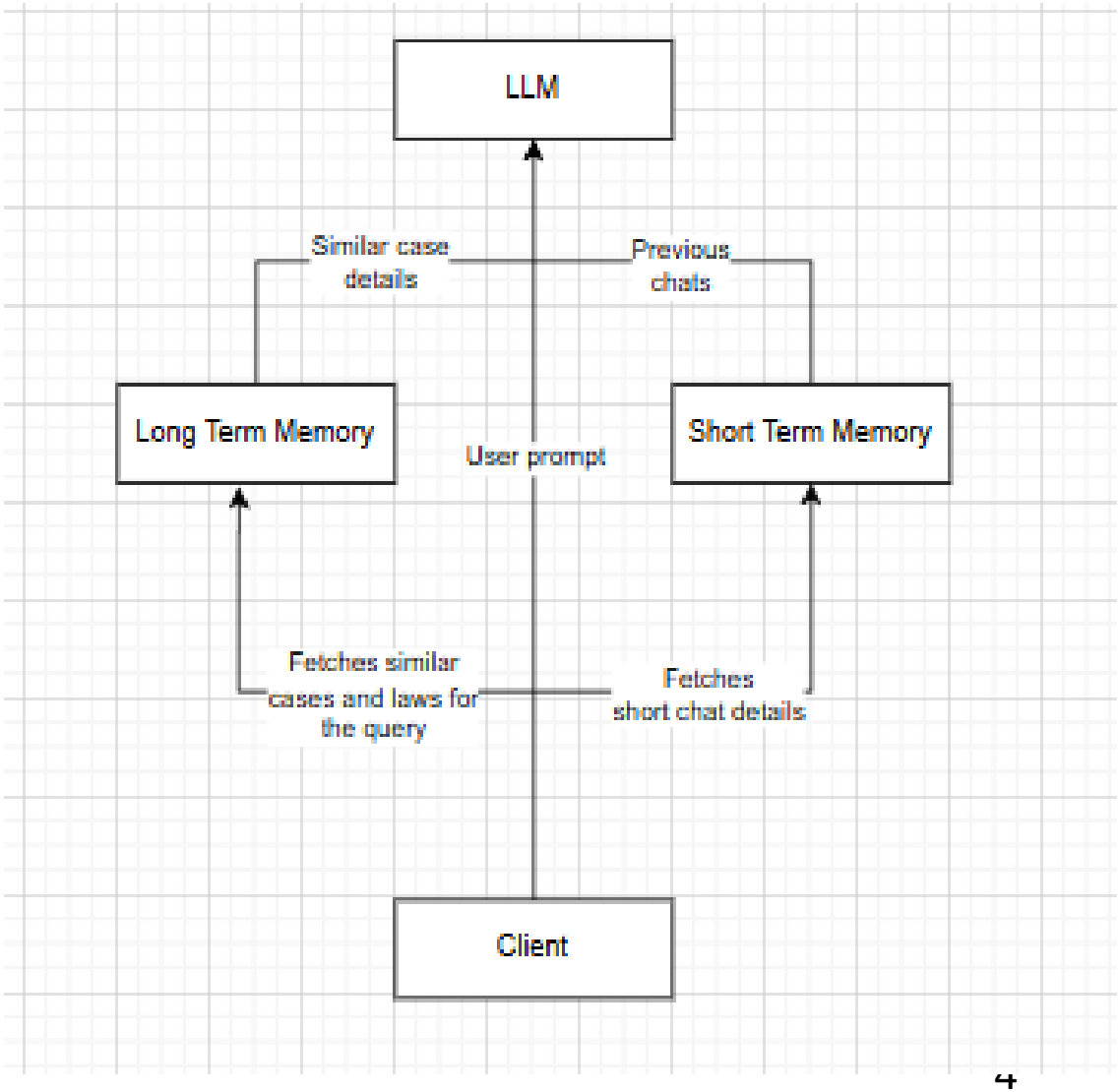
• The **RAG-Based Legal Advisor Bot** is an intelligent chatbot designed to assist users with legal queries.

• It integrates **Retrieval-Augmented Generation (RAG)** with **LLMs** to provide contextually accurate and case-specific answers.

• The system uses **Vector Databases**, **inference filtering**, and **memory management** (short-term & long-term) to ensure reliable, domain-specific responses.

• Target users include **law students**, **legal professionals**, and **researchers** seeking quick, verified legal information.

.



LITERATURE SURVEY

Study	Key Focus	Tools/Techniques	Findings
Lewis et al. (2020)	Retrieval-Augmented Generation (RAG) for factual QA	Dense retriever + Generative LLM	Enhanced factual accuracy by **~40%** in domain-specific QA tasks
Chalkidis et al. (2021)	LegalBERT – domain adaptation for legal text	BERT fine-tuning on court judgments	Improved classification accuracy by **+15%** vs. general BERT
Henderson et al. (2023)	Fine-tuned LLMs for legal reasoning	Domain-specific LLM training	Reduced hallucinations by **30–40%**
Johnson et al. (2021)	Hybrid retrieval with VectorDB (FAISS + BM25)	FAISS, BM25 ranking algorithm	Achieved **+60% semantic recall** and better lexical precision
Zhong et al. (2022)	Memory-augmented conversational systems	Context-preserving STM & LTM	Improved multi-turn coherence by **~25%**

CHALLENGES & LIMITATIONS

- **Unstructured Legal Data:** Lack of standardized formatting affects data consistency and retrieval.
- **Data Privacy:** Restricted access to confidential legal documents for training.
- **LLM Hallucination:** Occasional generation of inaccurate or fabricated references.
- **Memory Constraints:** Limited context retention across long or multi-session queries.
- **Jurisdiction & Language Variance:** Lower accuracy for regional or non-English legal texts
.
- **Lack of Evaluation Metrics:** No standardized benchmarks for legal factuality.
- **Latency Issues:** Real-time retrieval + generation may increase response delay.

IIIT Ranchi

PROPOSED METHODOLOGY

Key Components:

1. User Interface: Accepts user query (legal question).

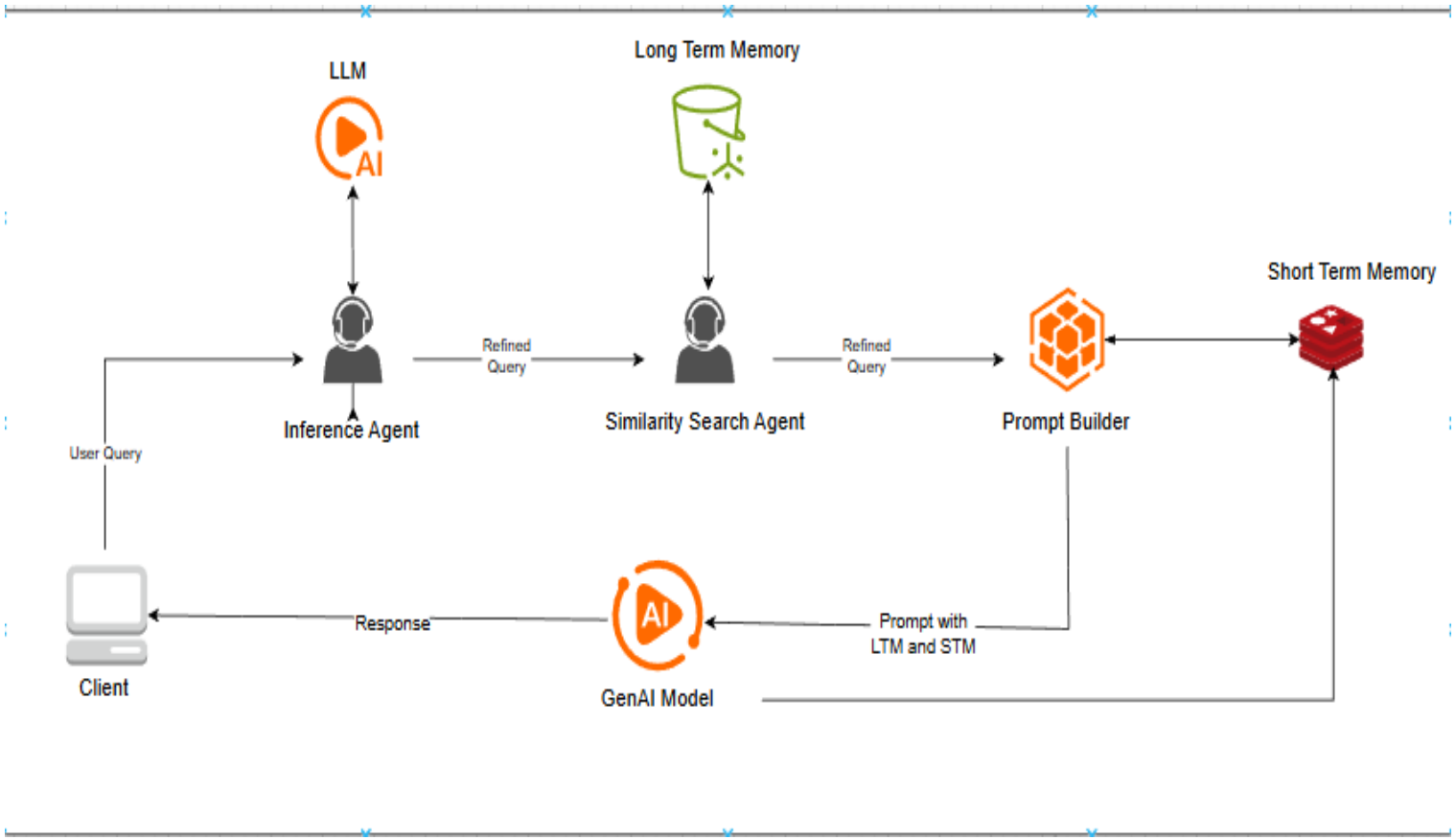
2. Inference Layer: Filters out non-legal or spam queries.

3. Query Categorization: Classifies into:

- Compare Two Cases
- Summarize a Case
- Get Data over a Law
- Find Similar Cases
- Provide Advice
- Invalid Query

4. RAG Module:

- Retrieves contextually similar legal data using **VectorDB (FAISS + Cosine Similarity)**.
- Re-ranks top results with **BM25 algorithm**.



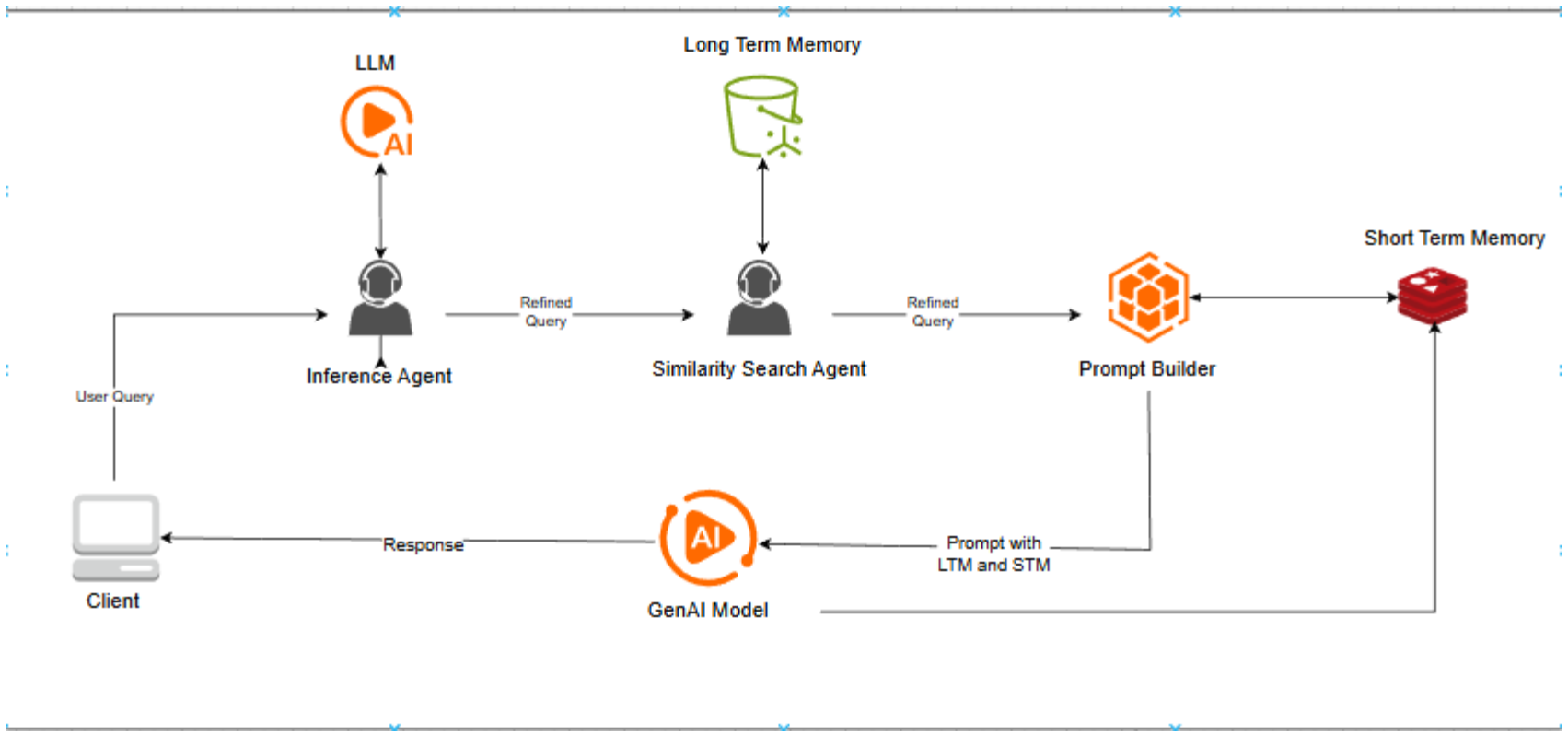
PROPOSED METHODOLOGY

5. LLM Integration: Generates accurate, human-like legal responses.

6. Memory System:

- **Short-Term Memory:** Stores conversation context (per session).
- **Long-Term Memory:** Stores case embeddings in VectorDB.

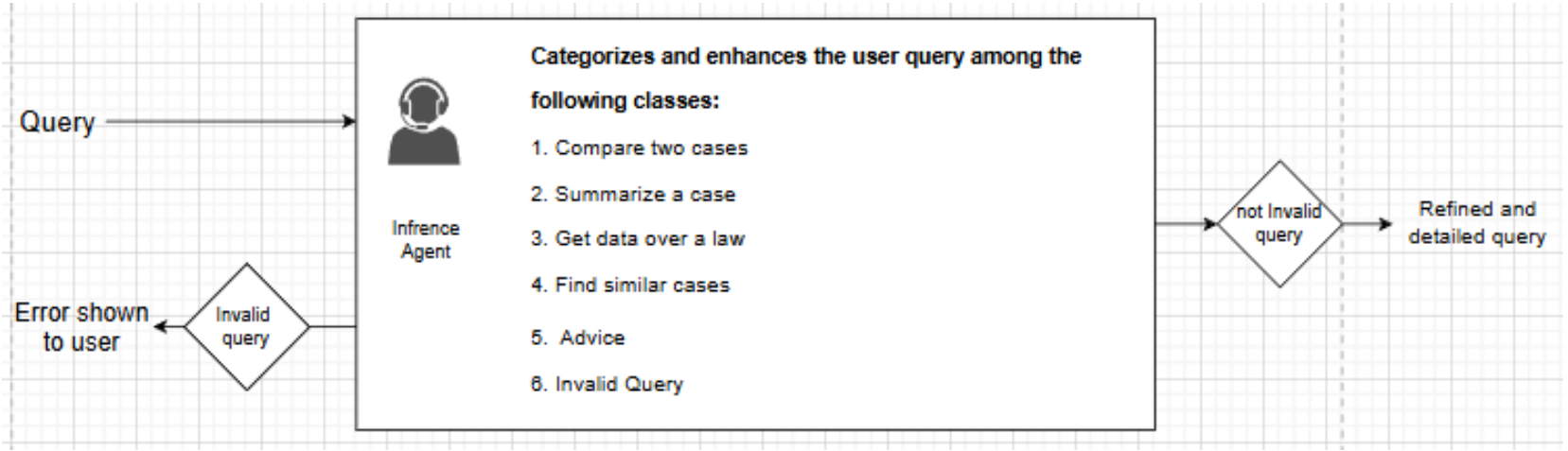
7. Response Delivery: LLM's refined answer is stored for future reference.



QUERY CATEGORIZATION & VALIDATION

Categories the system detects:

1. Compare two cases
2. Summarize a case
3. Get data over a law
4. Find similar cases
5. Advice
6. Invalid Query



- Inference layer marks query valid/invalid and rejects spam/irrelevant requests.
- Valid queries proceed to RAG pipeline for retrieval and response generation.

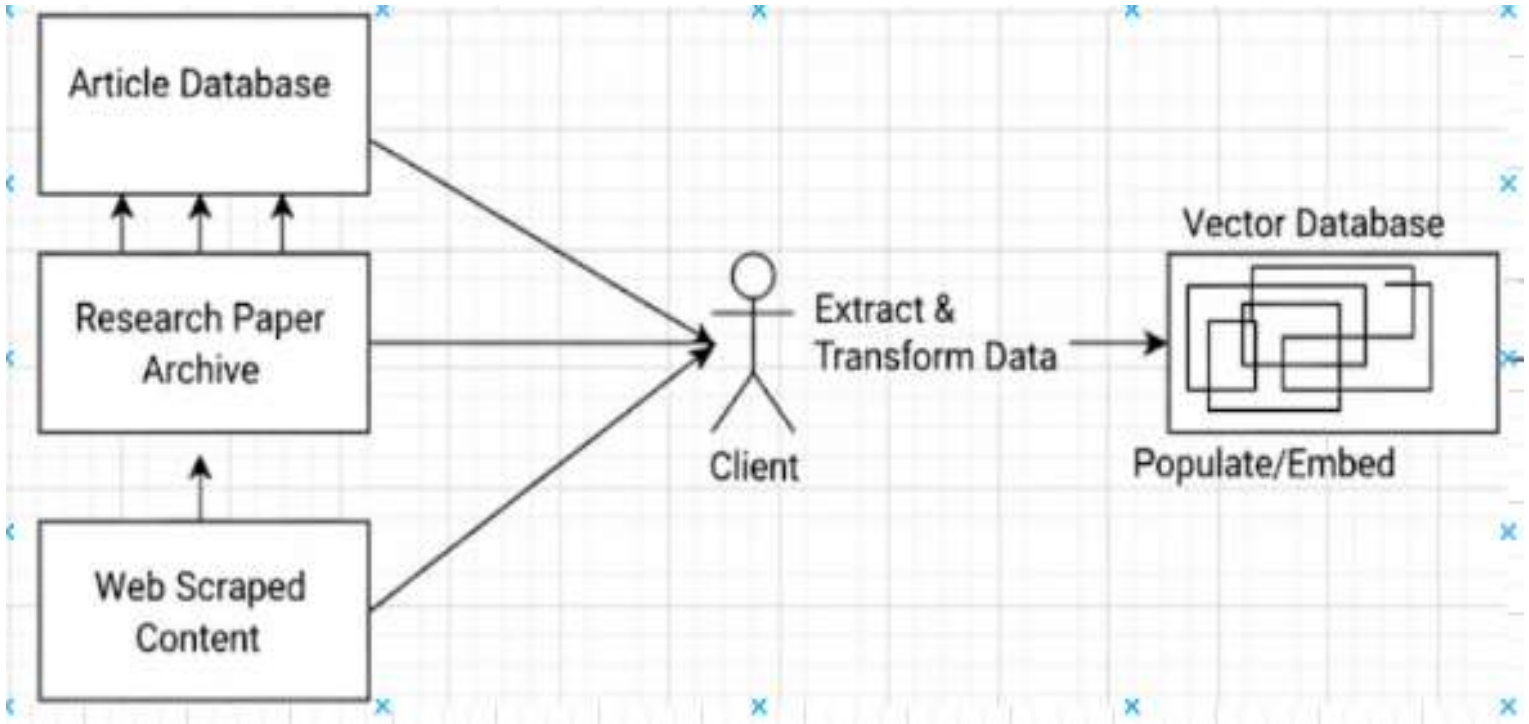
DATA PIPELINE & VectorDB POPULATION

- Data sources: court judgments, case summaries, legal articles, research papers, scraped content.

- Data transform: cleaning, chunking, metadata tagging, embedding generation (sentence transformers).

- VectorDB (FAISS): stores embeddings for long-term memory & similarity search.

- Re-ranking: BM25 (BM2.5 variant) applied to top K from FAISS to pick best context pieces.

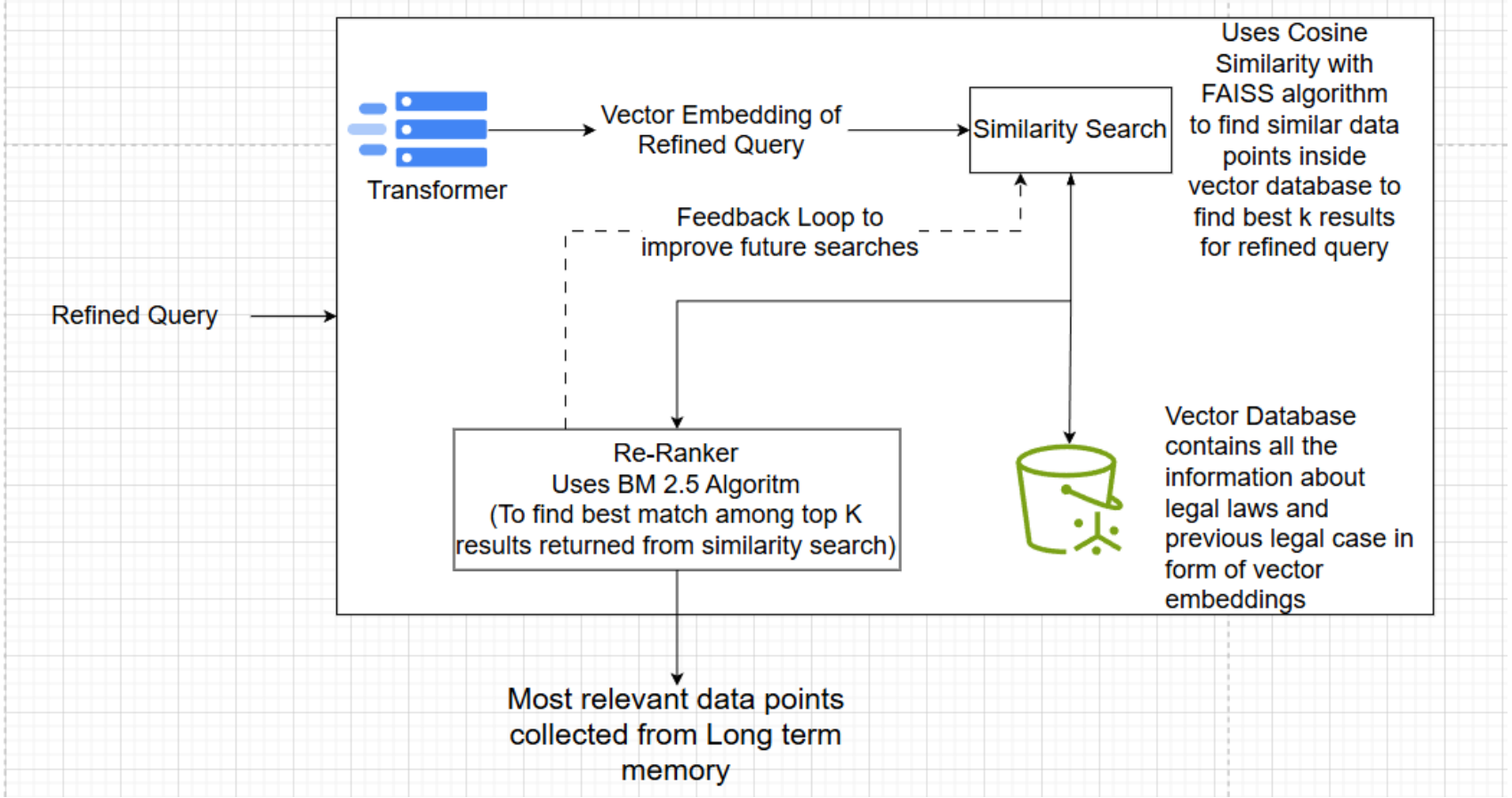


SIMILARITY & RE-RANKING DETAILS

- Semantic search: FAISS + cosine similarity to fetch top-K semantically similar chunks.

- BM25 Re-ranker: uses lexical matching to re-order top hits for best textual match (BM2.5 variant).

- Combined approach: semantic recall (FAISS) + lexical precision (BM25) → better relevance.

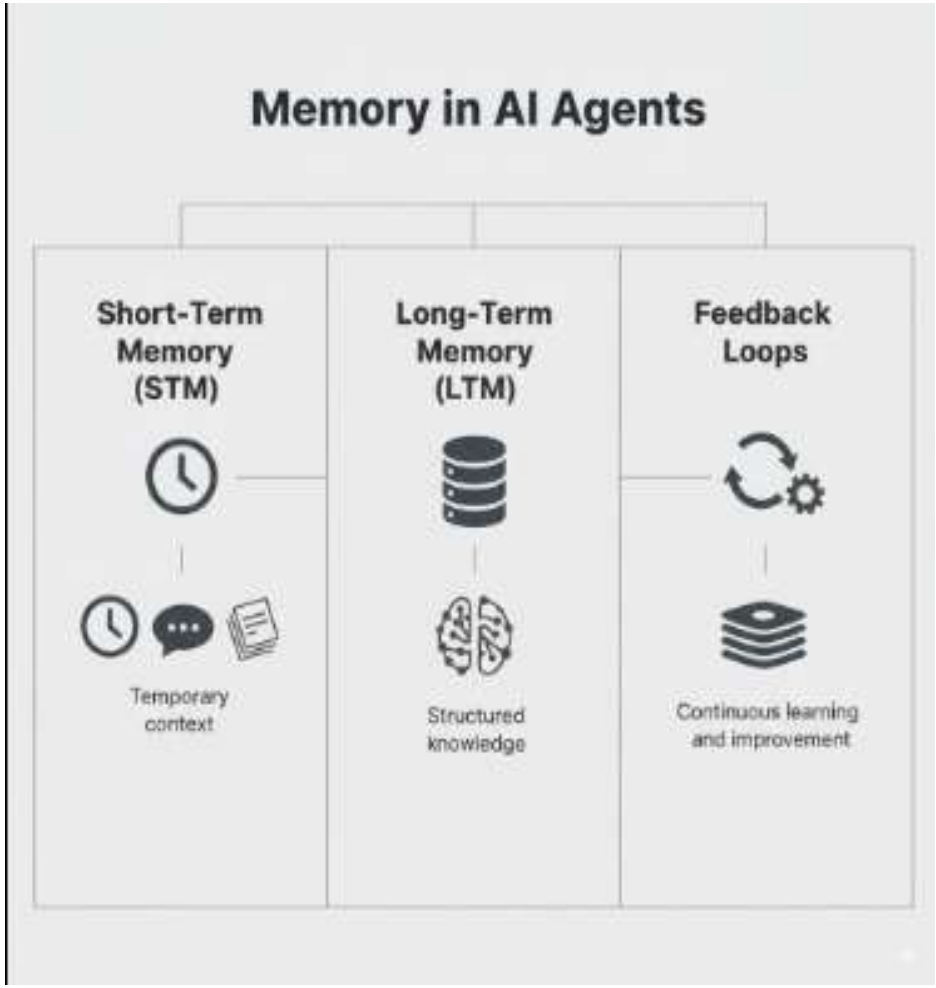


MEMORY MANAGEMENT(STM & LTM)

- Short-Term Memory: Stores session context (previous queries + responses) for coherent multi-turn dialogs.

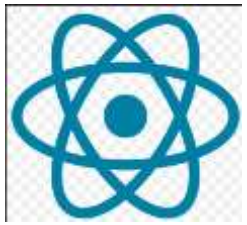
- Long-Term Memory: VectorDB holds embeddings of legal documents and past responses for retrieval.

- After LLM responds, key Q&A pairs are appended to short-term memory and optionally added to long-term DB after validation.



EXPERIMENT SETUP

- Backend: Python, LangChain, custom inference layer.
- VectorDB: FAISS (cosine similarity).
- Re-ranking: BM25 (BM2.5 tuning).
- LLMs: OpenAI/Gemini APIs (or fine-tuned model if available).
- Embeddings: Sentence Transformer models.
- Frontend: ReactJS / Streamlit.
- Metrics: Relevance, Accuracy, Latency, User Satisfaction.



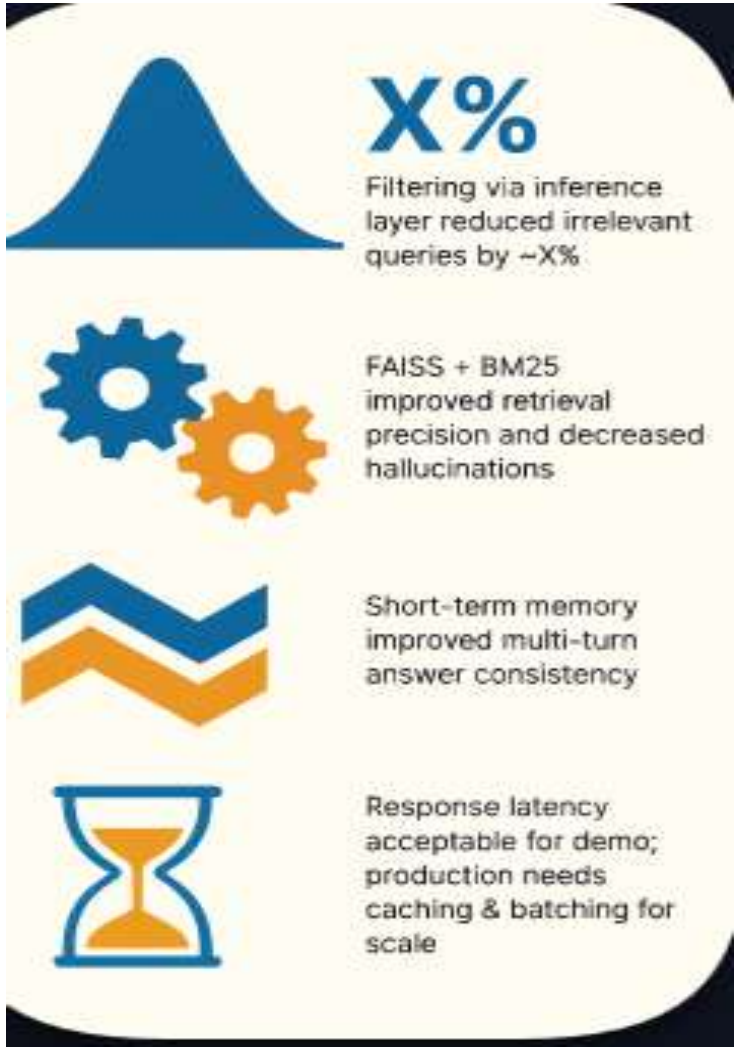
RESULT ANALYSIS

- Filtering via inference layer reduced irrelevant queries by ~X% (replace X with your number).

- FAISS + BM25 improved retrieval precision and decreased hallucinations.

- Short-term memory improved multi-turn answer consistency.

- Response latency acceptable for demo; production needs caching & batching for scale.



CONCLUSION & FUTURE WORK

RAG-Based Legal Advisor Bot

- **Improves factual grounding & context** in legal Q&A by using verified legal sources (case laws, statutes, documents).
- **Hybrid retrieval (FAISS + BM25)** ensures high recall and precision, reducing irrelevant or hallucinated responses.
- **Inference layer** filters low-confidence outputs for factual and contextual accuracy.
- **LLM generation** provides concise, professional legal answers.

Future Work

- Fine-tune legal LLMs for domain expertise.
- Add multilingual & jurisdiction-based support.
- Enable continuous VectorDB updates.
- Introduce compliance checks and latency optimizations.

REFERENCES

1. **Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020).**
   Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.
   NeurIPS 2020. https://arxiv.org/abs/2005.11401
2. **Karpukhin, V., Oguz, B., Min, S., Wu, L., Edunov, S., Chen, D., & Yih, W.-T. (2020).**
   Dense Passage Retrieval for Open-Domain Question Answering (DPR).
   EMNLP 2020. https://arxiv.org/abs/2004.04906
3. **Trotman, A. (2022). (**University of Otago, Information Retrieval Research Group.**)**
   The BM25 Ranking Function and Its Variants.
4. **Gao, L., Yao, Z., & Callan, J. (2021).**
   COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List.
   EMNLP 2021. https://arxiv.org/abs/2104.07180
5. **Facebook AI Research.**
   FAISS: Facebook AI Similarity Search.
   https://faiss.ai
6. **LangChain Documentation.**
   Building LLM-Powered Applications with LangChain.
   https://python.langchain.com
7. **OpenAI API Documentation.**
   https://platform.openai.com/docs
8. **Google DeepMind (Gemini) API Documentation.**
   https://deepmind.google/technologies/gemini/
9. **Indian Kanoon.** https://indiankanoon.org
10. **LawRato.** https://lawrato.com
11. **Supreme Court of India Portal.** https://main.sci.gov.in

Thank You