ChatBot - Conversational AI for PDF Manuals/Documents

Dissertation

Submitted in partial fulfillment of the requirements of the
M.Tech Data Science and Engineering Degree Program

By

Maj Abhishek Yadav
BITS ID No. 2022DA04453

Under the supervision of

Lt Col Chirag Chatterjee
(Joint Director)
Indian Army

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan)
INDIA

Jun 2024

ACKNOWLEGDEMENT

I would like to express my deepest gratitude to those who have been instrumental in the successful completion of this dissertation.

First and foremost, I am profoundly grateful to my supervisor, **Lt Col Chirag Chatterjee**, for his invaluable guidance, support, and mentorship throughout this journey. His insights and encouragement have been crucial in shaping this research, and I am deeply appreciative of the time and effort he invested in helping me succeed.

I also extend my heartfelt thanks to my mentor, **Mr. Vinaya Sathyanarayana**, whose wisdom and advice have been a constant source of inspiration. His thoughtful feedback and unwavering support were pivotal in overcoming challenges and refining my work.

A special thanks to my friend, **Mr. Abhishek Yadav**, for his constant encouragement, assistance, and companionship throughout this process. His support helped me stay focused and motivated.

Lastly, I am immensely grateful to the **Indian Army** as an organization, for not only shaping me as an individual but also providing the environment and resources that made this endeavour possible. The values instilled by the Army have been a guiding force in my academic and personal life.

Certificate from the Supervisor (Organizational)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Dissertation entitled **ChatBot - Conversational AI for PDF Manuals/Documents** and submitted by Major Abhishek Yadav ID No. 2022DA04453 in partial fulfillment of the requirements of DSECLZG628T / AIMLCZG628T Dissertation, embodies the work done by him/her under my supervision.

Signature of the Supervisor

Name : **Lt Col Chirag Chatterjee**
Designation: Joint Director

Date: 09 Sep 2024

ABSTRACT

The rise of AI and machine learning has greatly impacted military logistics. This project aims to create an advanced chatbot for PDF Manuals/Documents and Correspondence using Retrieval-Augmented Generation (RAG) techniques with Large Language Models (LLMs). The goal is to provide precise and efficient responses to queries, improving operational efficiency and decision-making within the Indian Army.

The chatbot will use RAG to deliver accurate answers from a centralized document store. Key steps include uploading documents with metadata including the sensitivity of the information and its version, generating embeddings with LLMs, storing these in a vector database, and implementing a retrieval mechanism to find relevant document segments. The LLM will generate responses based on user queries and retrieved segments, ensuring accuracy and relevance. The intuitive web application interface will have a role based access control and will include feedback mechanisms and regular updates to the document repository.

This AI-driven chatbot aims to enhance the accuracy and efficiency of accessing logistics information, improving decision-making and streamlining operations within the Indian Army.

Key Words:

RAG, Large Language Models, Supply Chain Management, Document Embeddings, Information Retrieval, Natural Language Processing.

List of Symbols & Abbreviation

1. Indian Army ………………………………………………………...IA
2. Natural language Processing ………………………………………. NLP
3. Large Language Models ……………………………………………LLM
4. Retrieval Augmented Generation …………………………………. RAG
5. Reserve Bank of India………………………………………………RBI
6. Standard Operating Procedure ……………………………………. SOP

List of Tables

List of Figures

Contents

1. Introduction

2. Methodology & Use Case

3. Objectives

4. Conclusion/Recommendations

5. Bibliography

Chapter 1

Introduction

i) Problem Definition

The logistics sector within the military is inundated with extensive manuals, instructions, and correspondence that are crucial for daily operations. However, the current systems for accessing and utilizing this vast repository of information are often inefficient and ineffective. Traditional methods of searching through documents manually or using basic search functions can be time-consuming and may not always yield accurate or contextually relevant results. This inefficiency can lead to delays in decision-making and operational disruptions within the IA. There is a pressing need for an advanced, reliable system that can quickly and accurately retrieve pertinent information from these documents to support better decision-making and enhance operational efficiency.

ii) Motivation

The motivation for this project stems from the critical role that logistics manuals and instructions play within the military. In the Army, these documents are followed to the letter and spirit, ensuring that operations are carried out precisely and consistently. However, the continuous updating and versioning of these documents add a layer of complexity to efficiently retrieving the necessary information. Traditional methods are not equipped to handle the dynamic nature of these documents, leading to significant inefficiencies.

The advancements in artificial intelligence and machine learning, particularly in the domain of NLP, present an opportunity to address these challenges. The emergence of RAG techniques, which combine the strengths of information retrieval and generative models, promises to revolutionize how information is accessed and utilized. By leveraging LLMs, we can create a system that understands and processes natural language queries, retrieving the most relevant information from a vast and continuously updated document repository. This project aims to bridge the gap between the current inefficiencies in military logistics information retrieval and the potential of cutting-edge AI technologies to provide accurate, contextually relevant answers swiftly.

iii) Aim

The primary aim of this project is to develop an advanced chatbot tailored specifically for handling PDF Manuals/Documents, Instructions, and Correspondence. The chatbot will utilize RAG techniques based on LLMs to deliver precise and efficient responses to user queries. The system will centralize various documents into a unified vector store, ensuring high-quality, contextually relevant answers. Key steps in achieving this aim include:

a) **Document Upload and Metadata Integration:** Uploading documents along with their relevant metadata into a centralized system.

b) **Embedding Generation:** Using LLMs to generate embeddings that represent the semantic content of document segments.

c) **Vector Database Storage**: Storing these embeddings in a centralized vector database to facilitate efficient similarity searches.

d) **Retrieval Mechanism Implementation:** Implementing a retrieval mechanism to obtain relevant document segments by searching the vector store for embeddings similar to the query embedding.

e) **Response Generation:** Using an LLM to generate responses based on both the user query and the retrieved document segments, ensuring accuracy and contextual relevance.

f) **Role Based Access Control:** Accessing of relevant information only as per the sensitivity of documents

g) **Continuous Improvement:** Establishing feedback mechanisms to enable ongoing improvements and regularly updating the document repository with new documents and amendments.

iv) Importance

The development of this AI-driven chatbot is crucial for several reasons:

a) **Enhanced Decision-Making:** By providing precise and contextually relevant answers, the chatbot will significantly improve the decision-making processes within the Army. Accurate information retrieval will enable all employees to make informed decisions quickly, thereby enhancing overall operational efficiency.

b) **Streamlined Operations:** The ability to swiftly access pertinent logistics information will streamline various operational processes, reducing delays and increasing productivity. This is particularly important in high-stakes environments where time and accuracy are critical.

c) **Technological Advancement:** Integrating advanced AI techniques like RAG with LLMs in a military context showcases the potential of AI to revolutionize various sectors. This project not only addresses immediate logistical challenges but also sets a precedent for future technological innovations within the military.

d) **User-Friendly Interface:** The intuitive web application interface ensures that users, regardless of their technical expertise, can easily interact with the system and obtain the information they need. This accessibility is essential for widespread adoption and effective utilization of the chatbot.

e) **Continuous Improvement:** The feedback mechanisms and regular updates to the document repository ensure that the system remains relevant and effective over time. This adaptability is crucial for maintaining high standards of accuracy and relevance in a rapidly changing environment.

In summary, the development of this AI-driven chatbot utilizing RAG for the Army promises to transform the way logistics information is accessed and utilized, leading to improved decision-making, streamlined operations, and significant technological advancements.

Chapter 2

Methodology & Use Case

i) RAG Model

It combines the strengths of retrieval-based and generation-based methods to improve the accuracy and reliability of generative AI models. This is achieved by supplementing generative processes with information from external sources, such as a company's catalog documentation.

Retrieval-based methods utilize a large corpus of text to find and extract relevant snippets for creating responses. These methods are excellent at providing factual and accurate information quickly, though they may sometimes lack creativity and fluency.

On the other hand, generation-based methods, or large language models (LLMs), generate text directly. While they can produce creative and fluent responses, they may also introduce inaccuracies and inconsistencies.

RAG integrates these two approaches by first retrieving relevant text snippets and then using an LLM to generate a new response informed by the retrieved information. This combination enhances the accuracy, reliability, and fluency of generative AI models.

The RAG workflow consists of three main steps:

a) **Retrieval**: identifying and retrieving relevant text snippets from external knowledge sources using techniques such as keyword-based search, semantic similarity search, or neural network-based retrieval.

b) **Augmentation**: Using the retrieved snippets to augment the generation process by providing additional context, improving response accuracy, or enhancing fluency.

c) **Generation**: Generating a new response using the augmented information through techniques like template-based generation, statistical language models, or neural network-based generation.

RAG offers several advantages over traditional approaches:

a) Reduced Risk of Hallucinations: By grounding responses in external knowledge, RAG mitigates the risk of generative AI models producing text that is not based on reality.

b) Improved Factual Accuracy: Ensures responses are based on reliable information retrieved from external sources.

c) Enhanced Fluency: Improves the fluency of responses by accessing a broader range of language structures and expressions found in the retrieved snippets.

d) Reduced Reliance on Retraining: Decreases the need for frequent retraining of generative AI models, as it can incorporate the latest information from external sources without complete model updates.

e) Enhanced Question Answering: Improves the accuracy of question-answering systems by providing access to relevant knowledge bases and documents.

f) Improved Summarization: Enhances the fluency and accuracy of summarization systems by utilizing summaries of other relevant documents and sources.

A flow diagram of the main components of the RAG architecture :-

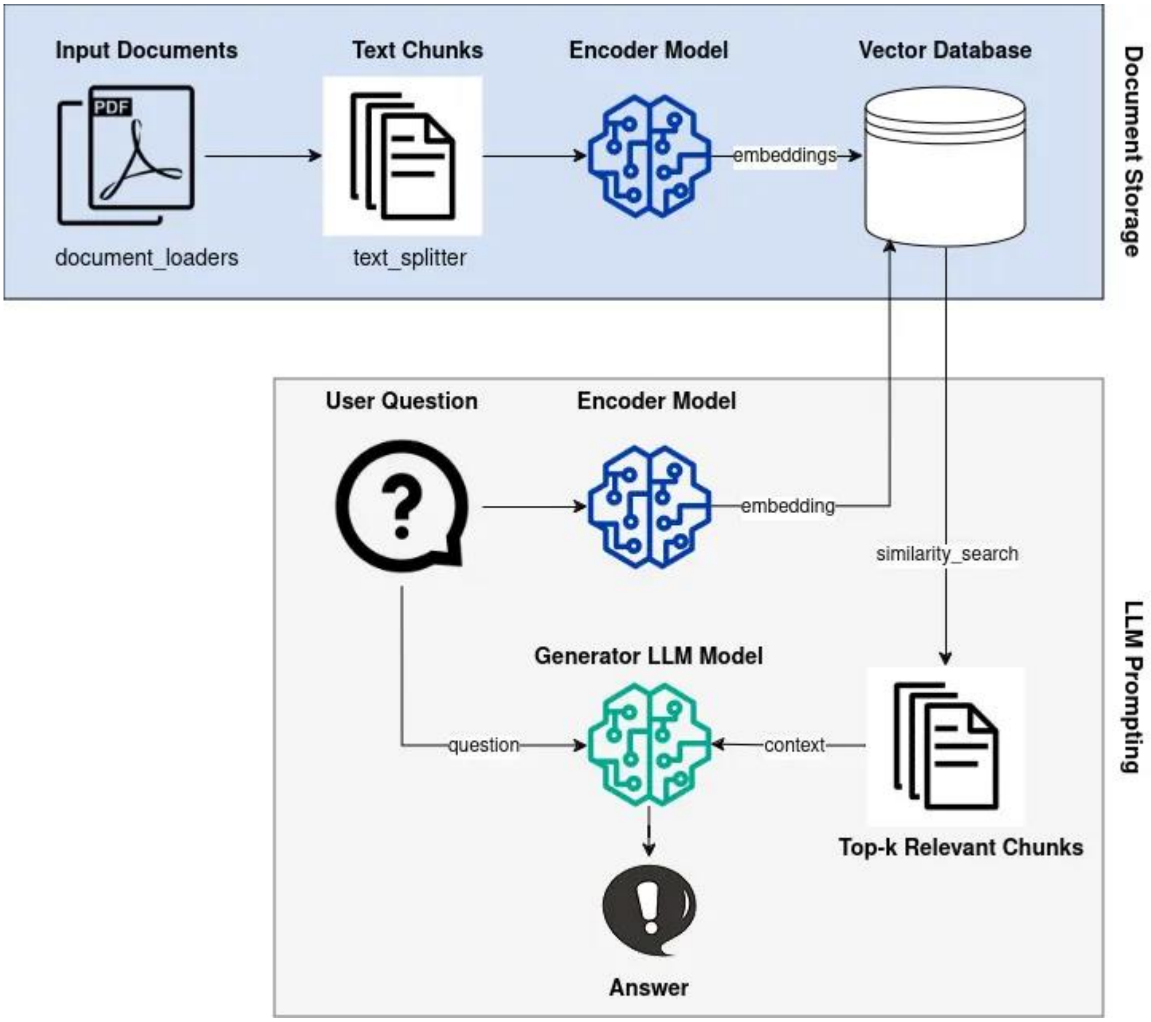
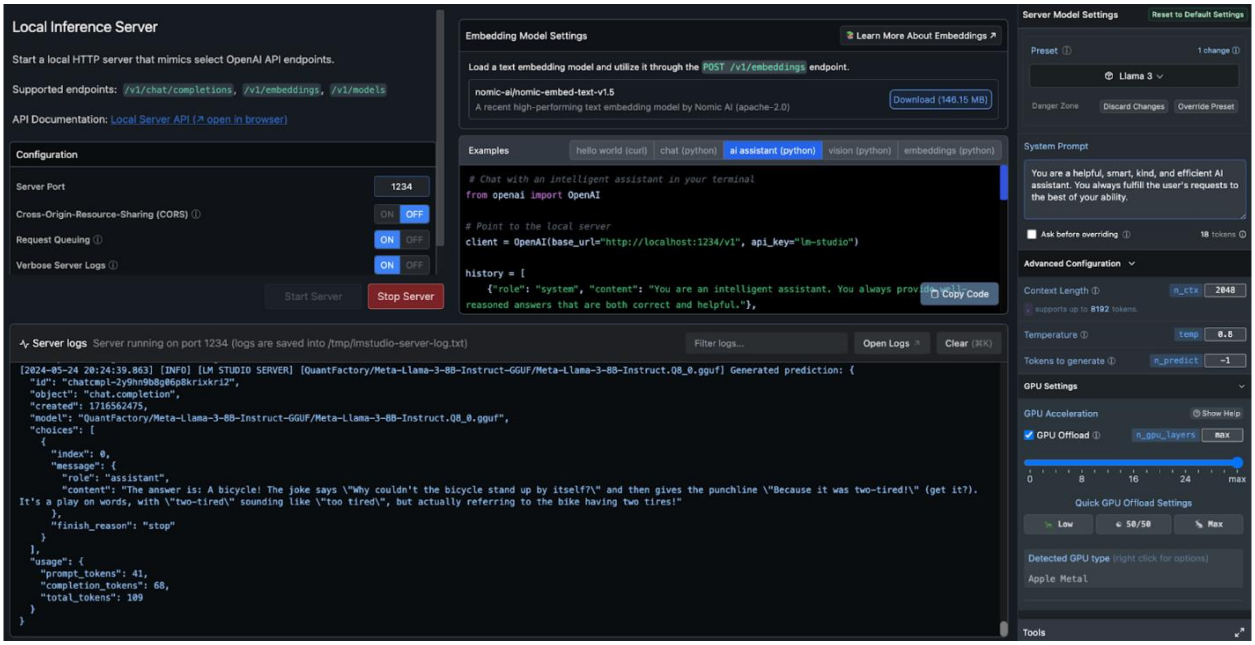


Fig 1 : RAG Based Chabot Architecture

ii) Running Local LLMs

Recent advancements in software and hardware have made it possible to run large language models (LLMs) on personal computers, a necessity for sensitive information within the Indian Army. LM Studio is a powerful tool that simplifies this process, offering a user-friendly interface for managing models, datasets, and configurations. Setting up LM Studio involves ensuring your computer meets specific system requirements, downloading and installing the software, and configuring it to run LLMs efficiently. Key features of LM Studio include the ability to run models offline, ensuring data privacy, and an interactive console for real-time engagement with the LLM, making it an invaluable resource for secure and effective deployment within the Army.

Running an LLM locally with LM Studio offers numerous benefits such as enhanced data privacy, cost savings by utilizing existing hardware, and the ability to customize model settings to match specific requirements. This is particularly critical for the Indian Army, where logistics manuals and instructions are frequently updated and must be followed meticulously. Offline capabilities ensure that sensitive information remains secure and accessible even without internet connectivity. Despite challenges like high hardware demands and setup complexity, the advantages of local LLM deployment make it a practical and strategic choice for military applications.



Local Inference Server

Start a local HTTP server that mimics select OpenAI API endpoints.

Supported endpoints: `/v1/chat/completions`, `/v1/embeddings`, `/v1/models`

API Documentation: Local Server API (↗ open in browser)

Configuration

Server Port: 1234

Cross-Origin-Resource-Sharing (CORS) ⓘ: ON [✓ OFF]

Request Queuing ⓘ: [✓ ON] OFF

Verbose Server Logs ⓘ: [✓ ON] OFF

Embedding Model Settings

Load a text embedding model and utilize it through the `POST /v1/embeddings` endpoint.

nomic-ai/nomic-embed-text-v1.5
A recent high-performing text embedding model by Nomic AI (apache-2.0)

Examples: hello world (curl) | chat (python) | [✓ ai assistant (python)] | vision (python) | embeddings (python)

```
# Chat with an intelligent assistant in your terminal
from openai import OpenAI

# Point to the local server
client = OpenAI(base_url="http://localhost:1234/v1", api_key="lm-studio")

history = [
    {"role": "system", "content": "You are an intelligent assistant. You always provide well-
reasoned answers that are both correct and helpful."},
```

Server logs Server running on port 1234 (logs are saved into /tmp/lmstudio-server-log.txt)

```
[2024-05-24 20:24:39.863] [INFO] [LM STUDIO SERVER] [QuantFactory/Meta-Llama-3-8B-Instruct-GGUF/Meta-Llama-3-8B-Instruct.Q8_0.gguf] Generated prediction: {
  "id": "chatcmpl-2y9hn9b8g06p8krixkri2",
  "object": "chat.completion",
  "created": 1716562475,
  "model": "QuantFactory/Meta-Llama-3-8B-Instruct-GGUF/Meta-Llama-3-8B-Instruct.Q8_0.gguf",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "The answer is: A bicycle! The joke says \"Why couldn't the bicycle stand up by itself?\" and then gives the punchline \"Because it was two-tired!\" (get it?).
It's a play on words, with \"two-tired\" sounding like \"too tired\", but actually referring to the bike having two tires!"
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 41,
    "completion_tokens": 68,
    "total_tokens": 109
  }
}
```

Server Model Settings

Preset ⓘ: Llama 3 (1 change ⓘ)

System Prompt: You are a helpful, smart, kind, and efficient AI assistant. You always fulfill the user's requests to the best of your ability.

☐ Ask before overriding ⓘ — 18 tokens ⓘ

Advanced Configuration

Context Length ⓘ: n_ctx 2048 — supports up to 8192 tokens.

Temperature ⓘ: temp 0.8

Tokens to generate ⓘ: n_predict -1

GPU Settings

GPU Acceleration

☑ GPU Offload ⓘ: n_gpu_layers max

Detected GPU type (right click for options): Apple Metal

Fig 2 : LLM GGUF file deployed locally for connecting to API endpoint

Security Considerations for running LLMs Locally with LM Studio

When deploying large language models (LLMs) locally, especially within sensitive environments like the Indian Army, robust security measures are essential to ensure the confidentiality, integrity, and availability of the data and systems. Below are detailed security measures being considered:

a) **Data Encryption** All data stored on the local machine, including model files and interaction logs, is encrypted using robust encryption standards such as AES-256. This prevents unauthorized access in case of physical theft or loss.

b) **Access Control** Roles and permissions to restrict access to different parts of the system based on the user's role within the organization. This minimizes the risk of unauthorized access to sensitive information.

c) **Logging and Monitoring** Enabled detailed logging of all system activities, including access attempts, configuration changes, and interactions with the LLM. This helps in auditing and identifying any suspicious activities.

Other important factors include Data Integrity and Backup, Physical Security and User Training and Awareness. By implementing these comprehensive security measures, the Indian Army can ensure that their deployment of LLMs using LM Studio remains secure, protecting sensitive information from unauthorized access and potential security breaches.

iii) Use case

The IA manuals, for which the chatbot is being developed, contain highly confidential information that is critical to the functioning of the Army. Dissemination of these manuals in a public forum is strictly prohibited. This use case is not unique to the IA but is also applicable to other government agencies that manage sensitive information.
For instance, the RBI publishes various instructions related to its SOPs and numerous schemes, which are analogous to the manuals issued by the Army. Although the context and content may differ, the underlying principle of handling and managing critical, confidential information remains the same.

Kisan Credit Card Scheme

To illustrate the functionality and significance of the chatbot, data related to the Kisan Credit Card Scheme, launched by the Government of India (GoI) and implemented by the RBI, has been utilized as test data. The RBI has issued a total of four circulars concerning the Kisan Credit Card Scheme. These circulars serve as an excellent proxy to understand the requirements and challenges of managing and retrieving information from a continuously updated set of documents, similar to the IA technical manuals.

A total of four circulars have been promulgated by the RBI regarding the Kisan credit card Scheme. They are as below

a) **Circular 1:** Initial launch and guidelines for the Kisan Credit Card Scheme.

kisan-02-02.pdf

b) **Circular 2:** Amendments to the initial guidelines, introducing new features.

kisan-05-12.pdf

c) **Circular 3:** Further modifications and enhancements to the scheme based on feedback and policy changes.

kisan-08-12.pdf

d) **Circular 4:** Additional amendments to streamline the implementation process.

kisan-10-16.pdf

These various instructions were published on different dates and contain information and amendment to the previous instructions. This is how IA also updates its technical manuals.

Chapter 3

Objectives

The custom chatbot for the Army was designed with three primary objectives to ensure secure and efficient management and retrieval of sensitive information. These objectives include deploying a local large language model (LLM), implementing role-based access control (RBAC), and enabling custom ranking of documents based on their newness.

i) Local LLM Deployment

The first objective was to deploy a local LLM to ensure that sensitive data remains within the secure environment of the IA. This was achieved using LM Studio and a locally deployed Streamlit application. The steps involved were:

a) Model Selection: Ability to download any model which provides required suitability for handling the required tasks.

b) Local Endpoint Creation: The LLM was deployed on a local machine using LM Studio, creating a local endpoint that processes data within the secure infrastructure of the IA, thus preventing data from being transmitted over the internet.

c) User Interface: A Streamlit application was developed to provide an intuitive interface for interacting with the chatbot. This application facilitates efficient query handling and response generation while ensuring data privacy.

The implementation of these objectives is done using the different model which is deployed using LM Studio whereas Streamlit application serves as the user interface, allowing interaction with the LLM the retriever is handled by LangChain. It has been implemented as follows

1. Define constants for the local OpenAI API: Model name: "sentence-transformers/all-MiniLM-L12-v2", Local API URL and key for OpenAI: OPENAI_API_BASE = "http://localhost:1234/v1", OPENAI_API_KEY = "1234".
2. Initialize the OpenAI LLM with the local API using the OpenAI() function:
3. Pass the local API URL and API key as arguments.

4. Create a conversational retrieval chain using ConversationalRetrievalChain.from_llm() by passing the LLM and the configured retriever as inputs.

ii) Custom Reranking

Chatbot should enable custom ranking of documents, prioritizing newer information over older, superseded documents. This ensures that users receive the most current and relevant information in response to their queries. The steps involved were:

a) Metadata Management: Each document was tagged with metadata, including its publication date and any subsequent amendments or updates.
b) Embedding and Storage: Documents were embedded using the LLM and stored in a vector database, along with their metadata.
c) Ranking Algorithm: A custom ranking algorithm was developed to prioritize newer documents. When documents with similar information are retrieved, the system prioritizes the latest versions, ensuring users receive the most up-to-date information.
d) Query Handling: Upon receiving a query, the system retrieves relevant document segments, ranks them based on their newness, and generates responses that reflect the most current information.

The implementation of these objectives is done by processing documents and tagging with metadata. Furthermore, documents will be prioritized based on their date and the retriever needs to be re-configured to prioritize newer documents and ensure relevant responses.

iii) Role-Based Access Control (RBAC)

The last defined objective is role-based access control. Role-Based Access Control (RBAC) is essential for secure and efficient document management in the Streamlit app. It ensures users can only access documents based on their role, department, and document sensitivity. RBAC prevents unauthorized access while streamlining access to relevant information. The system filters documents based on user roles and metadata (sensitivity, category, date). Users only see documents they are authorized to access, ensuring both security and efficient document retrieval. Recent documents are prioritized, providing users with up-to-date information. It will be implemented through a clear definition of user roles according to their departments and their associated access sensitivity levels. Each user is assigned a role mapped to specific access permissions, determining which sensitivity levels and categories of documents they can access. Roles are defined by department (e.g., HR, Finance, Procurement). For example, an HR user may have access to "Public" and "Internal" documents but not "Confidential" ones. The matrix includes:

a) Defining Sensitivity Levels: Documents were categorized into four sensitivity levels: **"Public," "Internal," "Confidential," and "Highly Confidential."**

b) Defining Category : A document can be categorized into multiple categories enabling composite access of a single documents : **Finance, HR, Procurement & Operations**

c) Sample authorization matrix:

Username	**Department / Categories**	**Sensitivity Level**	**Access Rights**
Alice	Finance	Confidential	Finance docs (Public, Internal, Confidential)
			Access to budget docs (Public, Internal, Confidential)
Bob	HR	Internal	HR docs (Public, Internal)
			No access to Confidential docs
Charlie	Procurement	Public	Procurement docs (Public)
			No access to Internal or Confidential docs

Table 1 : Authorisation Matrix

These objectives and their implementation ensure the chatbot effectively enhances the Army's capability to manage and retrieve sensitive information securely and efficiently, providing a robust solution adaptable for similar use cases in other government agencies.

Chapter 4

Conclusion/Recommendations

i) Conclusion

This project aimed to address the inefficiencies in accessing military logistics manuals and instructions by developing an AI-driven chatbot tailored for the Indian Army. Through the integration of Retrieval-Augmented Generation (RAG) techniques and Large Language Models (LLMs), the system enables fast, accurate, and contextually relevant responses to user queries. By leveraging a centralized vector database, the chatbot provides advanced retrieval mechanisms, ensuring that the most current and relevant information is delivered. Moreover, the implementation of Role-Based Access Control (RBAC) ensures that sensitive information is protected, while the deployment of local LLMs guarantees data privacy and security. This solution not only improves decision-making processes and operational efficiency but also showcases the potential of AI in transforming document management and retrieval systems in high-stakes environments. With continuous feedback mechanisms and regular updates, the chatbot is well-positioned to meet the evolving needs of the Indian Army and other similar use cases in government organizations.

ii) Recommendations

To enhance the accuracy and functionality of the ChatBot we will add more advanced document parsing method that supports diverse formats (e.g., Word, Excel) using libraries like Apache POI or Pandas. Incorporate a Named Entity Recognition (NER) module using SpaCy or HuggingFace Transformers to better understand and prioritize document contents. To further enhance the chatbot few long-term functionalities can include, integrate voice command support with multi-language capabilities, enable contextual memory for personalized responses, and provide offline functionality with document caching. Add automatic summarization, advanced QA with multi-hop reasoning, and system integration for real-time data access. Strengthen security with multi-factor authentication, enable collaboration features like real-time document sharing, and offer data visualization through interactive dashboards. Implement machine learning for document routing, user analytics for optimization, and human-in-the-loop escalation for complex queries. Ensure an adaptive, accessible UI, and include gamification and training modules for user engagement. Additionally, incorporate disaster recovery with automatic backups for reliable uptime.

Bibliography

[1] Build an LLM RAG Chatbot with LangChain - *https://www.vlinkinfocom/blog/build-an-llm-rag-chatbot/*

[2] Integrating Local LLM Frameworks: A Deep Dive into LM Studio and AnythingLLM - https://pyimagesearch.com/2024/06/24/integrating-local-llm-frameworks-a-deep-dive-into-lm-studio-and-anythingllm/

[3] How to Run LLM Locally Using LM Studio? - https://www.analyticsvidhya.com/blog/2024/07/run-llm-locally-with-lm-studio/

[4] Data pertaining to documents - www.rbi.org.in

[5] How to Build a Local Open-Source LLM Chatbot With RAG - https://towardsdatascience.com/how-to-build-a-local-open-source-llm-chatbot-with-rag-f01f73e2a131

Check list of items for the Final report

a) Is the Cover page in proper format? Y / ~~N~~

b) Is the Title page in proper format? Y / ~~N~~

c) Is the Certificate from the Supervisor in proper format? Has it been signed? Y / ~~N~~

d) Is Abstract included in the Report? Is it properly written? Y / ~~N~~

e) Does the Table of Contents page include chapter page numbers? Y / ~~N~~

f) Does the Report contain a summary of the literature survey? Y / ~~N~~

i. Are the Pages numbered properly? Y / ~~N~~

ii. Are the Figures numbered properly? Y / ~~N~~

iii. Are the Tables numbered properly? Y / ~~N~~

iv. Are the Captions for the Figures and Tables proper? Y / ~~N~~

v. Are the Appendices numbered? Y / ~~N~~

g) Does the Report have Conclusion / Recommendations of the work? Y / ~~N~~

h) Are References/Bibliography given in the Report? Y / ~~N~~

i) Have the References been cited in the Report? Y / ~~N~~

j) Is the citation of References / Bibliography in proper format? Y / ~~N~~

Declaration: I certify that I have properly verified all the items in this checklist and ensure that the report is in proper format as specified in the course handout.

X

Abhishek Yadav
Major

Date: 09 Sep 2024

X

Lt Col Chirag Chaterjee
JDOS

Date: 09 Sep 2024

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
Work Integrated Learning Programmes Division
II SEMESTER 23-24

DSECLZG628T / AIMLCZG628T DISSERTATION

(Final Evaluation Sheet))

NAME OF THE STUDENT: ABHISHEK YADAV

ID NO. : 2022DA04453

EMAIL ADDRESS : yabhishek.470n@gov.in

NAME OF SUPERVISOR : Lt Col Chirag Chatterjee

PROJECT TITLE : ChatBot - Conversational AI for PDF Manuals/Documents

(Please put a tick (□) mark in the appropriate box)

S.No.	**Criteria**	**Excellent**	**Good**	**Fair**	**Poor**
1	Work Progress and Achievements				
2	Technical/Professional Competence				
3	Documentation and expression				
4	Initiative and originality				
5	Punctuality				
6	Reliability				
	Recommended Final Grade				

EVALUATION DETAILS

EC No.	**Component**	**Weightage**	**Marks Awarded**
1	Dissertation Outline	10%	
2	Mid-Sem Progress Seminar Viva Work Progress	 10% 5% 15%	
3	Final Seminar/Viva	20%	
4	Final Report	40%	
	Total out of	100%	

Note : Mark awarded should be in terms of % of weightage (consider 10% weightage as 10 marks)

	Organizational Mentor	
Name	Lt Col Chirag Chatterjee	
Qualification	B.Tech	
Designation & Address	Joint Director, Army HQ	
Email Address	chijee.401a@gov.in	
Signature		
Date	09 Sep 2024	

NB: Kindly ensure that recommended final grade is duly indicated in the above evaluation sheet.