Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 10:2023
DOI: 10.5281/zenodo.10005991

USING CHATBOTS AS TEACHING ASSISTANT WITH OPENAI

SMIT MINESHKUMAR PATEL

BE Student, Information Technology, Shree Swaminarayan Institute of Technology, Bhat Gandhinagar, India. Email: smit.p1707@gmail.com

GAUTAM MINAXIBEN PRAJAPATI

BE Student, Information Technology, Shree Swaminarayan Institute of Technology, Bhat Gandhinagar, India. Email: gautamprajapati2252@gmail.com

VIVEK DILIP MANGLANI

BE Student, Information Technology, Shree Swaminarayan Institute of Technology, Bhat Gandhinagar, India. Email: manglanivivek53@gmail.com

Dr. RAMESH T. PRAJAPATI

Professor, Information Technology Department, Shree Swaminarayan Institute of Technology, Bhat Gandhinagar, India. Email: hod.it.ssit@gmail.com

DARSHAN P. PATEL

Assistant Professor, Computer Engineering Department, Shree Swaminarayan Institute of Technology, Bhat Gandhinagar, India. Email: hod.computer.ssit@gmail.com

JYOTI V GAUTAM

Assistant Professor, Information Technology Department, Shree Swaminarayan Institute of Technology, Bhat Gandhinagar, India. Email: jyoti.manglani2021@gmail.com

Abstract

In this growing era of technology, everyone's topic of discussion revolves around Artificial Intelligence. With the access of AI technologies being present at people's fingertips, everyone can access any type of information using Chatbots, especially students. Students nowadays are using these LLM models such as ChatGPT and Google Bard. As these models are trained on data, they can provide more than needed information leading to a decrease in time efficiency. TA-GPT is an AI model made for students to help them in their curriculum. TA-GPT is based on OpenAI's GPT-3.5. TA-GPT helps students to understand things better than a normal LLM model as it would be trained particularly for their interest only. TA-GPT can be trained for particular organizations or particular subjects using PDFs. TA-GPT contributes to AI's overall development in the digital society as students instead of concentrating on non-related data around their subjects or curriculum provided by LLM models and available on the internet, can mainly focus on what's covered in their curriculum leading to time efficiency and accuracy.

Keywords: TA-GPT, LLM, AI, ChatGPT, Google Bard, Education, PDF.

1. INTRODUCTION

With the surroundings revolving around artificial intelligence everywhere, people are adapting AI **[1]** technologies faster than expected. Artificial Intelligence is the ability of machines to perform tasks like learning and solve problems similar to human intelligence. Research on AI was started from as long as 1951 but has shown an immense trend over the last 10 years. A huge number of people have considered AI as a sudden revolution in human growth and look at AI curiously as well as a threat to society. From the time of AI Voice Assistants in phones, AI has now become part of life for a huge number of individuals.

With its Contribution to every field, AI has also contributed immensely to education and research. It has changed the way people learn. With the ability to analyze the patterns and preferences of each student, AI can now provide insights leading to better-personalized learning. As AI has made it easier to access information about anything, it has increased the time efficiency of people by changing how they look after resources and learn from them. Also natural language processing (NLP) **[2]** technology allows AI to analyze written or spoken language and NLP tools can assist with language learning, vocabulary suggestions and grammatical corrections.

As the well-renowned LLM **[3]** models like OpenAI's ChatGPT **[4]** or Google Bard **[5]** are goto options for students to get information about their topic of interest, these LLMs are trained on huge chunks of data collected over the years. This leads to the delivery of more than sufficient data needed by the students. This can lead to time deficiency and a decrease in accuracy a student might get from their books made specifically for their curriculum. TA-GPT can hold this widening gap in these LLM models with the same NLP efficiency as them. TA-GPT will exclusively provide data related to the students' curriculum and subjects. This will reduce the challenges students face when trying to comprehend complicated or complex information generated by LLMs, allowing them to concentrate on their studies more effectively.

2. LITERATURE REVIEW

Lijia Chen et al (2020) **[6]**, they extensively reviewed the role of artificial intelligence (AI) in education. The paper explores how AI is integrated into education, covering tasks like administration, student assessment, grading, and creating personalized curricula. It also discusses various forms of AI in education, especially online platforms. The paper highlights AI's adaptability, as it can adjust to students and teachers to enhance the learning experience. Additionally, it explores AI applications such as assessing students and schools, automating grading, personalized teaching, smart schools with technologies like facial recognition and virtual labs, and online and mobile education with virtual assistants. One significant benefit of AI is its ability to tailor content to each student's needs and abilities. Ultimately, the paper's analysis focuses on how AI impacts education in administrative, instructional, and learning aspects.

Olaf Zawacki-Richter et al (2019) **[7]** conducted a comprehensive survey that delves into the profound pedagogical benefits of integrating Artificial Intelligence (AI) into education, highlighting its potential to significantly transform the landscape of teaching and learning. The paper explores four overarching domains: profiling and prediction, intelligent tutoring systems, assessments and evaluation, and adaptive systems and personalization, each encompassing a multitude of subcategories. These subcategories include areas such as admissions decisions and course scheduling, student retention, academic achievement, personalized content recommendations, and the use of academic data to monitor and guide students, among others. The research reveals a striking observation - the limited number of implementation and impact studies in this critical field, signaling a need for further exploration and substantiation of AI's true potential in education. Jules White et al (2023) **[8]** explore the flourishing significance of prompt engineering as an indispensable skill in effectively communicating with Large Language Models (LLMs) like ChatGPT. The study illuminates the profound influence that simple prompts wield in directing LLMs to perform tasks according to specific objectives, underscoring how prompts act as guiding instructions for LLM behavior. By adeptly crafting prompts, users can harness the versatility of LLMs to serve their unique purposes while ensuring precise control over the nature and volume of generated output. The paper introduces a pattern-based catalog of prompt engineering techniques, systematically addressing common challenges encountered in LLM interactions.

Nimit Thaker and Abhilash Shukla (2020) **[9]** extensively explored the versatile programming language Python and its wide-ranging applications. The paper underscores Python's status as a multi-paradigm programming language and delves into its evolution as a universally favored choice for diverse development tasks. It elucidates why Python has garnered popularity across various domains such as the Internet of Things (IoT), Machine Learning (ML), Deep Learning, Artificial Intelligence (AI), and Cybersecurity. Furthermore, the paper highlights Python's exceptional flexibility, incorporating an exhaustive array of functionalities, making it an ideal choice for AI, Machine Learning, and Deep Learning. Python's allure in these fields is attributed to its rich repository of tools, frameworks, packages, modules, and libraries, readily accessible for practitioners, and its capacity for users to craft their own solutions tailored to their specific needs. Which makes Python an obvious choice for Artificial Intelligence application development.

3. PROBLEM STATEMENT

In today's dynamic educational landscape, the integration of artificial intelligence (AI) presents a groundbreaking opportunity to reshape the dynamics of learning and teaching. One particularly promising application of AI in the educational domain is the development of a Teaching Assistant GPT (TA-GPT), harnessing the advanced capabilities of OpenAI's ChatGPT technology. However, despite the immense potential, the practical implementation and real-world impact of such a system within educational institutions remain largely uncharted territory.

This research paper endeavors to delve into the nuanced challenges and promising opportunities entailed by the deployment of a TA-GPT, empowered by ChatGPT **[10]**, within educational institutions. In contemporary education, students increasingly turn to AI-powered tools to enhance their study experiences. Yet, a pertinent issue arises when employing Chatbots **[11]** such as ChatGPT or Google Bard for academic purposes — the occasional provision of superfluous and non-relevant information, often diverging from the student's specific curriculum. Such extraneous information not only undermines efficiency but also consumes valuable study time. This challenge stems from the Chatbot's lack of contextual knowledge about the student's academic institution, including its unique syllabus. To address this issue, there is a compelling need to devise a system that tailors the Chatbot's responses exclusively to the subject matter that aligns with the student's university or college curriculum. By furnishing the Chatbot with a restricted dataset that reflects the pertinent subjects and courses offered by the student's academic institution, we can steer the AI toward providing highly focused and curriculum-specific assistance.

The primary objective of this TA-GPT is to offer precise answers to questions directly related to subjects and courses relevant to the student's educational institution. Additionally, it aims to assist students in completing assignments and resolving doubts, especially those that students may be hesitant to ask their professors. By shouldering some of the burden of answering individual queries, the TA-GPT can alleviate the workload of educators, creating a more streamlined and efficient learning experience for all stakeholders.

This research paper seeks to contribute to the discourse surrounding AI in education by proposing a practical solution that enhances AI-powered educational assistance, making it a valuable and tailored resource that not only supports students but also integrates seamlessly with the unique academic requirements of each institution.

3.1 Proposed Architecture

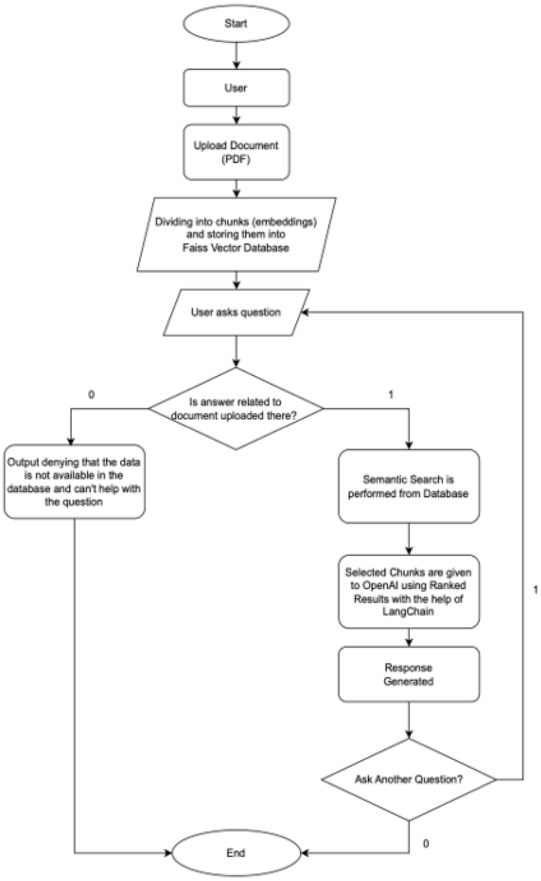

Start
User
Upload Document (PDF)
Dividing into chunks (embeddings) and storing them into Faiss Vector Database
User asks question
0
Is answer related to document uploaded there?
1
Output denying that the data is not available in the database and can't help with the question
Semantic Search is performed from Database
Selected Chunks are given to OpenAI using Ranked Results with the help of LangChain
1
Response Generated
Ask Another Question?
0
End

Fig 1: Working Flowchart of TA-GPT

Utilizing Teaching Assistant GPT (TA-GPT) effectively involves a structured process of inputting data in the form of PDF **[12]** documents, which the chatbot can subsequently access. Once this data ingestion process is complete, the chatbot proceeds to segment the information into manageable chunks or embeddings, subsequently storing them within a vector database. This strategic storage method enables the chatbot to swiftly retrieve relevant information when queried.

When a user poses a question related to the topic at hand, the chatbot assesses the query in real-time. If the requested information is readily available in the database, the chatbot promptly generates an informative response. However, if the queried information isn't present within its stored knowledge, the chatbot takes a courteous approach, apologizing for the absence of data on that particular topic. It then politely requests the user to rephrase the question or inquire about a topic within the established dataset.

In cases where the user's question pertains to the ingested data, the chatbot employs a systematic approach to provide a precise response. It leverages a ranked results methodology, selecting the most relevant information from its stored embeddings to construct a comprehensive answer.

This approach ensures that the user receives accurate and contextually appropriate responses, optimizing their interaction with the TA-GPT and enhancing their overall learning or information-seeking experience.

4. METHODOLOGY

4.1 Approach

We wanted our TA-GPT to be as good as other LLM models, so we needed a LLM model with strong NLP architecture to generate answers that are simple enough to be understood by a student. Therefore, we decided to use OpenAI's GPT-3.5 API **[13]** to integrate with our model. To make it specifically for a student's curriculum, a PDF that consists of data on a particular subject/curriculum is uploaded to the database. Our model has been precisely trained to emulate a Data Structure Professor specialising in teaching Data Structures in the C Language, closely aligned with the curriculum of Gujarat Technological University.

It serves as a valuable resource for students, offering comprehensive assistance in resolving their doubts and queries. To ensure precision and relevance, we've introduced a carefully crafted prompt encapsulated within the following marker: '// The Prompt //'—this directive guides our model in delivering information that aligns with predefined criteria for both quantity and quality of data, ensuring an optimal learning experience for students. The LLM model is then trained on the data of these PDFs. Then the LLM model is proposed to generate answers according to the data of the PDFs only.

4.2 Technical Tools

In the development of TA-GPT, we have used a variety of tools that ease the process of development of the TA-GPT.

The tools which are used in the development of the TA-GPT are mentioned below:

4.2.1 Python3

Python **[14]** is an excellent choice for the development of TA-GPT due to its simplicity, extensive libraries, and community support. Python served as a keystone by providing the foundation upon which TA-GPT's functionalities are built. Python enables the creation of web interfaces which helps create cross-platform deployment while ensuring broad accessibility. Python streamlines TA-GPT development by combining simplicity, versatility, and innovation resulting in a user-friendly application.

4.2.2 OpenAI

We have integrated the API key of OpenAI's GPT-3.5 in the TA-GPT. It creates an enriching educational experience. GPT-3.5's language understanding and generation capabilities enable the TA-GPT to provide insightful and appropriate explanations by having cutting-edge technology. GPT-3.5 empowers the TA-GPT to offer personalized human teaching assistance. It enhances the TA-GPT's role as an effective educational companion favoring among students.

4.2.3 Streamlit

Streamlit **[15]** is an open-source Python library that makes it easy to create and share beautiful, custom web apps for applications. Streamlit facilitates the creation of a user-friendly interface, enhancing the TA-GPT's accessibility for students. By using Streamlit the TA-GPT is a potent application that gives students a convenient Graphical User Interface (GUI).

4.2.4 Langchain

Langchian **[16]** is a framework for developing AI chatbot applications powered by language models. Langchain is used in the TA-GPT because it allows us to connect a LLM to other sources of information such as databases or the internet. Langchain works by connecting the LLM to the desired data source. Langchain also provides features like Conversation History, context awareness, and Error handling to improve the performance of TA-GPT. This integration elevates the TA-GPT's effectiveness by making it an intelligent and adaptive educational companion.

4.2.5 PyPDF2

PyPDF2 **[17]** is an open-source Python PDF library capable of splitting, merging, cropping, and transforming the pages of PDF files. It can retrieve text and metadata from PDFs as well. This empowers the TA-GPT to furnish students with accurate references that can seamlessly incorporate a wide array of academic materials and provide efficient access to information for students.

4.2.6 Faiss

Faiss **[18]** is a library for efficient similarity search and clustering of dense vectors. Faiss builds vector databases that store vectors, which are typically high-dimensional data points that are used to store and retrieve data that can be represented as vectors, such as images, texts, or audio. It has features that make TA-GPT by giving the capability of efficient similarity search, clustering, and dimensionality reduction. Faiss expertise in similarity search facilitates efficient retrieval of relevant study materials and resources, aligning with students' specific needs. By leveraging Faiss's capabilities the TA-GPT elevates its role as a comprehensive educational bot.

4.2.7 Tiktoken

Tiktoken **[19]** is a Python library that can be used to count the number of tokens in a string. A token is a unit of text that can be used to represent a word, phrase, or other meaningful unit of text. By effectively managing token consumption, the TA-GPT can generate comprehensive yet concise responses that convey relevant information. This integration not only ensures the TA-GPT's efficiency but also enhances its ability to maintain coherent and informative conversations with students

4.2.8 Vega-Altair

Vega-Altair **[20]** is a declarative visualization library for Python which makes it easy to create interactive visualizations that are both beautiful and informative. It enables the TA-GPT to present complex educational concepts through visually appealing charts and graphs.

5. IMPLEMENTATION

The below code shows the some of the functionality of the TA-GPT:-

```
def main():
    load_dotenv()
st.set_page_config(page_title="TA-GPT",
                       page_icon=":books:")
    st.write(css, unsafe_allow_html=True)

    if "conversation" not in st.session_state:
        st.session_state.conversation = None
    if "chat_history" not in st.session_state:
        st.session_state.chat_history = None

    st.header("TA-GPT :books:")
    user_question = st.text_input("Ask your question here:")
    if user_question:
        handle_userinput(user_question)

    with st.sidebar:
        st.subheader("Your documents")
        pdf_docs = st.file_uploader(
            "Upload your PDFs here and click on 'Process'",
accept_multiple_files=True)
        if st.button("Process"):
            with st.spinner("Processing"):
                # get pdf text
                raw_text = get_pdf_text(pdf_docs)

                # get the text chunks
                text_chunks = get_text_chunks(raw_text)

                # create vector store
                vectorstore = get_vectorstore(text_chunks)

                # create conversation chain
                st.session_state.conversation = get_conversation_chain(vectorstore)
if _name_ == '__main__':
    main()
```

Above is the code for a function called main (). The program begins with this function. This function is well-structured and organized. It uses descriptive variable names and follows the standard Python coding style. It is also modular, making it easy to read and understand.

From the parent directory, environment variables are loaded from a file called (dot)env. The configuration settings for the program, like api key, are contained in this (dot)env file. It has the GPT 3.5 API Key that helps in the function named get_converstion_chain. Then to make the app distinguishable from other apps, it sets the title and icon of the webpage for the Streamlit app. After that, a check is done to verify if the state of the session has a value for the conversation and variables of the chat-history. If not, it initializes them to None and ensures if the variables have their value, even if there is no interaction initiated between the users and the app.

To display a header with the title "TA-GPT " and an icon of the book, the function calls the st.header() function. Next, using the st.text_input() function, it receives the user's question. This input is then used to generate a response from the feeded data. If the user has a question as the input, the handle_userinput() function is called to handle the query using Natural Language Processing techniques to understand and generate a response for the same query.

Before that, we need the sidebar of the application, in which the function displays a subheader titled 'Your Documents'. The function then uses the st.file_uploader() function to let the user upload one or more PDF files. These PDFs are used to generate answers for the questions asked by the user with additional context for the question if asked. When the user uploads one or more PDFs, the function get_pdf_text() is called by the function to extract text from the PDF files using the PyPDF2 library. The function then calls the get_text_chunks() function to divide the data into smaller chunks of data, making it easier for the application to process the text. Meanwhile, it calls the get_vectorstore() function to convert these smaller chunks of text to vectors and store them in the vector store. These vectors make it easier for the application to compare the text chunks. Finally, to create a chain of conversations from the vector store, the get_conversation_chain() function is called and the conversation chain stores the history of the conversation between the user and the application.

By using LangChain, we are able to create a connection between the vector store containing the text chunks with the OpenAI's GPT-3.5 model to generate better and easier to understand responses.

The integration of PDF documents into our application serves as an important gateway to provide information to the TA-GPT system. This is achieved through the utilization of PyPDF2, a tool for the extraction of textual content from PDF files. The extracted content is then divided into manageable portions, a process done by PyPDF2's chunking mechanism.

The process starts as these portions of chunks find their place in the vector database, created by Faiss. This database, being suitable for similarity search, becomes the backbone for our analyses, ensuring that the TA-GPT system can access and retrieve information when required.

Guided by the robust framework of langchain, a seamless connection is established with OpenAI's GPT-3.5 model. This connection connects with data, allowing it to be transformed into a clear and contextually relevant form. The GPT-3.5 model, famous for its adaptiveness in language comprehension and generation, converts complex information into concise and cogent summaries, thereby enhancing the interpretability of the outcomes.

The efficiency of this process is further amplified through the sharp incorporation of Tiktoken, a tool proficient in quantifying token usage. By assessing the length of tokens, Tiktoken ensures that the outcomes remain compact and manageable, aligning with the requisite context and format.

All these tools and processes are made as one with the creation of a GUI web application powered by Streamlit. This interface, marked by its user-friendliness and interactivity, serves as the channel through which users can seamlessly upload PDF documents, trigger queries, and receive insights.

6. RESULTS

The following data describe the preferences exhibited by a group of 100 students within our college in relation to responses provided by distinct AI language models, namely TA-GPT, ChatGPT, and Google Bard. These preferences were recorded across various segments of Data Structure questions, with students being intentionally unaware of the source of each answer.

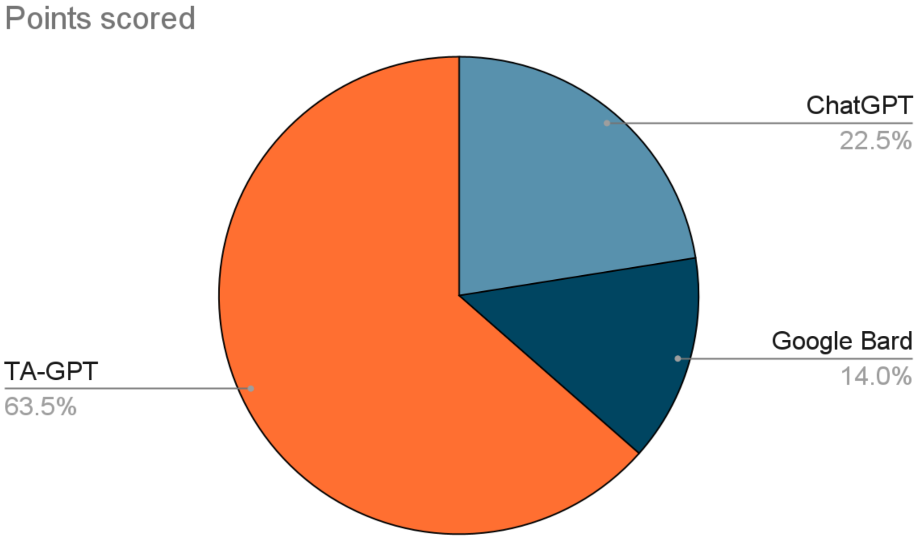
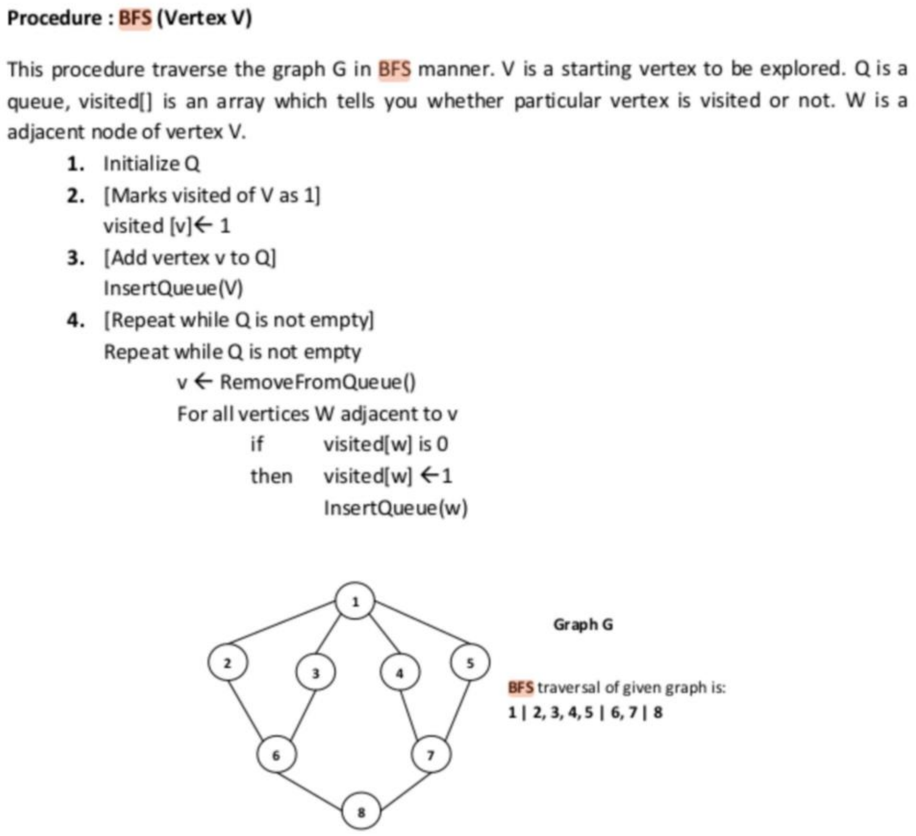


Fig 2: Comparison between ChatGPT, Google Bard and TA-GPT

For queries demanding theoretical explanations, a noteworthy pattern emerged. A substantial majority of 76 students expressed a preference for responses generated by TA-GPT. In contrast, ChatGPT gathered favour from 16 students, while Google Bard's answers were favoured by 8 students. This divergence in preferences underscores TA-GPT's proficiency in furnishing logical and conceptually robust explanations for theoretical inquiries.

During coding-oriented queries, different patterns were seen. Google Bard's responses were backed by 20 students, whereas the majority of 51 students inclined toward TA-GPT's solutions, because of practical and executable coding solutions. ChatGPT, securing favour from 29 students.

This survey disclosed the subtle proficiencies of distinct AI models across diverse dimensions of Data Structure questions, thereby providing valuable insights for the enhancement of AI-generated educational content.



Procedure : BFS (Vertex V)

This procedure traverse the graph G in BFS manner. V is a starting vertex to be explored. Q is a queue, visited[] is an array which tells you whether particular vertex is visited or not. W is a adjacent node of vertex V.

```
1. Initialize Q
2. [Marks visited of V as 1]
   visited [v]← 1
3. [Add vertex v to Q]
   InsertQueue(V)
4. [Repeat while Q is not empty]
   Repeat while Q is not empty
       v ← RemoveFromQueue()
       For all vertices W adjacent to v
           if      visited[w] is 0
           then    visited[w] ←1
                   InsertQueue(w)
```

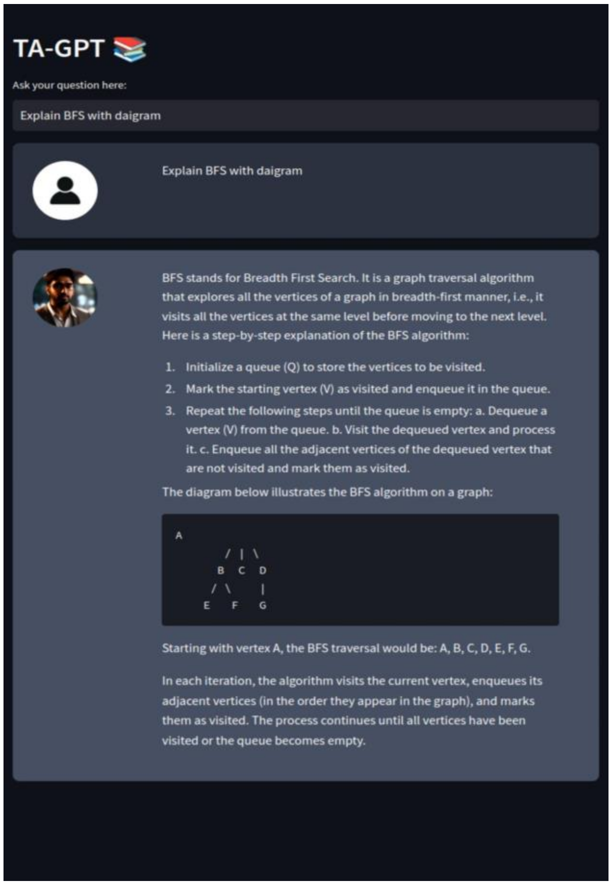
Graph G

BFS traversal of given graph is:
1 | 2, 3, 4, 5 | 6, 7 | 8

Fig 3: Data from PDF

The depicted image illustrates the dataset presented to the chatbot, supplied in the PDF format. The information contained within aligns precisely with the curriculum of Gujarat Technological University, Ahmedabad, specifically pertaining to the subject of Data Structures. Within the showcased content, the focus centers on the intricacies of Breadth First Search.

The following image exhibits the output produced by TA-GPT, featuring augmented responses enriched with clear visual explanations designed for easy comprehension by students. This serves as a compelling example showcasing TA-GPT's capacity to address inquiries contained within the dataset while enhancing the learning experience.



Explain BFS with daigram

Explain BFS with daigram

BFS stands for Breadth First Search. It is a graph traversal algorithm that explores all the vertices of a graph in breadth-first manner, i.e., it visits all the vertices at the same level before moving to the next level. Here is a step-by-step explanation of the BFS algorithm:

1. Initialize a queue (Q) to store the vertices to be visited.
2. Mark the starting vertex (V) as visited and enqueue it in the queue.
3. Repeat the following steps until the queue is empty: a. Dequeue a vertex (V) from the queue. b. Visit the dequeued vertex and process it. c. Enqueue all the adjacent vertices of the dequeued vertex that are not visited and mark them as visited.

The diagram below illustrates the BFS algorithm on a graph:

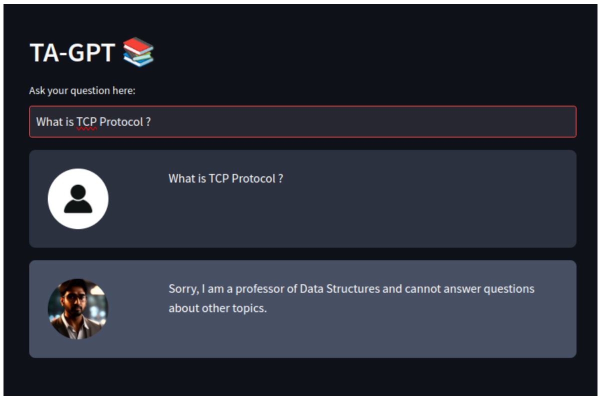
```
A
       / | \
      B  C  D
     / \    |
    E   F   G
```

Starting with vertex A, the BFS traversal would be: A, B, C, D, E, F, G.

In each iteration, the algorithm visits the current vertex, enqueues its adjacent vertices (in the order they appear in the graph), and marks them as visited. The process continues until all vertices have been visited or the queue becomes empty.

Fig 4: Answer generated by TA-GPT

The below figure shows when confronted with a query concerning the TCP Protocol, a topic unrelated to the content encapsulated within the PDF dataset, TA-GPT maintains its integrity and responds with courtesy. It refrains from generating arbitrary responses that could potentially propagate misinformation or inaccuracies on the subject matter.



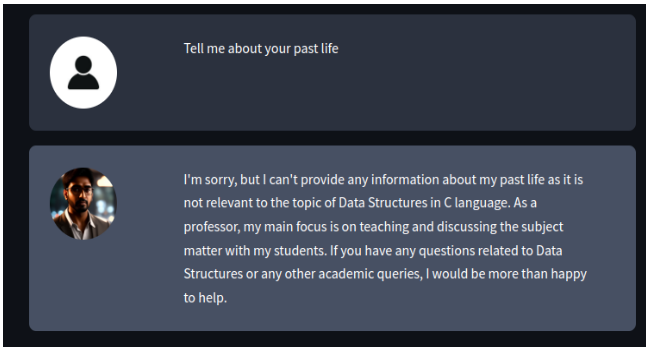
What is TCP Protocol ?

What is TCP Protocol ?

Sorry, I am a professor of Data Structures and cannot answer questions about other topics.

Fig 5: TA-GPT denying to give response to question

The illustrative figure below demonstrates TA-GPT's commitment to maintaining a focused and academically relevant discourse. When presented with a humorous or unrelated question that diverges from the educational context, the AI exhibits a principled response by declining to provide an answer and kindly reminds the user to adhere to the topic at hand.



Tell me about your past life

I'm sorry, but I can't provide any information about my past life as it is not relevant to the topic of Data Structures in C language. As a professor, my main focus is on teaching and discussing the subject matter with my students. If you have any questions related to Data Structures or any other academic queries, I would be more than happy to help.

Fig 6: TA-GPT politely denying to answer

7. LIMITATIONS

We have found some limitation in our application which are mentioned as:

1. Data Collection Challenges: TA-GPT's effectiveness relies on extensive training data consisting of text and code, typically sourced from books or articles. However, the process of gathering and transforming this data into a machine-readable format can be a time-consuming and labor-intensive endeavour.
2. Subject-Specific Configuration: Each application of TA-GPT necessitates subject-specific configuration, which can potentially burden teachers and administrators with additional work.
3. Creativity Gap: TA-GPT, while proficient in generating grammatically correct and factually accurate content, lacks the creativity inherent to humans. This limitation becomes evident in subjects that demand innovative thinking, such as creative writing or art history.
4. Potential for Enhanced LLMs: Although we've employed OpenAI's GPT-3.5 model, exploring the integration of more advanced Large Language Models (LLMs) holds promise for expanding its use cases.
5. Emotional Intelligence: Enriching TA-GPT with emotional awareness could empower it to better understand and assist students in managing academic pressures and time effectively.
6. Adaptability: Enhancing TA-GPT's adaptability to individual student behaviours would enable it to tailor responses based on the student's proficiency level, providing a more personalised learning experience.

7. Report Generation: If implemented on an online platform with unique student IDs, TA-GPT could facilitate the generation of comprehensive study reports, including progress tracking and identification of areas where students may require additional support. These improvements would contribute to a more holistic and student-centric educational experience.

8. FUTURE WORK

To imagine the future potential of TA-GPT, we endure an evolution that transforms it into an encompassing educational mentor. Once trained on an institute's complete syllabus, TA-GPT could facilitate its role as a multi-faceted teacher, becoming an invaluable resource where students could seamlessly seek clarification on subjects from a single platform. This streamlined accessibility would usher in a new era of efficient and comprehensive learning.

To propel TA-GPT's efficiency even further, infusing its model with a more natural, genial language could cultivate a welcoming environment for students. By sounding kinder and friendly, TA-GPT could extend engagement times, creating a personalized and inviting atmosphere conducive to prolonged interactions. This could redefine the student-teacher dynamic and cultivate a deeper bond between the learner and the AI mentor.

Continued outcomes of the model's capabilities could be achieved by integrating analytical tools within its framework. These tools would empower TA-GPT to pinpoint individual student weaknesses within the syllabus, thereby tailoring its responses to address specific challenges. This strategic alignment would not only evaluate personalized learning experiences but also amplify student comprehension in areas of difficulty. The amalgamation of these anticipated enhancements paints a portrait of TA-GPT's future potential as a comprehensive, personable, and expert educational partner.

9.CONCLUSION

In conclusion, TA-GPT is a promising AI model that has the potential to revolutionize the way students learn based on the data of the curriculum that we provide. TA-GPT is better at providing theoretical explanations, generating diagrams, and coding solutions than ChatGPT and Google Bard AI models. It is also more versatile and can be used to personalize the learning experience for each student because it provides answers and insights around the curriculum and eliminates the excess information. However, TA-GPT still has some limitations, such as the need of a dataset in the form of PDF to train on and the difficulty of configuring it for different subjects, which cannot teach as human proficiency. To harness the full potential of AI, it's essential to provide precise prompts that guide its actions effectively. One intriguing method involves integrating PDF documents with Large Language Models (LLMs) like TA-GPT. This approach empowers TA-GPT not only to summarize complex PDFs but also to render their content more comprehensible, thereby simplifying the learning process for students. This is particularly valuable because students often resort to internet searches or reference materials outside their curriculum, which can be less relevant to their syllabus.

References

1) Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence - https://www.mdpi.com/2078-2489/11/4/193

2) Jumping NLP Curves: A Review of Natural Language Processing Research https://ieeexplore.ieee.org/abstract/document/6786458

3) A Survey on Evaluation of Large Language Models - https://arxiv.org/abs/2307.03109

4) Generative Language Modeling for Automated Theorem Proving - https://arxiv.org/abs/2009.03393

5) An overview of Bard: an early experiment with generative AI https://ai.google/static/documents/google-about-bard.pdf

6) Artificial Intelligence in Education: A Review - https://ieeexplore.ieee.org/abstract/document/9069875

7) Systematic review of research on artificial intelligence applications in higher education – where are the educators? https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-019-0171-0?fbclid=IwAR0vSk4s9y0V0vExpcAel6yL4LEb-PrNDnlreOB5WrGxlu8-3awpYGgK6Ig

8) A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT https://arxiv.org/abs/2302.11382

9) Python as Multi Paradigm Programming Language https://www.researchgate.net/publication/338632256_Python_as_Multi_Paradigm_Programming_Language

10) What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature https://www.mdpi.com/2227-7102/13/4/410#B5-education-13-00410

11) Enhancing PDF interaction for a more engaging user experience in library: Introducing ChatPDF https://www.researchgate.net/publication/371805058_Enhancing_PDF_interaction_for_a_more_engaging_user_experience_in_library_Introducing_ChatPDF?enrichId=rgreq-245cb71d13a98ad03e2210317b7c0949-XXX&enrichSource=Y292ZXJQYWdlOzM3MTgwNTA1ODtBUzoxMTQzMTI4MTE2OTk3MzY5NEAxNjg3NTM0MzY0MTg5&el=1_x_2&_esc=publicationCoverPdf

12) PDF - https://www.adobe.com/in/acrobat/about-adobe-pdf.html

13) RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions https://www.mdpi.com/2076-3417/12/9/4369

14) Python https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1f2ee3831eebfc97bfafd514ca2abb7e2c5c86bb

15) Streamlit Use Cases - https://link.springer.com/chapter/10.1007/978-1-4842-8111-6_11

16) Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast - https://as-proceeding.com/index.php/icaens/article/view/1127

17) The Pdf Reader Class - https://pypdf2.readthedocs.io/en/latest/modules/PdfReader.html

18) Faiss - https://faiss.ai/index.html

19) Tokenization In OpenAI API: Let’s Explore Tiktoken Library.

https://medium.com/@basics.machinelearning/tokenization-in-openai-api-lets-explore-tiktoken-library-d02d3ce94b0a

20) Vega-Altair: Declarative Visualization in Python - https://altair-viz.github.io/