IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DIALOGUE-BASED STORY LEARNING:

An AI powered Chatbot

[1]Dr. Reena Bharathi, [2]Aditya Wani, [3]Amar Sherkar, [4]Nikita Bhagat, [5]Nivedita Sonavane

[1]Associate Professor, [2]Student, [3]Student, [4]Student, [5]Student

[1]Department of Computer Science,

[1]Nowrosjee Wadia College, Pune, India

Abstract: Artificial Intelligence is a mechanism that makes machines think, learn, and solve problems, like humans do. This paper presents the development of a conversational AI Chatbot designed to make learning stories easier for a learner. By combining rule-based techniques with NLP and transformer-based models, the Chatbot offers interactive dialogue on the knowledge base that it is supported with. The system emphasizes both accuracy and engagement, drawing on simplified data to maintain authenticity.

User feedback indicates strong engagement, suggesting that conversational agents can enhance learning experiences. This paper demonstrates how AI can play a meaningful role in creating Chatbots for educational domain, through interactive, accessible technologies.

***Index Terms* – Chatbot, Knowledge graph, Neo4j, Vectorization, RAG, Graph RAG.**

I. INTRODUCTION

Artificial Intelligence (AI) continues to transform various fields, with education being no exception. Traditional educational methods often fail to captivate and engage students, resulting in a lack of deep understanding and appreciation of historical events and personalities. Recognizing this educational gap, we embarked on the design and development of this project, aiming to leverage AI to make learning about this important historical figure interactive and informative [4].

Our project utilizes advanced AI technologies to create an interactive chat experience that will provide a conversational learning platform to users. A Chatbot, in this context, is an AI-driven software application designed to mimic human-like conversations with users. It utilizes Natural Language Processing (NLP) techniques to understand and generate human language, allowing users to interact with the system in a natural, conversational manner. The Chatbot receives user input, processes it through AI models, and generates relevant responses based on the context and historical data it has been trained on. We employed various tools and techniques, including Retrieval-Augmented Generation (RAG)for providing accurate and relevant information [7] Llama (Large Language Model-based AI) for natural language processing [22], Generative AI like GroqAI for accelerated processing [21], Streamlit for building the user interface, and Neo4j for managing the knowledge graph [32]. The RAG technique, which combines information retrieved from a structured knowledge base into coherent, contextually accurate responses, enhances the Chabot's ability to deliver precise answers, making it particularly effective for specialized historical knowledge [2]. This AI-powered Chatbot serves as a virtual historian, providing users with an engaging and informative learning experience [15]. In addition to its core capabilities, our project also incorporates multilingual support to cater to a global audience [20]. Through interactive storytelling and intelligent conversation, AI can make learning more engaging, improve knowledge retention, and create an immersive experience [14].

The paper is structured as follows: Section 2 presents the background and related work, followed by Section 3, that concludes the paper by summarizing key insights, discussing limitations, and the future scope of AI-powered historical Chatbot in education.

II. BACKGROUND AND RELATED WORK

Artificial Intelligence (AI) plays a central role in the design and development of Chatbots. It uses natural language processing (NLP) to enables Chatbots to decode the user inputs written in natural language, to understand the user's intent, to capture the entities, and identify any grammatical errors, synonyms etc.

The impact of AI on automation in various domains like E-commerce, Art industry, Education etc has been tremendous. AI has enhanced efficiency in education through intelligent tutoring systems, automated grading, and personalized learning platforms [22][21][24]. These AI-driven systems analyse student performance, provide real-time feedback, and create adaptive learning pathways tailored to individual needs [17][16]. Our study focused on personalized learning systems [15], which adapt to students' progress using data-driven analytics, allowing them to learn at their own pace [14]. Hybrid learning models, powered with AI, provides flexibility and a structured approach to learning, thereby improving engagement and comprehension [26][18]. This in turn supports for a reduced workload for the educators, thereby allowing more focus on personalized instruction [23]. Our research emphasizes the importance of integrating technology thoughtfully; ensuring digital tools complement traditional methods rather than replace them.

A. Graph data modelling, using Neo4j.

Neo4j, a graph database, has emerged as a powerful tool for representing and managing knowledge [32]. Its ability to model relationships between entities makes it particularly suitable for applications requiring semantic understanding and contextual relevance [30]. Research highlights its use in enhancing generative AI Chatbot accuracy by integrating structured data from knowledge graphs. This approach improves the contextual relevance of Chatbot interactions, achieving high accuracy in data-relevant enquiries [5][1].
In the realm of Chatbots, Neo4j facilitates the incorporation of extensive knowledge bases, enhancing the semantic links between elements. This integration improves response quality, making Chatbots more effective in providing accurate and enriched information across domains like healthcare, education, and business [9]. Furthermore, Neo4j has been utilized in project-based learning environments, where it supports retrieval-augmented generation (RAG) techniques [6]. Lastly, Neo4j's synergy with large language models has revolutionized document analysis [30].

B. Retrieval Augmented Generation (RAG)

The integration of Retrieval-Augmented Generation (RAG) into Chatbot systems has revolutionized the way responses are generated, combining the strengths of information retrieval and generative AI [6]. RAG-based Chatbots dynamically retrieve relevant information from external knowledge sources, ensuring that responses are both accurate and contextually appropriate [5]. RAG based Chatbots, enables a real-time access to data, thus overcoming the static nature of pre-trained language models [9]. This capability significantly enhances the Chabot's ability to provide up-to-date and precise answers, particularly in domains requiring dynamic information, such as customer support and education. RAG, combined with knowledge graphs, further exemplifies their potential, by structuring data into semantic relationships, thereby generating a more effective and context-aware responses [6][13]. In addition to improving response quality, RAG systems also contribute to the scalability and adaptability of Chatbots. By leveraging modular architectures, RAG-based Chatbots can be tailored to specific domains, making them versatile tools for various applications, from business process automation to academic assistance [1].

C. Technological concepts Used in developing the chatbot

During the course of designing our conversational Chatbot, we were introduced to the following technologies that would provide a good quality and precision to our model.

Table 1: AI Technology

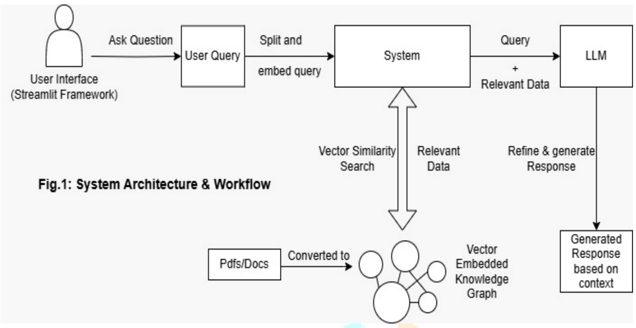
Sr. No.	Technology used	Purpose
1.	**Graph RAG**	A method that provides AI-generated responses by retrieving relevant knowledge from a structured graph database before generating an answer [7]
2.	**Embedding**	A numerical representation of words, sentences, or documents in a high-dimensional vector space, helps AI to understand relationships between entities.
3.	**A Chatbot**	An AI-powered conversational agent designed to interact with users through text or speech, providing responses based on previously provided information knowledge or real-time data retrieval.

4.	**Google AI API** (Gemini 1.5)	An AI service by Google used for natural language understanding, information retrieval, and contextual response generation [28].
5.	**Groq AI API** (Llama 3.2)	A language model API offering free services, used in this project for response generation due to cost-effectiveness and performance [6]
6.	**Streamlit**	Framework which get used for the front end designing of application as it provides a good visualization to application [29]

D. Architectural Model

Referring to figure 1, following is a brief description of its Components:

1. Ask Question: The user interacts with the Chatbot by asking a question in a specific language, such as English or any other supported language. The Chatbot interprets the input and forwards it for deeper semantic analysis.



2. Vector Similarity Search: A technique used to find the most relevant information by comparing the question to stored knowledge using mathematical vectors

3. Knowledge Graph: It structures and connects related pieces of information, and works alongside vector similarity search to enrich the search results. This helps in transforming raw matched patterns into meaningful and context-aware information that can be used to generate accurate answers.

4. Question + Relevant Information: The retrieved data, augmented with the question is fed to the LLM.

Fig.1: System Architecture & Workflow

5. LLM: The LLM is the large language model which works for the critical intelligence layer that transforms raw information into meaningful answers. It processes both the original question and retrieved context through neural network architectures that understand language semantics, syntax, and pragmatics, and generates the answer, which is the accurate response to the user's original question.

E. Implementation of AI powered Chatbot for Historical stories:

In this section we present the implementation of our AI powered Chatbot that provides a conversation-based approach to learning, based on its knowledge base. Following are the steps that were followed as part of the implementation of the Chatbot.

Step 1: Setting Up the Environment

To begin, we need a development environment where we can install and configure the necessary libraries and tools. This includes setting up our Neo4j instance and importing essential Python libraries for data processing, embedding generation, and language model usage. Generation of a Groq API key to access the model and a well- defined prompt to guide the kind of response we want from the Chatbot.

Step 2: Data Ingestion and Indexing

This step deals with the creation of a strong knowledge base for a robust information retrieval. We have used the Langchain library of Python, for loading the documents, into our knowledge base.

Step 3: Populating the Knowledge Graph

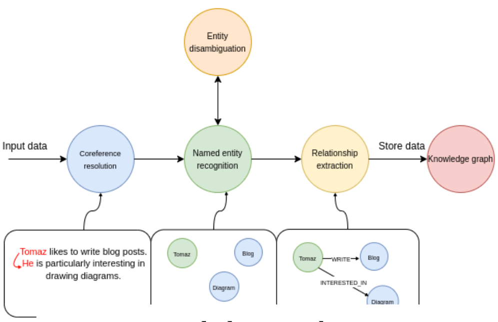


Fig.2: Knowledge Graph Process

We represent our knowledge base as a graph model. Figure 2, explains the process of knowledge graph creation, to represent our knowledge base.

Step 4: Storing the Embeddings in Neo4j Knowledge Graph

Instead of just using a traditional vector store, we have integrated the vector embeddings into the knowledge graph, as properties of nodes, while relationships between them (based on semantic similarity, context, or other criteria) can also be represented. Once the embeddings are indexed in the Neo4j database, tokenizing is completed the knowledge graph is successfully created.

Step 5: Creating Interactive Widgets

Interactive widgets are created using Streamlit open-source framework, thereby providing an user-friendly interface for conversing with the Chatbot. We also used the Groq AI based Groq API key to provide a fast and efficient response.

F. Challenges faced in developing our interactive Chatbot.

In the evolving landscape of artificial intelligence, choosing the right API for specific use cases is crucial for optimizing performance and achieving the best results. During the course of design and implementation of this Chatbot we explored four prominent AI APIs from OpenAI, Microsoft, Google, and Groq, each with distinct strengths and weaknesses. These APIs—OpenAI API (ChatGPT-4), Microsoft Azure OpenAI, Google Gemini 1.5, and Groq AI API (Llama 3.2) - each serve unique functions in the AI ecosystem, making them suitable for different industries and applications [3].

Retrieval-Augmented Generation became the best choice of our methodological innovation. This dual-approach methodology allowed with flexibility in data analysis, breaking down traditional barriers in computational research [13][10]. The technological evolution was started by working on critical transitions, starting with initial attempts using Gemini API, as the Gemini API model have a qualities like technical superiority, multimodal Capabilities [28], reasoning skills, multilingual Proficiency which help us to consider about Gemini API. After a trial with Gemini API, we understood its significant limitations in knowledge graph creation and data interpretation. Due to the inability of the Gemini API to meet our requirements for graph creation and data interpretation; we have transitioned to the OpenAI model [30].

Our path then led us through the OpenAI API, where cost constraints became apparent, ultimately finding resolution with the Groq API integrated with the Llama 3.2 Model. This combination forms a powerful solution, making it highly adaptable for processing both raw and historical data with exceptional accuracy. The Llama 3.2 Model [3] proved to be a game-changer, showcasing exceptional performance across varied data input scenarios.

Our RAG [23] model is able to adapt to various data sources and maintain high accuracy while providing multi-domain scalable insights. The underlying technology support Groq API's processing, Llama 3.2's language comprehension [13], and a proprietary RAG scheme for data retrieval and generation filled this gap.

Table 2, provides a summary of our journey, in selecting the best AI API for our interactive Chatbot.

Table 2: AI API Comparison

Feature	**OpenAI API (ChatGPT-4)**	**Google AI API (Gemini 1.5)**	**Microsoft AI API (Azure OpenAI)**	**Groq AI API (Llama 3.2)**
Token Limit	Supports 128K tokens, allowing long-context reasoning and structured responses.	Offers 1M tokens, the highest among all, making it ideal for long-document processing and complex AI tasks.	Supports 128K tokens, similar to OpenAI GPT-4, suitable for enterprise applications.	Supports 8K-16K tokens, lower than others, but optimized for speed and efficiency.
Multimodal Support	Yes, supports text, image, video, and audio, making it highly versatile.	Yes, supports text, image, video, and audio, making it highly versatile.	Yes, supports text, image, video, and audio, making it highly versatile.	Text and Image.
Knowledge Graph	Not integrated, requires additional implementation for structured historical data retrieval.	Integrated, capable of analyzing large text and images to create structured historical links.	Integrated, optimized for enterprise-grade structured data retrieval.	Integrated, enables fast and structured fact retrieval in Chatbot applications.
Embedding	Not integrated, necessary for improving semantic search and contextual response generation.	Not integrated, could enhance contextual search in multimodal data.	Not integrated, but can be leveraged for high-performance vector search in large datasets.	Integrated, enhances search accuracy and contextual matching,

				response generation is working.
Why Used	Selected for high reasoning ability and structured responses.	Chosen for its advanced multimodal capabilities and long-context support.	Preferred for enterprise security, scalability, and integration with Microsoft cloud services.	Used because of free services, making it cost-effective for real-time Chatbot needs.

G. A glimpse of our Conversational Chatbot

The objective of our research project was to build an intelligent, interactive Chatbot, for providing a dialogue-based learning on stories in History. We tested our Chatbot, to provide a conversational based learning on the history of Chhatrapati Shivaji Maharaj. This Chatbot enables users to have deep and meaningful conversations about the life, achievements, and legacy of Chhatrapati Shivaji Maharaj. The knowledge base is based on the book "Shivaji Maharaj's Diary" of Rajmata Prakashan [27]. Some of the screenshots are given below.

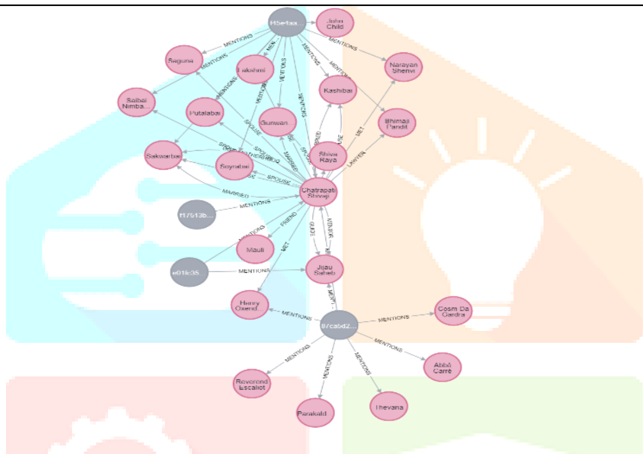


Fig.3: Knowledge Graph, with vector embeddings

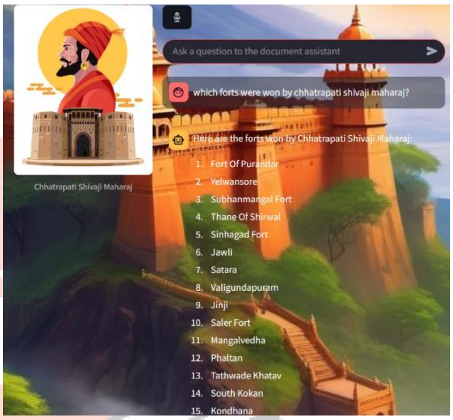


Fig.4: Question and Responses

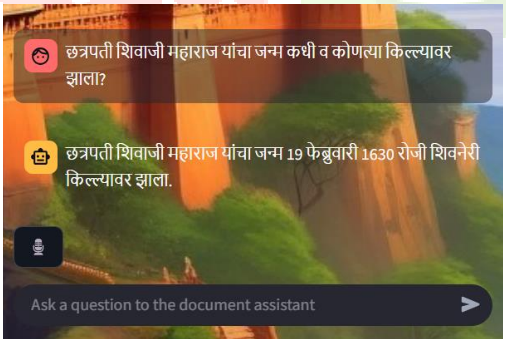


Fig.5: Multi Lingual Support

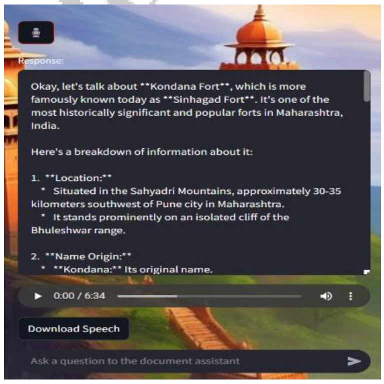


Fig.6: Voice Chat

III. Conclusion and Future Scope.

Through our project, "Dialogue-Based story Learning: An AI powered Chatbot" we set out to show how Artificial Intelligence can make education more interactive, accessible, and meaningful. By using advanced technologies like Retrieval-Augmented Generation (RAG), the Llama 3.2 language model, GroqAI API for faster processing, and Neo4j for managing our knowledge graph, our interactive Chatbot provides a realistic platform for conversation with users, and generate deep, context-rich answers to their questions. With multilingual support built in, we also made sure that people from different parts of the world can engage with the platform in their own language, making the experience even more inclusive.

Of course, we also encountered some challenges. The Chatbot's knowledge is only as strong as the information it's trained on, so keeping the knowledge base up-to-date and expanding it over time will be very important. Also, even with multilingual support, capturing the true cultural and historical nuances in every language remains an area for future improvement. Some of the future enhancements that can be considered are to provide a dynamic knowledge base, with multimedia integration, interactive quizzes and games based on the knowledge base etc. The Streamlit Agraph library can be explored, for storing the ever growing dynamic knowledge graph, so we can successfully deploy the project on streamlit cloud.

IV. Acknowledgment

We are grateful to Dr. Kishore Bhosale (Professor, Department of Botany) for providing us with multiple documents on Chhatrapati Shivaji Maharaj, that formed the knowledge base for testing our chatbot. Further we express our gratitude to Mr. ArunKumar Nair (Founder, Canspirit.AI) who provided the learning platform to learn all the technologies related to creating AI based chatbots.

References

[1] F.Yu, "Building A Grapg+LLM powered RAG Application from PDF Documents," Medium.com, 15 January 2024.
[2] C. Galleta, "Created a Neo4j Search Engine with Generative AI," Medium.com, 13 March 2024.
[3] S. Nawaz, "How to Use Groq API: The Comprehensive Guide You Need," ampcom.com, 26 March 2024.
[4] Siddhant Agarwal, "Mahabharatha_Graph_Modeling," Medium.com,14 March 2024.
[5] S. Savita," Realtime Powerful RAG Pipeline using Neo4j(Knowledge Graph Db) and Langchain", Youtube.com. 2024.
[6] S. Mondal, "RAG with knowledge graph, Advance RAG, Graph RAG with Neo4J, Streamlit RAG", Technow_With_Me (Youtube.com), 2024.
[7] Zilliz, "GraphRAG Explained: Enhancing RAG with Knowledge Graphs," Medium.com, 8 July 2024.
[8] Kamala Bot, "Bring Your Data To Life By Graphing It On Streamlit: Create Knowledge Graphs With Streamlit." Github.com, 3 August 2024.
[9] S. Shamasna, "RAG X: Graph RAG," 9 September 2024. https://blog.gopenai.com.
[10] S. Hedden, "Graph RAG Using Knowledge Graphs and Vector Databases," Medium.com, 6 September 2024.
[11] J. Sasson, "Local Rag From Scratch," towardsdatascience.com, 5 November 2024.
[12] S. Singh, "Knowledge Augmented Generation (KAG) is the Best Approach to RAG", Medium.com, 12 December 2024.
[13] S. Hedden, "How to Build a Graph RAG App," Medium.com, 30 December 2024.
[14] Z. Y. Rong Wu, "Do AI Chatbots improve students learning outcomes? Evidence from a meta-analysis," 3 March 2023. https://bera-journals.onlinelibrary.wiley.com.
[15] C. Kooli, "Chatbots in Education and Research: A Critical Examination of Ethical Implications and Solutions," 23 March 2023, https://www.mdpi.com.
[16] M. G. L. M. Lasha Labadze, "Role of AI Chatbots in education: Systematic literature review," 31 October 2023, https://doi.org/10.1186/s41239-023-00426-1.
[17] M. G. L. M. Lasha Labadze, "Interacting with educational Chatbots : A systematic review," 9 July 2022. https://doi.org/10.1007/s10639-022-11177-3.
[18] E. A. S. M. MoneerhAleedy, "Using AI Chatbots in Education: Recent Advances Challenges and Use Case", November 2022.https://link.springer.com.
[19] F. Z. C. Q. Qingyu Guo, "A survey on Knowledge graph-based recommender systems," 2022. https://ieeexplore.ieee.org/document/9216015.
[20] J. S. Sebastian Wollny, "Are We There Yet? - A Systematic Literature Review on Chatbots in Education," 15 June 2021, https://doi.org/10.3389/frai.2021.654924.
[21] A. A.-I. Chinedu Wilfred Okonkwo, "Chatbots applications in education : A systematic review," 2021, https://doi.org/10.1016/j.caeai.2021.100033.

[22] C. E. Shanshan Yang, “Opportunities and Challenges in Using AI Chatbots in Higher Education,” 15 January 2020. https://doi.org/10.1145/3371647.3371659 .
[23] T. D. Jose Quiroga Perez, “Rediscovering the use of Chatbots in education: A systematic literature review,” 3 September 2020,https://onlinelibrary.wiley.com.
[24] B. W. T. Sam, “A Review of Chatbots in Educaion : Pratical Steps Forward,” 2019. https://eprints.qut.edu.au.
[25] M. S. Rainer Winkler, “Unleashing the Potential of Chatbots in Education: A State-of-The-Art Analysis,” 9 July 2018, https://journals.aom.org.
[26] F. C. D. S. Fabio Clarizia, “Chatbot: AN Education support System for Student,” 23 September 2018, https://link.springer.com.
[27] Diary of Shivaji Maharaj: by Namadevrao Jadhav | Published by:-Rajmata Prakashan| 1 January 2013.
[28] “Gemini API quickstart,”, https://ai.google.dev/gemini-api.
[29] “StreamlitFramework,” Blog.com.
[30] “Neo4j Vector Index,” Python.langchain.com.
[31] “To Build Chatbot Using Neo4j,” Github.com.
[32] “Neo4j Graph Database,” Graphacademy.neo4j.com.