Efficient Course Information Retrieval: An AI-Powered University Syllabus Chatbot

Tharun Jayaprasad
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
tharun.jayaprasad2020@vitstudent.ac.in

Chitteshwari Satish
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
chitteshwari.satish2020@vitstudent.ac.in

MD Danish Anwar
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
mddanish.anwar2020@vitstudent.ac.in

Sanjna Srivastava
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
sanjna.srivastava2020@vitstudent.ac.in

Swarnalatha P*
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
pswarnalatha@vit.ac.in

***Abstract*—The VIT Syllabus Helper is a LLM chatbot designed for core courses in the Computer Science department at Vellore Institute of Technology (VIT). The chatbot leverages a combination of state-of-the-art open-source technologies to provide robust functionality, including Sentence Transformers for embeddings, FAISS CPU for vector storage, and integration of Llama 2, a large language model, using the Chainlit library for an interactive conversational interface. The project's key focus is on improving the learning experience for students at VIT by providing an intelligent and efficient tool for accessing course-related information. By utilizing Sentence Transformers, the chatbot can understand and process natural language queries with higher accuracy and context awareness. Additionally, Faiss CPU is employed to efficiently store and retrieve vector representations of text, compensating for GPU's speed with its easy accessibility.**

***Index Terms*—Llama 2, LLM, Faiss, Chainlit**

I. Introduction

A. Scope

The scope of this endeavor is to create an all-encompassing open-source chatbot tailored to serve the Computer Science department at VIT comprehensively. It will enable students and faculty to access detailed information about each core course, including the breakdown of lab work, theory syllabus, and project components. The chatbot will provide insights into the credit distribution for every course, facilitating a deeper understanding of the academic structure. Moreover, it will assist users in identifying and fulfilling course prerequisites, ensuring a smooth academic journey, without getting hassled by the cryptic syllabi found on VTOP.

B. Objective

The main objective of this paper is to introduce a versatile and comprehensive chatbot to the Computer Science department at VIT. This chatbot will serve as a centralized resource for students and faculty seeking information about core courses, offering insights into both theoretical and practical components, including lab assignments and project work. Additionally, it will provide clarity on course credits, helping students make informed academic decisions.

Another important aim of the chatbot is to simplify the process of identifying and fulfilling prerequisites for each course. It will offer guidance on the foundational knowledge needed for success in specific courses, ultimately enhancing students' academic achievements and satisfaction. The open-source nature of the chatbot underscores its adaptability and sustainability, ensuring it can evolve to meet the changing needs of the Computer Science department at VIT, ultimately leading to improved academic outcomes and enhanced student experiences. Presentations are communication tools that can be used as demonstrations, lectures, speeches, reports, and more.

II. Literature Review

In their research paper, 'The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research' [1] the authors, Tariq Alqahtani, Hisham A. Badreldin, Mohammed Alrashed, Abdulrahman I. Alshaya and Sahar S. Alghamdi thoroughly examines the transformative impact of Artificial Intelligence (AI), Natural Language Processing (NLP), and Large Language Models (LLMs) in higher education and research. The

exploration encompasses applications such as personalized learning, grading, curriculum design, and career guidance in education, delving into AI's contributions to text generation, data analysis, literature reviews, and peer review processes in research. The survey also addresses the emerging use of NLP models for mental health support. Throughout the paper, the author emphasizes the imperative for responsible AI use, ethical considerations, and advocates for a balanced integration of AI and human support to optimize outcomes in education and research.

The literature survey of the paper,' Leveraging Large Language Models to Power Chatbots for Collecting User Self-Reported Data' [2], explores the application of LLMs (Large Language Models), such as GPT-3, in the development of chatbots for collecting user self-reported data in the context of digital health. The authors, Jing Wei, Sungdong Kim, Hyunhoon Jung, and Young-Ho Kim, highlight the limitations of existing commercial chatbot frameworks, emphasizing the need for more flexible and dynamic conversational agents. The study investigates the potential of LLMs, with billions of pre-trained parameters, to power chatbots capable of engaging in naturalistic conversations and effectively collecting self-report data on health-related topics. The research evaluates the impact of prompt design factors, including information specification format and personality modifiers, on the slot-filling ability and conversation styles of the resulting chatbots. The authors discuss the advantages, such as versatile responses, context tracking, and low-effort bootstrapping, as well as drawbacks, including randomness and repetitiveness, of LLM-driven chatbots. The study also addresses ethical considerations and proposes strategies to mitigate potential issues. Overall, the literature survey contributes empirical insights into the feasibility and challenges of leveraging LLMs for developing chatbots focused on data collection in the field of personal informatics.

The literature survey of the paper,' Llama 2: Early Adopters' Utilization of Meta's New Open-Source Pretrained Model ' [3], delves into the burgeoning field of artificial intelligence (AI), focusing on the introduction and early adoption of Llama 2, an open-source pre-trained model released by Meta. Authored by Konstantinos I. Roumeliotis, Nikolaos D. Tselikas, and Dimitrios K. Nasiopoulos, the survey investigates the foundational elements of Llama 2 and explores how early adopters leverage its capabilities in AI projects. The authors emphasize the significance of understanding the perspectives and experiences of these early adopters, who play a pivotal role in driving innovation and providing insights for further model enhancements. The survey provides insights into the strengths, weaknesses, and areas of improvement of Llama 2, offering valuable guidance for the AI community and Meta to enhance future model iterations. Furthermore, it discusses the implications of Llama 2's adoption on the broader open-source AI landscape, addressing challenges and opportunities for developers and researchers. This early exploration of the Llama 2 pre-trained model serves as a foundational basis for future research investigations, shedding light on the practical implications and ethical considerations associated with its use in the evolving landscape of AI technologies.

The paper titled "Chat Vector: A Simple Approach to Equip LLMs with New Language Chat Capabilities" by Shih-Cheng Huang et al [4]. explores the development of Large Language Models (LLMs) for non-English languages, with a focus on aligning them with human preferences. The authors propose a computationally efficient method called "chat vector," which involves restructuring the conventional training paradigm from continual pre-training, Supervised Fine Tuning (SFT), and Reinforcement Learning from Human Feedback (RLHF) to continual pre-train + chat. The chat vector is derived by subtracting the pre-trained weights of a base model (LLaMA2) from its chat-enhanced counterpart (LLaMA2-chat). The empirical studies primarily focus on Traditional Chinese, with evaluations based on toxicity, ability to follow instructions, and multi-turn dialogue. The results demonstrate the chat vector's efficacy in improving conversational skills. The approach is extended to models pre-trained in Korean and Simplified Chinese, showcasing its versatility. The paper contributes a significant solution for aligning LLMs with human preferences efficiently across various languages, achieved through the innovative concept of the chat vector.

The literature survey of the paper, 'Effects of Generative Chatbots in Higher Education' [5], reveals a growing interest in the transformative impact of generative chatbots in higher education. Ilieva et al. (2023) highlight the challenges faced by traditional learning technologies in providing interactive and real-time feedback while emphasizing the potential of intelligent chatbots based on generative artificial intelligence (AI) to address these shortcomings. The authors propose a theoretical framework for blended learning with chatbot integration, offering advantages such as comprehensive understanding, enhanced educational experiences, and unified applications in teaching–learning activities within universities. The study underscores the role of generative chatbots in guiding both students and instructors, streamlining pedagogical activities, and reducing the workload on educators. Additionally, the paper explores the characteristics of existing educational chatbots, emphasizing their ability to provide conversational assistance, support multi-modality, offer multilingual capabilities, and integrate with other software systems. The survey also touches upon the rapid growth of the conversational AI market, with generative chatbots anticipated to play a significant role in reshaping the educational landscape. Furthermore, the study by Ilieva et al. (2023) provides insights into the application of chatbots in higher education, focusing on the assessment of their influence on students' learning experiences and instructors' teaching methods. The proposed conceptual framework contributes to the systematic evaluation of perceptions and readiness for chatbot-based learning, emphasizing the need for institutions to develop AI adoption strategies and invest in digital innovations. The literature survey, therefore, demonstrates a consensus on the potential of generative chatbots to revolutionize teaching and learning practices in higher education.

III. Methodology

The VIT Syllabus Helper chatbot was designed to serve the dual purpose of answering to user queries in a much more user friendly manner, as well as to leverage the power of generative AI on a CPU machine while making use of only open source tools. The process of achieving so is detailed in the following sections.

A. Application Planning

The problem statement and objective of the chatbot was decided and elaborated upon to identify the various subsystems that needed to be developed. The design and planning was carried out using HTA (Hierarchical Task Analysis) and STN (State Transition Network).

B. System Architecture and Requirements Specification

Once the plan was solidified, the formal architecture of the system was designed.

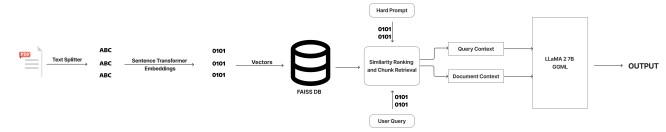

Fig. 1. System Architecture of The Chatbot

Based on this architecture, the hardware and software requirements were enlisted to build the application.

C. Data Collection

The data used for the chatbot is PDF copies of each subject mentioned in the Program Core of the Computer Science and Engineering (core) stream. Each subject's document has information related to the subject, in terms of credits, credit structure, course objectives, modules of the course, and reference and prescribed reading. This information is natively documented in tabular form, making it difficult for the processor to interpret. Thus, we modified each

D. Data Pre-processing

The data collected in the form of PDFs is first extracted using langchain package's PyPDFLoader library. The text was chunked with the help of the RecursiveCharacterTextSplitter library. Finally, the chunks of size 500 characters and a chunk overlap of 50 were embedded using the HuggingFaceEmbeddings library.

E. Data Transformation

The text embeddings created were then converted into intercommunicating neural networks using the Sentence Transformers library [H1 Mini LM v6]. The transformed vector embeddings were stored in the FAISS (Facebook AI Similarity Search) database. FAISS was opted as the vector stored due to it being open-source, more efficient because of the lack of latency and storage of metadata of the vectors. The database stored the embeddings as well as retrieved the relevant information related to the query and our data, based on the similarity index of vectors and clustering methods.

F. User Query and Prompt Engineering

The query provided by the user is called the user prompt. The user prompt undergoes the same pre-processing and transformation process as that of the context data corpus. Thus, the query was transformed into a vector embedding. Along with the user prompt, we also include a hard prompt. The purpose of the hard prompt was to avoid hallucinations [6] by the LLM.

```
 Use the following pieces of information
to answer the user's question.
If you don't know the answer, just say
that you don't know, don't try to make
up an answer. If the answer cannot be
found in the given context information,
do not answer it. Just say that the answer
cannot be found in the VIT Syllabus. Do
not answer queries of personal, political,
world affairs, trivial, medical or
subjective nature. Only answer questions
pertaining to the field of computer
science.
Context: {context}
Question: {question}

Only return the helpful answer below and
nothing else.
Helpful answer:
```

The aforementioned hard prompt was fed to the LLM using the PromptTemplate library. The queries were then passed through a similarity index and chunk retrieval system.

G. Query Look-up and Response Processing

The RetrievalQA library provided the sequential chain infrastructure for inducing our data corpus into the LLM's context. The LLM used for the chatbot was a quantised version of LLaMA 2 7B [7]. In order to use the model, the model needed to be locally hosted/loaded on our CPU machine. The CTransformers library was used for this. CTransformers library is Python bindings for the Transformer models written in C/C++ using the GGML library. Finally, The LLM takes in the hard prompt, user prompt and data corpus to generate the final output that is desirable for the user, along with providing its source.

H. User Interface

The user interface was constructed using Chainlit. Chainlit is an open source python package tailored specially for developing chatbot user interfaces. It has multiple integrations with a lot of popular libraries such as Langchain, OpenAI Assistants, Llama Index and so on. The user interface developed for the syllabus chatbot was made to be intuitive, and easy to use. Token streaming, cache and search history enabling, and buttons for prompt examples were also implemented in order to give users the best and positive

experience.

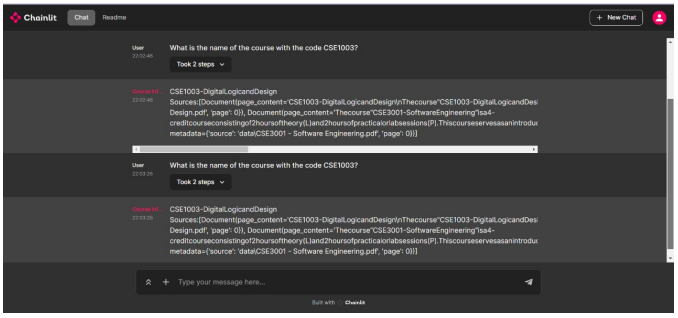

Fig. 2. Chatbot UI using Chainlit

IV. Comparative Study

A. GOMS Analysis

To access information about the VIT course syllabus, you can follow two distinct methods. One is the native method of downloading the syllabus page from VTOP and deciphering the keywords and the other is via the proposed chatbot. The sequential complexity of both these methods are analyzed using GOMS (Goals Operations Methods and Selection Rules).

1) Native VTOP Lookup: Start by opening the VTOP website, logging in with your credentials (username and password), and entering the captcha. Once logged in, navigate to the curriculum section by clicking on the menu, selecting academics, and then choosing the curriculum option. Locate the specific program or course of interest, download the syllabus, and open the downloaded file to read the entire syllabus.

2) Chatbot Query Entry: If you have a chatbot link, initiate the process by moving the mouse to the chatbot input box, clicking on it, and typing your question. After entering your query, press the enter key, wait for the chatbot response, and carefully read the provided information. When engaging with a chatbot, users can opt for the "Use-Type-Method" to type and ask new questions directly or the "Use-Question-History-Method" to select a pre-existing question from the chat history. The decision is guided by simple rules: if the question is already in the history, use the latter method; otherwise, employ the former. These methods are designed to enhance user interactions by accommodating both new inquiries and the retrieval of previous questions from the chat history.

```
  GOMS FOR CHATBOT QUESTION ENTRY:
GOAL: USE-CHATBOT
     [select* GOAL: USE-TYPE-METHOD
          MOVE-MOUSE-TO-CHATBOT-INPUT-BOX
          CLICK-INPUT-BOX
          GOAL: USE-TYPE-QUESTION
             HOME-TO-KEYBOARD
             FORMULATE-QUESTION
             TYPE-QUESTION
          VERIFY-QUESTION
             PRESS-ENTER-KEY
        GOAL: USE-QUESTION-HISTORY-METHOD
          MOVE-MOUSE-TO-PREVIOUS-CHAT-
QUESTIONS-ICON
          CLICK-ON-ICON
          SCROLL-THROUGH-CHAT-HISTORY-
QUESTIONS
          CLICK-ON-REQUIRED-QUESTION
          VERIFY-QUESTION
          PRESS-ENTER-KEY]

  SELECTION RULES:

  RULE 1: If the question to be asked to
the chatbot has already been asked before
and is present in the question history,
use the USE-QUESTION-HISTORY-METHOD
RULE 2: Else, use the USE-TYPE-METHOD
```

B. KLM Analysis

The KLM (Keystroke-Level Model) was used to estimate the time taken by each of the approaches by assigning specific weights to keyboard and mouse actions. The operations considered were K (Keystroke), P (Mouse Point), B (Mouse Click), M (Mental Preparation), H (Hover) and R (System Response). The time taken by an average user to perform these operations were considered while analyzing the approaches.

Method	Equation	Time (in seconds)
VTOP	(2+n1+n2)K + 5P + 5B + 10M + 2*H	21.36 + (n1+n2)*0.28
Chatbot - Without History	(1+m1)K + 2P + 2B + 4M + 1H + 1R	188.68 + m1*0.28
Chatbot - With History	4*P + 4*B + 4*M + 1*R	190.6

TABLE I
KLM Analysis for Searching Methods

1) Native VTOP Method: The formula (2+n1+n2)K + 5P + 5B + 10M + 2*H calculates the estimated time for the entire process, emphasizing the higher weights on keyboard actions and mouse movements. This method involves moving the mouse, clicking on fields, pressing keys, and hovering over elements, with (n1) and (n2) representing variables for username and password.

2) Chatbot - Without History: The formula (1+m1)K + 2P + 2B + 4M + 1H + 1R calculates the estimated time for the entire interaction, with m1 keystrokes being utilized to type the user query. This method outlines the user's steps, from initiating the chatbot to receiving a response, providing insights into efficiency and guiding design optimizations for chatbot applications.

3) Chatbot - With History: The formula 4*P + 4*B + 4*M + 1*R calculates the estimated time for the entire interaction. In practical terms, users navigate to the chatbot, access the history, select a question, send a query, and receive a response. This analysis offers insights into the efficiency of incorporating chat history within the chatbot application, potentially guiding enhancements for improved user experience and system optimization.

V. Result and Discussion

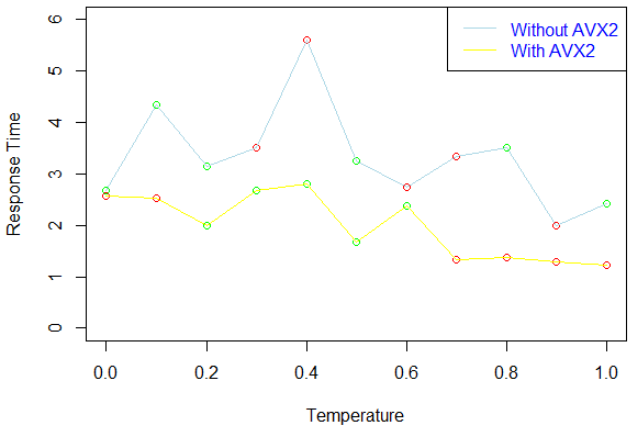

Fig. 3. LLM Temperature Vs. Response Time (minutes) Plot

The execution of the chatbot using Llama 2 for various temperatures between the range of (0,1) shows that:

Computation Time is significantly reduced while using FAISS DB with AVX2 support as compared to without. The points in the graph plotted in Green refer to an accurate result while the Red points refer to an inaccurate result. We note that while executing with AVX2 support, the accurate results are obtained when the temperature is between 0.2 and 0.7. On the other hand, without AVX2 support, the correlation between temperature and accuracy seems to be less significant. Temperature values of 0.2 and 0.5 give accurate results in both cases.

Therefore, the results show that accurate results can be obtained from the chatbot in as low as 1.5 minutes, with just one click of a button. When compared with the traditional method of looking up a syllabus on VTOP, this proves to be not just low effort but also faster overall.

VI. Conclusion

In conclusion, the creation of a chatbot tailored for Vellore Institute of Technology's Computer Science department's Program Core subjects marks a pivotal advancement in enhancing the educational experience for both students and faculty. This innovative tool, designed to offer comprehensive course-related information, holds the promise of transforming how academic information is accessed and comprehended within the department. It provides detailed insights into core courses, credit distributions, and course prerequisites with increased performance than the traditional method, i.e., in lesser steps and reduced time, all while running on just a CPU machine.

With its integration of the latest cutting-edge open-source technologies in the market like Llama 2, the chatbot assures user-friendliness and scalability. It has far reaching future scope, as its functionality can be increased to meet the demands of all other categories of courses and in different branches in the university. Furthermore, it can also be expanded to other universities. In essence, this comprehensive open-source chatbot has the potential to foster a more informed and efficient academic journey for students and faculty alike.

Acknowledgment

We would like to express our sincere gratitude to our esteemed teacher, Mrs. Swarnalatha P, whose guidance and expertise were instrumental in the successful completion of this project. Her unwavering support and valuable feedback greatly contributed to our learning and the overall quality of our work. Additionally, we extend our appreciation to Vellore Institute of Technology for providing a conducive learning environment and resources that empowered us to undertake and complete this project.

References

[1] Huang, Shih-Cheng, et al. "Chat Vector: A Simple Approach to Equip LLMs With New Language Chat Capabilities." arXiv preprint arXiv:2310.04799 (2023).

[2] Wei, Jing, et al. "Leveraging large language models to power chatbots for collecting user self-reported data." arXiv preprint arXiv:2301.05843 (2023).

[3] Roumeliotis, K.I.; Tselikas, N.D.; Nasiopoulos, D.K. Llama 2: Early Adopters' Utilization of Meta's New Open-Source Pretrained Model. Preprints 2023, 2023072142.

[4] Tariq Alqahtani, Hisham A. Badreldin, Mohammed Alrashed, Abdulrahman I. Alshaya, Sahar S. Alghamdi, Khalid bin Saleh, Shuroug A. Alowais, Omar A. Alshaya, Ishrat Rahman, Majed S. Al Yami, Abdulkareem M. Albekairy, The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research, Research in Social and Administrative Pharmacy, Volume 19, Issue 8, 2023

[5] Zheng, Zhonghua, et al. "Building emotional support chatbots in the era of llms." arXiv preprint arXiv:2308.11584 (2023).

[6] Martino, Ariana, Michael Iannelli, and Coleen Truong. "Knowledge injection to counter large language model (LLM) hallucination." European Semantic Web Conference. Cham: Springer Nature Switzerland, 2023.

[7] Touvron, Hugo, et al. "Llama 2: Open foundation and fine-tuned chat models." arXiv preprint arXiv:2307.09288 (2023).

[8] Wikipedia contributors. "GOMS." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 7 Apr. 2023. Web. 21 Nov. 2023.

[9] Wikipedia contributors. "Human processor model." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 3 May. 2023. Web. 21 Nov. 2023.

[10] Wikipedia contributors. "Vector database." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 21 Nov. 2023. Web. 21 Nov. 2023.