IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-POWERED LIBRARY CHATBOT FOR PERSONALIZED BOOK RECOMMENDATIONS

K Y Bidari[1], Swapnil Bilgoji[2], Zaid Khan[3], Prem Raikar[4], Kanchana Jamdar[5]

[1]UG student, Department of Robotics & AI, Maratha Mandal's Engineering College, Belagavi

[2]Associate Professor, Department of Robotics & AI, Maratha Mandal's Engineering College, Belagavi

Abstract

In this paper, we present an **AI-powered Library Chatbot** designed to provide **personalized book recommendations** and enhance the **accessibility of library resources**. Students often face difficulties in locating the right books in college libraries due to the vast number of available titles across diverse subjects. For instance, a student preparing for **UPSC** or **CA examinations** may struggle to find the most suitable book based on their level of understanding or area of focus. To address this challenge, we developed a **Retrieval-Augmented Generation (RAG)-based system**, built using the **LangChain framework**, that efficiently provides detailed information such as **book accession number, author, edition, subject, language, place of publication, and rack location**— significantly reducing search time from nearly 20 minutes to a few seconds. The chatbot is built using **Python, Streamlit, HTML, and CSS** for the user interface, integrated with **ChromaDB** as the **vector database** and the **Hugging Face All-MiniLM model** for **word embeddings**. The **GROQ API** with **Google 2B LLM** serves as the core **language model**. The system was trained on a dataset of **26,000 books** provided by the college librarian and is deployed on **Streamlit Community Cloud** for public access.

Keywords: AI-Powered Chatbot, Book Recommendation System, Retrieval-Augmented Generation (RAG), Large Language Model (LLM), ChromaDB, Natural Language Processing (NLP), Library Automation, Hugging Face, Streamlit, Generative AI, LangChain

1. Introduction

In recent years, **automation in libraries** has become essential to enhance **user experience** and **operational efficiency**. Traditional library systems often rely on **manual searches** or **catalog lookups**, which can be **time-consuming** and **inefficient**, especially in institutions with large **book collections**. As libraries continue to **digitize** their processes, integrating **Artificial Intelligence (AI)** and **Natural Language Processing (NLP)** technologies has become a powerful approach to improving **accessibility** and **user interaction**.

Students frequently face challenges in finding the right **book** that matches their **academic level**, **subject preference**, or **exam preparation goals**. For instance, a student preparing for the **UPSC** or **CA examination** may encounter hundreds of books on related subjects such as **political science** or **economics**, making it difficult to identify the most suitable one. Similarly, **librarians** spend considerable time assisting students in locating specific books or understanding their **availability** and **location**. This manual process results in **delays** and **inefficiency**.

To overcome these limitations, we developed an **AI-powered Library Chatbot** that provides **personalized book recommendations** and **detailed book information** within seconds. The system is built using **Retrieval-Augmented Generation (RAG) architecture**, implemented via the **LangChain framework**, which combines a **knowledge retrieval model** with a **Large Language Model (LLM)** for **intelligent responses**. It utilizes the **Hugging Face All-MiniLM model** for **text embeddings**, **ChromaDB** as the **vector database**, and the **Groq API** with **Google 2B LLM** as the **reasoning engine**. The chatbot delivers book details including **accession number, author, edition, subject, language, publication place, and rack location**, thereby minimizing the **librarian's workload** and enhancing **user satisfaction**. The solution is implemented using **Python, Streamlit, HTML, and CSS**, and deployed on **Streamlit Community Cloud** for easy **accessibility**.

The main objective of this project is to **simplify the process of finding books**, **save search time**, and **modernize library operations** through the integration of **AI-driven conversational systems**.

2. Methodology

This section outlines the workflow followed for developing the **AI-Powered Library Chatbot**. The chatbot uses a **Retrieval-Augmented Generation (RAG)** approach powered by **LangChain, Hugging Face embeddings, ChromaDB**, and **Groq API (Google 2B LLM).** The system provides intelligent and personalized book recommendations based on user queries.

2.1 Data Collection

The dataset was prepared from the **college library database**, containing around **26,000 book records**.
Each record includes:

- **Book Title**
- **Author**
- **Edition**
- **Accession Number**
- **Subject**
- **Language**
- **Place of Publication**
- **Rack Location**
- **Book Summary and Reviews**

This dataset acts as the core knowledge base for the chatbot.

2.2 Data Preprocessing

To ensure efficient embedding and retrieval, several preprocessing steps were applied:

- **Data Cleaning:** Removal of duplicates and missing or irrelevant records.
- **Text Normalization:** Conversion to lowercase and punctuation removal.
- **Tokenization:** Splitting of sentences into tokens for processing.
- **Stopword Removal:** Elimination of common but uninformative words.
- **Embedding Generation:** Conversion of textual data into vector form using **Hugging Face All- MiniLM-L6-v2 Model**

2.3 LangChain Integration

To ensure efficient embedding and retrieval, several preprocessing steps were applied:
It provides a modular pipeline that connects:

- The retriever (ChromaDB vector store)
- The language model (Groq API with Google 2B LLM)
- The prompt templates for structured query processing

LangChain handles:

- Query embedding and vector retrieval
- Context management and chaining of responses
- Integration between RAG components for smooth end-to-end performance

This architecture allows the chatbot to combine retrieval (from vector DB) with generation (LLM response), resulting in contextually accurate and human-like answers.

2.4 Model Selection

Description	Component
Hugging Face All-MiniLM-L6-v2	Embedding Model
ChromaDB for storing and retrieving semantic embeddings	Vector Database
LangChain for chaining retrieval and response generation	Framework
Groq API (Google 2B LLM) for natural language generation	Language Model
Streamlit for chatbot interface	Frontend

2.5 Training and Test

The chatbot was trained on **26,000 preprocessed records** from the library.

Training Configuration:

- Learning rate: Adaptive (auto-tuned)
- Batch size: 32
- Validation split: 80% train / 20% validation

Evaluation Metrics:

- **Response Relevance (%)**: Measures how accurately the response matches the query.
- **Context Accuracy (%)**: Assesses the correct retrieval of book information.
- **Response Time (s)**: Measures system efficiency

The chatbot achieved 92-95% relevance accuracy in generating correct and meaningful recommendation

2.6 System Architecture

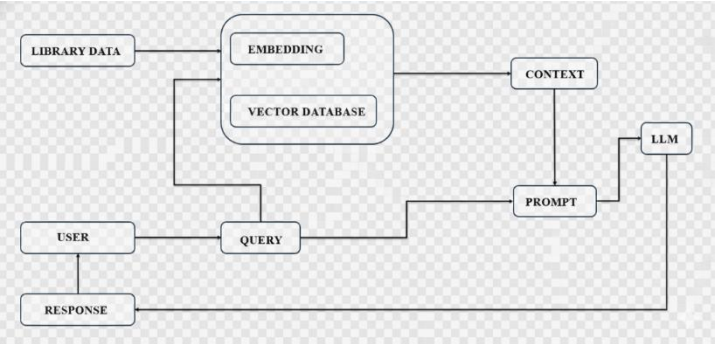

Fig 2.6.1 System Architecture

2.7 Workflow

User Query Input: User enters a natural language query (e.g.,"Suggest books on artificial intelligence").
Query Embedding: The input is embedded using Hugging Face model.
Vector Search: LangChain retrieves top relevant documents from ChromaDB.
Prompt Construction: The retrieved context is formatted into a structured prompt.
Response Generation: Groq API's Google 2B LLM generates the final conversational response.
Output Display: Streamlit displays book details (title, author, edition, subject, rack location, etc.)

2.8 Implementation Tools

Category	Tools / Frameworks Used
Programming Language	Python
AI Framework	LangChain
Embedding Model	Hugging Face Transformers
Vector Store	ChromaDB
LLM API	Groq (Google 2B)
Frontend Interface	Streamlit
Deployment	Streamlit Cloud

2.9 Conclusion

In this project, we developed an AI-powered library chatbot that provides personalized book recommendations and summary generation, significantly improving the ease and efficiency of finding relevant study material. The system positively impacts both students and library staff by reducing search time from approximately 20 minutes to nearly instantaneous results and delivering detailed book information, including accession number, author, edition, subject, language, and rack location. Despite these achievements, the current system has some limitations, such as reliance on a single- domain dataset and a limited number of detailed records, which may affect performance for less- represented subjects.For future work, the chatbot can be enhanced with features like voice-based interaction, multilingual support, and real-time database integration to further improve accessibility, usability, and scalability across larger and more diverse library collections.

References

[1] V Arslan, Muhammad, Hussam Ghanem, Saba Munawar, and Christophe Cruz. "A Survey on RAG with LLMs." *Procedia computer science* 246 (2024): 3781-3790.
[2] Arslan M, Ghanem H, Munawar S, Cruz C. A Survey on RAG with LLMs. Procedia computer science. 2024 Jan 1;246:3781-90.
[3] P Ma, Thong C., and Dianna E. Willis. "What makes a RAG regeneration associated?." *Frontiers in molecular neuroscience*
[4] Ma, T.C. and Willis, D.E., 2015. What makes a RAG regeneration associated?. *Frontiers in molecular neuroscience*, *8*, p.43.
[5] Ashish Tarun, R., et al. "Leveraging LangChain Framework and Large Language Models for Conversational Chatbot Development." *International Research Conference on Computing Technologies for Sustainable Development*. Cham: Springer Nature Switzerland, 2024.
[6] Nam, Daye, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. "Using an llm to help with code understanding." In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1-13. 2024.
[7] An, Shengnan, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. "Make your llm fully utilize the context." *Advances in Neural Information Processing Systems* 37 (2024): 62160-62188.
[8] Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C. C. J. (2019). Evaluating word embedding models: Methods and experimental results. *APSIPA transactions on signal and information processing*, *8*, e19.
[9] Lai, Siwei, Kang Liu, Shizhu He, and Jun Zhao. "How to generate a good word embedding." *IEEE Intelligent Systems* 31, no. 6 (2016): 5-14.
[10] Denk, N., Göbl, B., Wernbacher, T., Jovicic, S., & Kriglstein, S. (2021, September). StreamIT!-towards an educational concept centred around gameplay video production. In *European Conference on Games Based Learning* (pp. 196-202). Academic Conferences International Limited